Optimization of Business Processes

Version of September 10, 2021
MG book*s*, Amsterdam

Optimization of Business Processes

Ger Koole

MG books
Amsterdam

Preface

The abundance of data and the necessity for many companies and institutions to maximize customer satisfaction and profit margins make this an excellent time for optimization specialists to work in. Knowledge of mathematical methods is required to solve business optimization problems. However, these are not the only skills that are required, solving quantitative business problems in a successful way demands also:
- the right non-scientific skills, including computer and communication skills, and a general idea of how the process of mathematical modeling works;
- enough knowledge of the practical side of optimization, such as the use of mathematical packages, and the ability to interpret their outcomes;
- adequate knowledge of the application domain.

If knowledge of any of these three domains is lacking, then a consultant cannot fully participate in a project, and he or she is bound to play a minor, often technical, role in the process. It needs no explanation that knowledge of computer tools and mathematical modeling is necessary. The same holds for communication skills and general knowledge of the solution process. The necessity of domain knowledge is less obvious: one often thinks that the *problem owner* brings in the domain knowledge, and the *problem solver* (or model builder) the modeling knowledge. However, to be able to communicate about the problem, to avoid non-implementable "optimal" solutions, to make it possible that the optimization specialist feels responsible for the proposed solution, and to give confidence to the problem owner that the problem solver knows what he or she is doing, it is necessary that he or she has the appropriate domain knowledge. An educational advantage of learning domain knowledge is that it illustrates well how mathematical techniques are used in practice.

In this book we pay attention to all aspects of the modeling process, while giving a central place to the business problem. This is what makes it different from most other books on optimization or Operations Research, in which the model and its solution techniques are the only subjects. This does not mean that this book could be read

instead of an introductory book on models and solution techniques. On the contrary, these notes can well be used for a graduate course in Business Analytics, assuming that the student already followed introductory courses in Operations Research (OR) and probability theory. At the same time, we hope that certain chapters are also appealing to students with a more qualitative background, and that they help to give a sound idea of the possibilities of OR for solving business problems.

Overview

These notes consist of three parts.

The first deals with mathematical models. They give an overview of stochastic models that are used in business applications. The focus is not on all mathematical details and possible extensions, but on its use in applications. Further investigation is stimulated by the section "Further reading" at the end of each chapter. You will find in this part all the models used in Part III.

The second part deals with modeling, some general properties of models, and software that is helpful when solving models. These chapters aim to give a general idea of the process of solving business problems using mathematical models. No prior knowledge is needed to read these chapters.

The third part is about various application areas. It starts with an introductory chapter giving an overview of application domains where Operations Research/ Management Science (OR/MS) is successfully used. It also explains which subjects can be found in the following chapters. In each of these chapters an application area is introduced and the standard models are discussed. The focus is both on solving the models and on the interplay between the business problems and the mathematical models.

How to use this book

This book is written for students and professionals with some background in probability theory and optimization. It is written for students OR/MS, Business Analytics, Industrial Engineering or similar curricula. It is my experience that students often have little knowledge of the non-mathematical aspects of OR/MS. For this reason I suggest that a course based on this book contains all chapters of Part II and some chapters of Part III. The chapters of Part I deal with the usual subjects of an introduction to stochastic models and optimization. Students who have already followed a

course on this subject can skip (parts of) these chapters.

In a single chapter we can only give short introductions to all of the applications areas. The same holds for the chapters with mathematical background. Evidently, we highly recommend further study. For this reason we included for each subject a section called "Further reading" containing some general texts on it, and references that allow the reader to study in more detail the aspects considered in the chapter.

Acknowledgements

This book is based on lecture notes that were written and improved each year while teaching at the Vrije Universiteit Amsterdam, from 1998 on. The input and reactions of all students that followed the different versions of the courses are acknowledged.

Many colleagues and students helped me improving earlier versions of this text, of whom I would like to mention René Bekker, Sandjai Bhulai, Marco Bijvank, Jan-Pieter Dorsman, Geert Jan Franx, Martijn Onderwater, Paulien Out, Corrie Quant, Rhonda Righter, Dennis Roubos, and Dirk Sierag. Rob van der Mei helped me with the section on location covering, Yoram Clapper wrote parts of the section on vehicle routing. Arnoud de Bruin helped me improve Chapter 16. Auke Pot helped me with older versions of Chapter 17, the current one is mostly based on the overview written with Siqiao Li. Marco Bijvank wrote the first versions of the Erlang C and X calculators that are used in Chapter 17. Kemal Berkan Arik rewrote them, added many more, and improved the website. A paper written by Maarten Soomer has been useful when writing on revenue management.

Ger Koole
Amsterdam/Etroubles
1998–2021

Contents

Part I

Stochastic Models

Chapter 1

Probability

The great Dutch soccer player Johan Cruijff (1947-2016), known for his aphorisms, knew it: ”Randomness is logical” (”Toeval is logisch”). Many people without a mathematical background think real-life phenomena are either random or completely predictable and that little can be be said in the case of randomness. They do not realize that randomness plays a role in almost everything around us, and that randomness can be measured and controlled. Probability and statistics give us the rules to calculate with random phenomena. Therefore it is fundamental to all mature sciences and economic activities.

Randomness is also a crucial aspect of many business problems. In this chapter we give a refresher of probability theory and we introduce some more advanced topics.

1.1 Random variables

Consider an experiment that can have multiple possible outcomes. For such an experiment we call the set of possible outcomes the *sample space*, usually denoted with Ω. A probability measure $\mathbb{P}$ assigns probabilities to subsets of Ω. These subsets are called *events*. Thus $\mathbb{P}(A)$ is the probability that the event $A \subset \Omega$ occurs. Evidently $\mathbb{P}() = 0$, $\mathbb{P}(\Omega) = 1$, and $\mathbb{P}(A) \leq \mathbb{P}(B)$ if $A \subset B$.

The practical interpretation of probability is that if an experiment is repeated many times, then the probability of an event signifies the long-run fraction of times (or *relative frequency*) that the event occurs. This is called the frequentist interpretation of probability. It shows also how probabilities can be estimated, by observing replications of an experiment and then taking the frequency. Indeed, frequentism is strongly related to statistical estimation methods. (Alternatively, probability can also

be seen as the quantification of a belief: this is the Bayesian interpretation.)

Example 1.1.1 The daily number of arrivals to a certain service center can be any natural number. In this case $\Omega = \{0, 1, \ldots\} = \mathbb{N}_0$. Repetition of this experiment leads to a row of natural numbers, e.g., 1,1,3,1,0,5,3,3,2,2,0,2. Then, for example, the fraction of 0's, 2/12 for the current realizations, can be used as an approximation for $\mathbb{P}(\{0\})$.

The sample space can be any set, but usually $\Omega \subset \mathbb{R}$. If this is not the case, then there is often a one-to-one relation between the elements of Ω and (a subset of) the real numbers. A random experiment taking values in the real numbers is called a *random variable*.

Example 1.1.2 Consider a machine, working at the beginning of a day, that may go down during that day or not. Consider the state of the machine at the end of the day. Then we can take $\Omega = \{\text{on}, \text{off}\}$, with $\mathbb{P}(\{\text{on}\})$ the probability that the machine does not go down. Now define the random variable X by taking $X(\text{off}) = 0$ and $X(\text{on}) = 1$. Then $\{X = 1\}$ corresponds to the event that the machine is on.

From now on we consider random variables taking values in the real numbers. There is no need to consider the (possibly different) underlying sample space.

There are several ways to characterize random variables. One is through the (cumulative) distribution function, usually denoted with F_X or F. The distribution function F_X of a random variable X denotes the following: $F_X(t) = \mathbb{P}(X \leq t) = \mathbb{P}(X \in (-\infty, t])$. Indeed, any (reasonable) set is composed of sets of the form $(-\infty, t])$. For example,

$$\mathbb{P}(X \in (s, t]) = \mathbb{P}(X \in (-\infty, t]) - \mathbb{P}(X \in (-\infty, s]) = F_X(t) - F_X(s),$$

for $s < t$. Therefore F_X fully specifies the distribution of X.

Note that F_X is always increasing, and that $F_X(-\infty) = 0$ and $F_X(\infty) = 1$. The set of all distributions can be roughly divided into two groups: those for which F_X is piecewise constant with at most a countable number of jumps, the discrete distributions, and those for which F is continuous and differentiable, the continuous distributions. (Also mixed versions of both types are possible, but they are less relevant.)

Discrete distributions are characterized by the probability mass on the points where F is discontinuous, i.e., where F makes a jump. For these values $\mathbb{P}$ has a non-negative value. Applied to a single value we call $\mathbb{P}$ the *probability mass function*. The set $\{t|\mathbb{P}(t) > 0\}$ is called the *support* of the random variable.

Example 1.1.3 We roll a die with possible outcomes $\{1,2,3,4,5,6\}$, each with equal probability. Then $F(t) = \lfloor t \rfloor /6$ for $t \in [0,6]$, 0 for $t < 0$ and 1 for $t > 6$. The probability mass function is given by $\mathbb{P}(t) = 1/6$ for $t \in \{1,2,3,4,5,6\}$.

Continuous distributions are characterized by $dF(t)/dt$, the *density* of the distribution, usually denoted with f. Note that

$$\int_u^v f(t)dt = \int_u^v F'(t)dt = F(v) - F(u),$$

thus f completely determines F. It also follows that $\int_{-\infty}^{\infty} f(t)dt = 1$. More generally, $\mathbb{P}(X \in A) = \int_A f(t)dt$ for $A \subset \mathbb{R}$. Instead of $\mathbb{P}(X \in A)$ we sometimes write $\mathbb{P}_X(A)$ or even $\mathbb{P}(A)$ when it is clear which random variable is meant.

Example 1.1.4 Consider a random variable X with distribution function $F(t) = t$ for $t \in [0,1]$, 0 for $t < 0$ and 1 for $t > 1$. Then $f(t) = 1$ for $t \in [0,1]$, and 0 elsewhere. It is right away clear that $\int_{-\infty}^{\infty} f(t)dt = 1$. ($X$ has a so-called *uniform distribution*; see page 16.)

Sometimes we are interested in the point t such that $F(t) \geq p$ for some $p \in (0,1)$. This point t is given by $t = F^{-1}(p)$ as long as F^{-1} is well defined. However, this is not always the case. For example, a discrete distribution is piecewise constant and F^{-1} is nowhere defined. To solve this we introduce the *quantile function*

$$F^{-1}(p) = \min_t \{F(t) \geq p\}. \tag{1.1}$$

Note that when F is strictly increasing then the quantile function coincides with the regular inverse of F.

The final characterization of a distribution is through its *hazard rate* function. This is the subject of Section 1.5. First however we need to introduce some other basic concepts of probability.

1.2 Expectations and moments

Quite often we are not interested in the outcome of an experiment, but in some function of the outcome. In the case of a random experiment, from a practical point of view, we are interested in the (long-run) average value of repetitions of the experiment. In mathematical terms, for a discrete distribution we define the expectation of $g(X)$, written as $\mathbb{E}g(X)$, as

$$\mathbb{E}g(X) = \sum_{x \in \mathbb{R}: \mathbb{P}(X=x)>0} g(x)\mathbb{P}(X = x). \tag{1.2}$$

If X has a continuous distribution with density f then we have

$$\mathbb{E}g(X) = \int_{-\infty}^{\infty} g(x)f(x)dx. \tag{1.3}$$

Note that any probability can be written as an expectation: $\mathbb{P}(X \in A) = \mathbb{E}\mathbb{I}\{X \in A\}$ with $\mathbb{I}$ the indicator function, i.e., $\mathbb{I}\{\cdot\} = 1$ if the argument is true, 0 otherwise.

The definition of $\mathbb{E}g(X)$ given in Equation (1.3) can be written as $\int g(x)dF(x)$. This notation is also used for the discrete case of Equation (1.2).

Note that, in general, $\mathbb{E}g(X) \neq g(\mathbb{E}X)$. In practice however, people often ignore variability when taking decisions. This is called the *Flaw of Averages*: Plans based on averages are wrong on average.

An special type of expectation that we often use in this text is $\mathbb{E}(X-t)^+ = \mathbb{E}\max(X-t,0)$ with $t \in \mathbb{R}$, usually $t > 0$. The interpretation is as follows. Let X be the demand of a certain product, and t the supply or inventory. Then $\mathbb{E}(X-t)^+$ is the expected demand in excess of the supply t. For this reason we call $\mathbb{E}(X-t)^+$ the *expected excess*.

Another reason to study expectations is the following. A random variable is completely specified by its distribution function. But unless it has a known distribution depending on a few parameters, it cannot be easily characterized. This is certainly the case with measured data. Instead one often gives (estimators of) its first few *moments*. The kth moment of a r.v. X is defined as $\mathbb{E}X^k$. The first moment is of course simply the expectation. The kth *central* moment is defined by $\mathbb{E}(X-\mathbb{E}X)^k$. The second central moment is known as the variance, usually denoted with $\sigma^2(X)$, its root is called the standard deviation. The variance has the following properties:

$$\sigma^2(X) = \mathbb{E}(X-\mathbb{E}X)^2 = \mathbb{E}X^2 - (\mathbb{E}X)^2,\ \sigma^2(aX) = a^2\sigma^2(X), \text{and } \sigma^2(a) = 0.$$

A dimensionless number that characterizes the variation is the *squared coefficient of variation* of a random variable X. It is defined by $c^2(X) = \sigma^2(X)/(\mathbb{E}X)^2$. It has the following properties: $c^2(aX) = c^2(X)$ and $c^2(a) = 0$. For examples of computations, see Section 1.6.

The variance $\sigma^2(X)$ and its root $\sigma(X)$, the *standard deviation* (SD), are the most common measures for dispersion. Another measure is the expected absolute deviation from the mean (sometimes called MAD, from mean absolute deviation), defined by $\mathbb{E}|X-\mathbb{E}X|$. The MAD is easier to interpret than the SD, but the SD has nicer mathematical properties and many results (such as the density of the normal distribution, see Section 1.6.9) are formulated in terms of it.

1.3 Multiple random variables and independence

Consider two random variables, X and Y (defined on the same probability space). We are interested in answering questions such as: what is $\mathbb{E}(X+Y)$? and $\mathbb{E}XY$?

The r.v. (X,Y) can be considered as a single two-dimensional random variable. If (X,Y) is discrete, then its distribution is defined by probabilities $\mathbb{P}((X,Y)=(x,y)) \geq 0$ with $\sum_{x,y}\mathbb{P}((X,Y)=(x,y))=1$, where the summation ranges over all x,y such that $\mathbb{P}((X,Y)=(x,y))>0$.

X itself is a random variable, with $\mathbb{P}(X=x)=\sum_y \mathbb{P}((X,Y)=(x,y))$. This is called the *marginal distribution*. In the same way the marginal distribution of Y can be found.

It holds that

$$\mathbb{E}(X+Y)=\sum_{x,y}(x+y)\mathbb{P}((X,Y)=(x,y))=$$

$$\sum_x x\sum_y \mathbb{P}((X,Y)=(x,y))+\sum_y y\sum_x \mathbb{P}((X,Y)=(x,y))=\mathbb{E}X+\mathbb{E}Y. \tag{1.4}$$

Thus, independent of the *simultaneous* distribution of (X,Y), $\mathbb{E}(X+Y)$ depends only on the marginal distributions of X and Y.

On the other hand, it does not always hold that $\mathbb{E}XY=\mathbb{E}X\mathbb{E}Y$, as the following example shows: take $\mathbb{P}((X,Y)=(0,1))=\mathbb{P}((X,Y)=(1,0))=1/2$. Then $\mathbb{E}XY=0\neq 1/4=\mathbb{E}X\mathbb{E}Y$. For $\mathbb{E}XY=\mathbb{E}X\mathbb{E}Y$ we need an additional condition, which we introduce next: *independence*.

The events A and B are independent if

$$\mathbb{P}(A\cap B)=\mathbb{P}(A)\mathbb{P}(B).$$

Two random variables X and Y are called independent if

$$\mathbb{P}(X\leq x, Y\leq y)=\mathbb{P}(X\leq x)\mathbb{P}(Y\leq y)$$

for all x,y. In that case the joint 2-dimensional distribution function is the product of the marginal 1-dimensional distribution functions. The same holds for the density of a continuous distribution.

In the case of a discrete distribution we now have

$$\mathbb{E}XY=\sum_{x,y}xy\mathbb{P}((X,Y)=(x,y))=\sum_{x,y}xy\mathbb{P}(X=x)\mathbb{P}(Y=y)=$$

$$\sum_x x\mathbb{P}(X=x)\sum_y y\mathbb{P}(Y=y)=\mathbb{E}X\mathbb{E}Y,$$

using independence in the second step.

From this follows, by some calculations, the following important formula for independent random variables X and Y:

$$\sigma^2(X+Y) = \sigma^2(X) + \sigma^2(Y).$$

We give some examples on how to compute expectations of functions of multiple independent random variables. Because of its relevance for the Pollaczek-Khintchine formula (see Section 5.3) we concentrate on the calculation of second moments.

An important class of compound distributions are random mixtures of the form $S = ZX_1 + (1-Z)X_2$ with $Z \in \{0,1\}$ and all variables independent. Then $\mathbb{E}S = p\mathbb{E}X_1 + (1-p)\mathbb{E}X_2$ with $p = \mathbb{P}(Z=1)$. Now

$$\mathbb{E}S^2 = \mathbb{E}[ZX_1 + (1-Z)X_2]^2 = \mathbb{E}(ZX_1)^2 + \mathbb{E}((1-Z)X_2)^2 + 2\mathbb{E}Z(1-Z)X_1X_2.$$

Using $Z(1-Z) = 0$, $\mathbb{E}Z^2 = \mathbb{E}Z$, and $\mathbb{E}(1-Z)^2 = \mathbb{E}(1-Z)$, all because $Z \in \{0,1\}$, and independence, we find

$$\mathbb{E}S^2 = p\mathbb{E}X_1^2 + (1-p)\mathbb{E}X_2^2. \tag{1.5}$$

After some computations it follows that

$$\sigma^2(S) = p^2\sigma^2(X_1) + (1-p)^2\sigma^2(X_2) + p(1-p)\mathbb{E}(X_1 - X_2)^2.$$

It is interesting to compare this with distributions of the form $\hat{S} = pX_1 + (1-p)X_2$, a convex combination of X_1 and X_2. Then $\mathbb{E}S = \mathbb{E}\hat{S}$, but

$$\mathbb{E}\hat{S}^2 = p^2\mathbb{E}X_1^2 + (1-p)^2\mathbb{E}X_2^2 + 2p(1-p)\mathbb{E}X_1\mathbb{E}X_2$$

and

$$\sigma^2(\hat{S}) = p^2\sigma^2(X_1) + (1-p)^2\sigma^2(X_2).$$

By comparing the expressions we see that the variance of a convex combination is always smaller than the corresponding random mixture.

An important example of a random mixture is the hyperexponential distribution, in which case the X_i are exponentially distributed.

1.4 Conditional probability

There is another important concept related to probability measures that will be used often throughout this book: conditional probability. Consider two events $A, B \subset \Omega$.

Then the conditional probability that A occurs given B, written as $\mathbb{P}(A|B)$, is defined by

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \tag{1.6}$$

which is only defined if $\mathbb{P}(B) > 0$. We often write $\mathbb{P}(AB)$ instead of $\mathbb{P}(A \cap B)$.

For a random variable X we are sometimes not just interested in $\mathbb{P}(X \in A|X \in B) = \mathbb{P}(A|B)$, but in $\mathbb{P}(X \in A|X \in B)$ for all possible A. The resulting random variable, taking values in B, is denoted with $X|X \in B$. Now the expectation $\mathbb{E}[g(X)|X \in B]$ can be defined in the obvious way. For example, if X is discrete, we get:

$$\mathbb{E}[g(X)|X \in B] = \sum_x g(x)\mathbb{P}(X = x|X \in B).$$

We use Equation (1.6) to obtain:

$$\mathbb{P}(A) = \mathbb{P}(AB) + \mathbb{P}(AB^c) = \mathbb{P}(A|B)\mathbb{P}(B) + \mathbb{P}(A|B^c)\mathbb{P}(B^c),$$

where B^c denotes the complement of B. This is called the *law of total probability*. It can be generalized as follows: let $B_1, B_2, \ldots$ be events such that $B_i \cap B_j = \emptyset$, and $\cup_{k=1}^{\infty} B_k \supset A$. Then

$$\mathbb{P}(A) = \sum_{k=1}^{\infty} \mathbb{P}(A|B_k)\mathbb{P}(B_k). \tag{1.7}$$

Using (1.6) we find

$$\mathbb{P}(A|C) = \frac{\mathbb{P}(AC)}{\mathbb{P}(C)} = \sum_{k=1}^{\infty} \frac{\mathbb{P}(AB_kC)\mathbb{P}(B_kC)}{\mathbb{P}(B_kC)\mathbb{P}(C)} = \sum_{k=1}^{\infty} \mathbb{P}(A|B_kC)\mathbb{P}(B_k|C), \tag{1.8}$$

a useful generalization of (1.7).

Let X be some r.v. on the same probability space. By integrating or summing Equation (1.7) over the probability space we find another useful formula:

$$\mathbb{E}X = \sum_{k=1}^{\infty} \mathbb{E}(X|B_k)\mathbb{P}(B_k). \tag{1.9}$$

1.5 Hazard rates

The idea of the hazard rate comes from the maintenance of systems, where it is crucial to know the remaining lifetime of a component given that it is currently functioning. It is also important to insurances.

Let X be a positive, continuous random variable with density f. Then, with $\overline{F}(t) = 1 - F(t)$:

$$\mathbb{P}(X \leq t+h | X > t) = \frac{\mathbb{P}(t < X \leq t+h)}{\mathbb{P}(X > t)} = \frac{\int_t^{t+h} f(s)ds}{\overline{F}(t)} \approx \frac{f(t)h}{\overline{F}(t)}. \tag{1.10}$$

This approximation gets more accurate and eventually becomes an equality as $h \to 0$. To be able to write this in a mathematically correct way we introduce the following concept.

Definition 1.5.1 *A function $f(h)$ is* of small order $g(h)$, *notated as $o(g(h))$, if*

$$\lim_{h \to 0} \frac{f(h)}{g(h)} = 0.$$

Example 1.5.2 $h^2 = o(h)$, because $h^2/h \to 0$.

Using Definition 1.5.1 we can make Equation (1.10) more precise:

$$\mathbb{P}(X \leq t+h | X > t) = \frac{f(t)h}{\overline{F}(t)} + o(h). \tag{1.11}$$

This motivates the definition of the *hazard rate* $\lambda(t)$.

Definition 1.5.3 *The hazard rate $\lambda(t)$ of a random variable with density f is given by*

$$\lambda(t) = \frac{f(t)}{\overline{F}(t)}.$$

Thus $\lambda(t)h$ is approximately the probability that X fails in the first h time units after t. Instead of hazard rate one also uses the term *failure rate*; this terminology comes evidently from its use in the study of systems that are prone to failure. We use the more neutral term hazard rate.

Example 1.5.4 An exponential distribution (see Section 1.6.5) with parameter γ has $F(t) = 1 - \exp(-\gamma t)$, $f(t) = \gamma \exp(-\gamma t)$, and thus $\lambda(t) = \gamma$ for all t.

The hazard rate completely characterizes a distribution. To see this, define $\Lambda(t) = \int_0^t \lambda(s)ds$, the *hazard function*. We see that

$$\Lambda(t) = \int_0^t \frac{f(s)}{\overline{F}(s)} ds = -\log \overline{F}(t), \tag{1.12}$$

and therefore $\overline{F}(t) = \exp(-\Lambda(t))$. Thus the distribution function F is completely determined by λ (and by Λ as well).

1.5.1 Minima and sums of random variables

A useful property of random variables with hazard rates is the following. Consider independent X and Y with hazard rates λ_X and λ_Y. Then

$$\mathbb{P}(\max\{X,Y\} \leq t+h | X, Y > t) = \mathbb{P}(X, Y \leq t+h | X, Y > t) =$$

$$\mathbb{P}(X \leq t+h | X > t)\mathbb{P}(Y \leq t+h | Y > t) =$$

$$[\lambda_X(t)h + o(h)][\lambda_Y(t)h + o(h)] = o(h),$$

because $h^2 = o(h)$. This can also be interpreted as follows: if events are happening at a certain rate in parallel, then the probability of more than one event happening in an interval of length h is $o(h)$.

A similar argument can be used to determine the hazard rate of minima of random variables:

$$\mathbb{P}(\min\{X,Y\} \leq t+h | X, Y > t) =$$

$$\mathbb{P}(X \leq t+h, Y > t+h | X, Y > t) + \mathbb{P}(X > t+h, Y \leq t+h | X, Y > t) +$$

$$\mathbb{P}(X, Y \leq t+h | X, Y > t) =$$

$$[\lambda_X(t)h + o(h)][1 - \lambda_Y(t)h + o(h)] + [1 - \lambda_X(t)h + o(h)][\lambda_Y(t)h + o(h)] + o(h) =$$

$$\lambda_X(t)h + \lambda_Y(t)h + o(h).$$

Thus the hazard rate of a minimum is the sum of the hazard rates.

Instead of maxima we can also look at sums:

$$\mathbb{P}(X+Y \leq s+t+h | X > s, Y > t) \leq \mathbb{P}(X \leq s+h, Y \leq t+h | X > s, Y > t) =$$

$$\mathbb{P}(X \leq s+h | X > s)\mathbb{P}(Y \leq t+h | Y > t) = \tag{1.13}$$

$$[\lambda_X(s)h + o(h)][\lambda_Y(t)h + o(h)] = o(h).$$

Again, the probability of two events happening in h time is $o(h)$.

1.6 Some important distributions

In this section we present some distributions that play a role in this book. They are all implemented in mathematical packages such as Maple and Matlab, and also in spreadsheet packages such as Excel.

1.6.1 The alternative or Bernoulli distribution

A random variable Z on $\{0,1\}$ has an alternative or Bernoulli distribution with parameter $p \in (0,1)$ if $\mathbb{P}(Z=1) = 1 - \mathbb{P}(Z=0) = p$. It represents the outcome of flipping a coin: if $p = 0.5$ then the coin is called *unbiased*. We find $\mathbb{E}Z = \mathbb{E}Z^2 = p$, and thus $\sigma^2(Z) = p(1-p)$.

1.6.2 The geometric distribution

A random variable N on $\mathbb{N}$ has a geometric distribution with parameter p if the following holds for $n \in \mathbb{N}$:

$$\mathbb{P}(N=n) = (1-p)^{n-1}p.$$

Some important properties of the geometric distribution are:

$$\mathbb{E}N = 1/p, \quad \mathbb{P}(N \leq n) = 1 - (1-p)^n.$$

A geometric distribution can be constructed as follows. Consider a number of independent alternative experiments with parameter p. If 1 occurs then we stop. The total number of experiments then has a geometric distribution with parameter p.

A special property of the geometric distribution is the fact that it is *memoryless*:

$$\mathbb{P}(N = m+n | N > n) = \frac{\mathbb{P}(N=m+n)}{\mathbb{P}(N>n)} = \frac{(1-p)^{m+n-1}p}{(1-p)^n} = \mathbb{P}(N=m),$$

for all $m > 0$ and $n \geq 0$. We conclude that the distribution of $N - n | N > n$ is independent of n. Thus, in terms of life times, the remaining life time distribution is independent of the current age. This is why N is called memoryless.

1.6.3 The binomial distribution

A binomial random variable N with parameters K and p has the following distribution:

$$\mathbb{P}(N=n) = \binom{K}{n} p^n (1-p)^{K-n},$$

for $n \in \{0,\ldots,K\}$. Remember that $\binom{K}{n} = \frac{K!}{n!(K-n)!}$. A binomial random variable can be interpreted as the sum of K alternative experiments with parameter p. Thus N denotes the number of 1's. From this it follows that $\mathbb{E}N = pK$ and $\sigma^2(N) = p(1-p)K$. From this interpretation it is also intuitively clear that sums of binomial random variables with the same probabilities are again binomial random variables.

1.6.4 The Poisson distribution

The Poisson distribution is a discrete distribution on $\mathbb{N}_0$. For a Poisson distribution N with parameter λ holds:

$$\mathbb{P}(N = n) = \frac{\lambda^n}{n!} e^{-\lambda}.$$

We have

$$\mathbb{E}N = \lambda \text{ and } \mathbb{E}N^2 = \lambda(1 + \lambda)$$

and thus

$$\sigma^2(N) = \lambda \text{ and } c^2(N) = \frac{\sigma^2(N)}{(\mathbb{E}N)^2} = \frac{1}{\lambda}.$$

For the absolute deviation it holds that

$$\mathbb{E}|N - \mathbb{E}N| = 2e^{-\lambda} \frac{\lambda^{\lfloor \lambda \rfloor + 1}}{\lfloor \lambda \rfloor !}. \tag{1.14}$$

See Exercise 1.7 for the derivation.

An important property of the Poisson distribution is the fact that sums of independent Poisson distributed random variables have again Poisson distributions. From this and the Central Limit Theorem (see Section 1.7) it follows that the Poisson distribution $Poisson(\lambda)$ with λ big is well approximated by the normal distribution $N(\lambda, \lambda)$. (See Subsection 1.6.9 for more information on the normal distribution.)

A Poisson distribution can be interpreted as the limit of a number of binomial distributions; see Section 2.1.

Another property is as follows. Construct from the Poisson distribution N two new distributions N_1 and N_2: each point in N is assigned independently to N_1 according to an alternative distribution with success probability p, otherwise it is assigned to N_2. Then N_1 and N_2 have independent Poisson distributions.

1.6.5 The exponential distribution

The exponential distribution plays a crucial role in many parts of this book. Recall that for X exponentially distributed with parameter $\mu \in \mathbb{R}_{>0}$ holds:

$$F_X(t) = \mathbb{P}(X \leq t) = 1 - e^{-\mu t}, \quad f_X(t) = F_X'(t) = \mu e^{-\mu t}, \quad t \geq 0,$$

$$\mathbb{E}X = \int_0^\infty t\mu e^{-\mu t} dt = \frac{1}{\mu}, \quad \mathbb{E}X^2 = \int_0^\infty t^2 \mu e^{-\mu t} dt = \frac{2}{\mu^2}, \tag{1.15}$$

$$\sigma^2(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2 = \frac{1}{\mu^2}, \text{ and } c^2(X) = \frac{\sigma^2(X)}{(\mathbb{E}X)^2} = 1.$$

For the hazard rate we find

$$\lambda(t) = \frac{f(t)}{\overline{F}(t)} = \frac{\mu e^{-\mu t}}{e^{-\mu t}} = \mu.$$

An extremely important property of the exponential distribution is the fact that it is *memoryless*:

$$\mathbb{P}(X \leq t+s | X > t) = \frac{\mathbb{P}(X \leq t+s, X > t)}{\mathbb{P}(X > t)} = \frac{\mathbb{P}(X \leq t+s) - \mathbb{P}(X \leq t)}{e^{-\lambda t}} =$$

$$\frac{e^{-\lambda t} - e^{-\lambda(t+s)}}{e^{-\lambda t}} = 1 - e^{-\lambda s} = \mathbb{P}(X \leq s).$$

We continue with some properties of $\min\{X, Y\}$ if both X and Y are exponentially distributed (with parameters λ and μ, respectively) and independent:

$$\mathbb{P}(\min\{X, Y\} \leq t) = 1 - \mathbb{P}(\min\{X, Y\} > t) = 1 - \mathbb{P}(X > t, Y > t) =$$

$$1 - \mathbb{P}(X > t)\mathbb{P}(Y > t) = 1 - e^{-\lambda t}e^{-\mu t} = 1 - e^{-(\lambda+\mu)t}.$$

Thus $\min\{X, Y\}$ is again exponentially distributed with as rate the sum of the individual rates. Repeating this argument shows that the minimum of any number of exponentially distributed random variables has again an exponential distribution. We also have:

$$\mathbb{P}(X \leq Y | \min\{X, Y\} \geq t) = \frac{\mathbb{P}(X \leq Y, \min\{X, Y\} \geq t)}{\mathbb{P}(\min\{X, Y\} \geq t)} =$$

$$\frac{\mathbb{P}(X \leq Y, X \geq t, Y \geq t)}{\mathbb{P}(X \geq t, Y \geq t)} = \frac{\mathbb{P}(X \leq Y, X \geq t)}{\mathbb{P}(X \geq t)\mathbb{P}(Y \geq t)} = \frac{\int_t^\infty \int_x^\infty \lambda e^{-\lambda x} \mu e^{-\mu y} dy dx}{e^{-\lambda t}e^{-\mu t}} =$$

$$\frac{\int_t^\infty \lambda e^{-\lambda x} e^{-\mu x} dx}{e^{-\lambda t}e^{-\mu t}} = \frac{\frac{\lambda}{\lambda+\mu} e^{-\lambda t}e^{-\mu t}}{e^{-\lambda t}e^{-\mu t}} = \frac{\lambda}{\lambda+\mu}.$$

This means that the probability that the minimum is attained by X in $\min\{X, Y\}$ is proportional to the rate of X, independent of the value of $\min\{X, Y\}$.

Finally, consider aX with a a constant and X exponentially distributed with parameter μ. Then

$$\mathbb{P}(aX \leq t) = \int_0^{\frac{t}{a}} \mu e^{-\mu x} dx = \frac{1}{a} \int_0^t \mu e^{-\mu \frac{x}{a}} dx = \mathbb{P}(Y \leq t)$$

with Y exponentially distributed with parameter μ/a. Thus the parameter of the exponential distribution is a *scale* parameter: changing it does not change the shape, only the scale. A consequence is that the coefficient of variation does not depend on the parameter.

1.6.6 The hyper-exponential distribution

The hyper-exponential distribution is a random mixture of k exponential distributions, the nth having probability p_n and parameter μ_n. We find for X hyper-exponential and X_n exponential(μ_n):

$$F_X(t) = 1 - \sum_{n=1}^{k} p_n e^{-\mu_n t}, \quad f_X(t) = \sum_{n=1}^{k} p_n \mu_n e^{-\mu_n t}, \quad t \geq 0,$$

and also

$$\mathbb{E}X = \sum_{n=1}^{k} \frac{p_n}{\mu_n}, \quad \sigma^2(X) = \sum_{n=1}^{k} \frac{2p_n}{\mu_n^2} - \Big(\sum_{n=1}^{k} \frac{p_n}{\mu_n} \Big)^2,$$

using (1.5) for the calculation of $\mathbb{E}X^2$. It can be seen that $c^2(X) \geq 1$.

1.6.7 The gamma distribution

The sum of k independent exponentially distributed random variables with parameter $\mu \in \mathbb{R}_{>0}$ has a gamma distribution with parameters $k \in \mathbb{N}$ and μ. For obvious reasons k is called the shape parameter and μ is called the scale parameter. We have for $X \sim \text{gamma}(k, \mu)$:

$$\mathbb{E}X = \frac{k}{\mu}, \; \sigma^2(X) = \frac{k}{\mu^2}, \text{ and } c^2(X) = \frac{1}{k}. \tag{1.16}$$

The density f_X and distribution function F_X are as follows, for $t \geq 0$:

$$f_X(t) = \frac{\mu e^{-\mu t} (\mu t)^{k-1}}{(k-1)!}, \quad F_X(t) = 1 - \sum_{n=0}^{k-1} \frac{(\mu t)^n}{n!} e^{-\mu t}.$$

For N a Poisson distribution with parameter μt the following interesting relation exists: $F_X(t) = \mathbb{P}(N \geq k)$. The intuition behind this relation is explained in Chapter 2.

Remark 1.6.1 The gamma distribution can be generalized to non-integer values of k using the *gamma function*; for k integer the distribution is sometimes called the *Erlang distribution*. A sum of exponential distributions not having the same parameter is sometimes called a *hypo-exponential distribution*. An even more general class of distributions composed of exponential *phases* are the *phase-type distributions*. Both hyper and hypo-exponential distributions are special cases.

1.6.8 The uniform distribution

The uniform distribution on $[0,1]$ has a density 1 on $[0,1]$ and 0 elsewhere. Its distribution function F is given by $F(t) = t$ on $[0,1]$, $F(t) = 0$ for $t \leq 0$ and $F(t) = 1$ for $t \geq 1$. Let X be uniform on $[0,1]$. Its properties are:

$$\mathbb{E}X = \frac{1}{2},\ \sigma^2(X) = \frac{1}{12},\ \lambda(t) = \frac{1}{1-t}.$$

This distribution can simply be generalized to the domain $[a,b]$ by taking $f = (b-a)^{-1}$ on it and 0 elsewhere.

1.6.9 The normal distribution

The normal distribution arises naturally when averaging i.i.d. (independent and identically distributed) random variables, see the Central Limit Theorem (Section 1.7). A normally distributed r.v. X with parameters μ and σ (denoted as $X \sim N(\mu, \sigma^2)$) has a density f given by:

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}, \quad t \in \mathbb{R}.$$

This density is symmetric around μ, and has the well-known bell shape. The distribution is also known as the Gaussian distribution.

A closed-form expression for the distribution function does not exist. The expectation and variance are given by:

$$\mathbb{E}X = \mu, \quad \sigma^2(X) = \sigma^2.$$

The standard normal distribution is denoted with $N(0,1)$, and $(X-\mu)/\sigma \sim N(0,1)$. The distribution function of the standard normal distribution is usually denoted with

Φ (and its density with ϕ). Thus, with $X \sim N(0,1)$, $\mathbb{P}(X \leq x) = \Phi(x)$. Traditionally, books on probability or statistics contained a table with $\Phi(x)$ for different values of x. They are nowadays replaced by software, for example, the Excel function NORMDIST.

To have a general idea of the meaning of the standard deviation of the normal distribution it is convenient to remember:

$$\mathbb{P}(\mu - \sigma \leq X \leq \mu + \sigma) \approx 0.68 \text{ and } \mathbb{P}(\mu - 2\sigma \leq X \leq \mu + 2\sigma) \approx 0.95.$$

An important property of the normal distribution is that independent sums of normal distributions have again normal distributions.

For the normal distribution we can calculate the expected excess $\mathbb{E}(X-t)^+$ (see Section 1.2). It is given by

$$\mathbb{E}(X-t)^+ = \frac{\sigma}{\sqrt{2\pi}}\exp\Big(\frac{-(\frac{t-\mu}{\sigma})^2}{2}\Big) - (t-\mu)\Big(1 - \Phi\Big(\frac{t-\mu}{\sigma}\Big)\Big). \tag{1.17}$$

From this we can calculate the expected absolute deviation from the mean:

$$\mathbb{E}|X-\mu| = \sqrt{\frac{2}{\pi}}\sigma. \tag{1.18}$$

1.6.10 The lognormal distribution

When Y has a normal distribution, then the positive random variable $X = e^Y$ has a lognormal distribution.

The normal distribution arises when we encounter sums of random variables, as is discussed in the next section. The lognormal shows up in case of multiple multiplicative effects. There is evidence that this is the case with the length of conversations, and indeed the lognormal distribution fits well historical data on call durations and other positive durations we encounter in practice.

If $Y \sim N(\mu, \sigma^2)$, then

$$\mathbb{E}X = e^{\mu+\sigma^2/2}, \quad \sigma^2(X) = (e^{\sigma^2}-1)e^{2\mu+\sigma^2}. \tag{1.19}$$

The hazard rate of the lognormal distribution has a form which is similar to the density: it first increases to a maximum, and then it decreases, with 0 as its limit.

When implementing the lognormal distribution we often need an expression for $\mathbb{E}Y$ and $\sigma(Y)$ given $\mathbb{E}X$ and $\sigma(X)$. They are given by

$$\mathbb{E}Y = \log \mathbb{E}X - 0.5 \log\Big(1 + \frac{\sigma^2(X)}{(\mathbb{E}X)^2}\Big) \text{ and } \sigma(Y) = \sqrt{\log\Big(1 + \frac{\sigma^2(X)}{(\mathbb{E}X)^2}\Big)} \tag{1.20}$$

1.7 Limit theorems

In the beginning of this chapter it was noted that the practical interpretation of the probability of an event is that of the long-run frequency that the event occurs. To make probability theory practically relevant it is therefore crucial that within the mathematical framework frequencies of events converge to the corresponding probabilities. For this reason limit theorems, and especially the *law of large numbers*, are at the heart of probability.

Consider some r.v. X for which we like to approximate $\mathbb{E}g(X)$. We do n i.i.d. experiments $X_1, \ldots, X_n$ with $X_i \sim X$. Then the Law of Large Numbers tells us that

$$\frac{g(X_1) + \cdots + g(X_n)}{n} \to \mathbb{E}g(X) \tag{1.21}$$

with probability 1. If we take $g(x) = \mathbb{I}\{x \in A\}$ for some event A, then we find:

$$\frac{\mathbb{I}\{X_1 \in A\} + \cdots + \mathbb{I}\{X_n \in A\}}{n} \to \mathbb{P}(X \in A),$$

which means that the frequency of an event converges to its probability, exactly as we want it to be.

Equation (1.21) is intuitively clear, because

$$\sigma^2\Big(\frac{g(X_1) + \cdots + g(X_n)}{n}\Big) = \frac{n}{n^2}\sigma^2(g(X)) \to 0,$$

the variance of the average of n i.i.d. random variables converges to 0.

How quickly does the variance of the average converge to 0? To answer this question, we make the following assumption to simplify notation: $g(x) = x$. This is not a restriction, as $g(X)$ is also a random variable. We use the following notation: $\hat{X}_n = (X_1 + \cdots + X_n)/n$. Then $\sigma^2(\hat{X}_n) = \sigma^2(X)/n$. Thus $\sqrt{n}(\hat{X}_n - \mathbb{E}X)$ has expectation 0 and variance $\sigma^2(X)$, independent of n. In addition, the distribution of $\sqrt{n}(\hat{X}_n - \mathbb{E}X)$ tends in the limit to a normal distribution:

$$\frac{\sqrt{n}(\hat{X}_n - \mathbb{E}X)}{\sigma(X)} = \frac{X_1 + \cdots + X_n - n\mathbb{E}X}{\sqrt{n}\sigma(X)} \to N(0,1),$$

with $N(0,1)$ the standard normal distribution. This result is known as the *Central Limit Theorem*. It is often used in statistics and in simulations, by assuming that it already holds for moderate values of n. In the following sections we will use it exactly

for these reasons: first for parameter estimation, and then for Monte Carlo simulation.

However, in these situations the variance $\sigma^2(X)$ is usually not known, it has to be estimated as well. For this reason we introduce the *sample variance* $S_n^2(X)$ by

$$S_n^2(X) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{X}_n)^2.$$

It can be shown that $\mathbb{E}S_n^2(X) = \sigma^2(X)$, i.e., $S_n^2(X)$ is an unbiased estimator of $\sigma^2(X)$. Again, the Central Limit Theorem tells us that

$$\frac{\sqrt{n}(\hat{X}_n - \mathbb{E}X)}{S_n} \to N(0,1),$$

with S_n the square root of S_n^2. This gives us the possibility to derive, for n sufficiently large, confidence intervals for $\mathbb{E}X$: with Φ the distribution function of $N(0,1)$, the $1-2\alpha$ confidence interval is given by

$$\Big[\hat{X}_n - \Phi^{-1}(1-\alpha)\frac{S_n}{\sqrt{n}}, \hat{X}_n + \Phi^{-1}(1-\alpha)\frac{S_n}{\sqrt{n}}\Big]. \tag{1.22}$$

It is interesting to note that in order to reduce the length of the confidence interval by a factor 2, approximately 4 times as many observations are needed.

Also quantiles and other performance measures can be analyzed in the same way, the former by writing expressions of the form $\mathbb{P}(X \geq t)$ as $\mathbb{E}\mathbb{I}\{X \geq t\}$.

The average and the sample variance are implemented in Excel under the functions AVERAGE and VAR.

1.8 Parameter estimation

Most of the models discussed in Part III need one or more parameters as input. Estimating these parameters is the subject of this section. Sometimes the form of a distribution needs to be estimated as well, but this happens rarely: often we have certain reasons to assume that a random variable has a certain distribution (for example the Poisson distribution as a model for customer arrivals; see Section 2.1), or the form of the distribution is of little or no relevance to the model (as is the case for the higher moments of the service time distribution in the Erlang B model of Theorem 5.4.3).

Many distributions (such as the exponential and Poisson distributions) are determined by a single parameter, the expectation. According to the Law of Large Numbers it suffices in these cases to average the outcomes to obtain an estimate of the parameter value.

Example 1.8.1 In call centers, most performance models use exponential service times. The parameter is estimated by averaging over a large number of realizations. Another parameter that needs to be approximated is the *patience* of customers: some callers abandon before they get connected. In these cases their patience was shorter than their waiting time. In cases where the waiting time is shorter than the patience, then the latter is not observed. Thus we deal with so-called *censored data*. If the patience X is exponentially distributed, and Y is the waiting time distribution, then for $Z = \min\{X, Y\}$ it follows (see Exercise 1.20) that $\mathbb{E}X = \mathbb{E}Z/\mathbb{P}(X < Y)$. By applying the Law of Large Numbers to both $\mathbb{E}Z$ and $\mathbb{P}(X < Y)$ we find as estimator for $\mathbb{E}X$ the sum over all waiting times divided by the number of abandoned customers. This is a special case of the *Kaplan-Meier estimator*.

Evidently, the average over a finite number of realization is rarely exactly equal to the expectation. Therefore we should also take the variance of the parameter estimation into account, using the Central Limit Theorem.

For a random variable X with a single parameter the estimator of $\mathbb{E}X$ can be used to estimate $\sigma^2(X)$. For a random variable with more parameters a separate estimator of the variance should be used: the sample variance $S_n^2(X)$ (see Section 1.7).

Example 1.8.2 We repeat an experiment involving a biased coin: $X \in \{0, 1\}$. We are interested in $\mathbb{P}(X = 1)$. For this reason we take $g(X) = \mathbb{I}\{X = 1\} = X$. Note that the last equality holds because $X \in \{0, 1\}$. Thus $\hat{X}_n = (X_1 + \cdots + X_n)/n \to \mathbb{P}(X = 1)$. Now we should realize that $\sigma^2(X) = \mathbb{P}(X = 1)(1 - \mathbb{P}(X = 1)) \leq 1/4$. To obtain a 95% precision interval of width 0.02 for our estimate $\hat{X}_n$ (± 0.01) we need that $\sigma(\hat{X}_n) \leq 0.005$ because of the normal approximation (see Section 1.6.9). We have $\sigma(\hat{X}_n) = \sigma(X)/\sqrt{n} \leq 1/(2\sqrt{n})$, Thus to obtain $\sigma(\hat{X}_n) \leq 0.005$ we need $n \geq 10000$: we need not less than 10000 repetitions to obtain the required precision!

Example 1.8.3 The number of arrivals N to a service system is counted for 12 identical periods: 2, 2, 2, 4, 2, 5, 1, 2, 1, 1, 0, 5. If N has a Poisson distribution then it suffices to compute $\hat{N}_n = 2.25$. This gives immediately an estimator $\hat{\sigma}^2(N)$ of the variance as $\mathbb{E}N = \sigma^2(N)$: $\hat{\sigma}^2(N) = 2.25$. Thus $[2.25 - \Phi^{-1}(0.975)\sqrt{2.25}/\sqrt{12}, 2.25 + \Phi^{-1}(0.975)\sqrt{2.25}/\sqrt{12}] = [2.25 - 2\sqrt{2.25}/\sqrt{12}, 2.25 + 2\sqrt{2.25}/\sqrt{12}] = [1.38, 3.12]$ is a 95% confidence interval for the parameter of the Poisson distribution. When the distribution of N is unknown we have to compute the sample variance: $S_n^2(N) = 2.57$. In this case the confidence interval for the expected number of arrivals becomes even wider.

The last example showed realizations of the number of arrivals to a service system, from identical periods. This is often an unrealistic assumption. In the general situation we have to make predictions of the future on the basis of historical data. This process, commonly known as *forecasting*, is discussed in Section 2.5.

1.9 Monte Carlo simulation

In Section 1.2 we saw how to compute $\mathbb{E}g(X)$ when the distribution of X is completely specified. However, obtaining a closed-form expression is not always possible. As an alternative, we could use the Law of Large Numbers: we make the computer generate a number of experiments $x_1, \ldots, x_n$ according to the distribution X and then we use $(g(x_1) + \cdots + g(x_n))/n$ as an estimate for $\mathbb{E}g(X)$. Then what remains is an estimation problem as in Section 1.8. This method is called (computer) *simulation*. Note that we usually have no idea about the distribution of the value we measure, otherwise we would probably be able to compute its value directly. Therefore we need the sample variance to compute confidence intervals for our observations.

There are two types of simulation models. The first type is the one described above, with X often multi-dimensional. This type is usually called *Monte Carlo simulation*. The other type consists of models in which we are interested in the long-run behavior of systems evolving over time (so called *dynamic systems*). This form of simulation is called *discrete-event simulation*. Discrete-event simulation is dealt with in Chapter 3. Here we consider Monte Carlo simulation.

Example 1.9.1 The estimation of next year's income statement of any company depends on many unknowns. Instead of working with point estimates for all entries, financial managers can work with distributions. Now, not only the expected income can be calculated, but also the probability of loss, and so forth.

There are several tools available for Monte Carlo simulation. Most used are those that are add-ins to the spreadsheet Excel, such as Crystal Ball. A disadvantage is that they are less focused on a mathematical analysis of the output. However, most tools do calculate the sample variance. In Crystal Ball it can be found in the report under "variance".

We finish this discussion of Monte Carlo simulation with discussing the way "random" numbers can be generated. This is crucial to any simulation program.

1.9.1 Pseudo-random numbers

The basic functionality of any simulation program is the fact that it can generate *pseudo-random numbers*. Pseudo-random numbers are not really random (they are generated by some deterministic algorithm), but a sequence of pseudo-random numbers resembles, for most practical purposes, sufficiently well to real random numbers. Pseudo-random numbers are usually integers between 0 and some very large number (say N). From that we can construct realizations of other probability distributions. For example, by dividing by N we get numbers that are uniformly distributed on $[0,1]$. Any random variable X with a known inverse distribution function F^{-1} can now be sampled as follows: if U is uniformly distributed on $[0,1]$, then $F^{-1}(U)$ has the same distribution as X:

$$\mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x) = \mathbb{P}(X \leq x).$$

For some distribution functions the inverse has a simple closed-form expression. This is for example the case for the exponential distribution. For other distribution functions the inverse is harder or even impossible to give. Distributions of this kind that we often use are the gamma distribution and the normal distribution. The gamma distribution can be seen as a sum of exponential distribution, and is therefore as easy to simulate as the exponential distribution. For the normal distribution efficient numerical methods exist.

The user is often not aware of the mathematical details discussed above. For example, the spreadsheet program Excel comes with a pseudo-random number generator RAND(), that can be used in conjunction with for example GAMMAINV() to generate random numbers from a number of different distributions. Additional functionality can be obtained by using add-ins, for example Crystal Ball. For debugging purposes it is essential that the program can be instructed to use the same random sequence every time, by setting a 'seed' value (the initial value of the random number generator).

1.10 Further reading

A book (and a website) dedicated to the Flaw of Averages is Savage [138].

Hazard rates and their properties are discussed in most books on reliability, such as Barlow & Proschan [18], Chapter 9 of Ross [130] (text book level) and Aven & Jensen [12] (advanced).

Most introductions to probability theory derive the Law of Large Numbers and the Central Limit Theorem. See, e.g., Ross [131].

The Kaplan-Meier method was introduced in [85]. Its details can be found on wikipedia and almost any text book on statistics.

1.11 Exercises

Exercise 1.1 Prove that $\mathbb{E}\max\{X, Y\} \geq \max\{\mathbb{E}X, \mathbb{E}Y\}$ for X and Y independent. Hint: Prove first that $\mathbb{E}\max\{a, X\} \geq \max\{a, \mathbb{E}X\}$.

Exercise 1.2 By Equation (1.4) we know that the expectation of the sum of a number of random variables is equal to the sum of the expectations. Show that the same does not hold for medians. Note that the median for a continuous distribution function F is defined by $F^{-1}(0.5)$.

Exercise 1.3 Prove Equation (1.11).

Exercise 1.4 Show that the sum of two independent Poisson distributions has again a Poisson distribution.

Exercise 1.5 A shop has a certain inventory of a certain product. The demand is exponentially distributed. Demand in excess of the inventory is lost. Give a formula for the expected sales and simplify this as much as possible.

Exercise 1.6 Consider a positive continuous random variable X. A graphical representation of it is the *Lorenz curve* L:

$$L(p) = \frac{\int_0^{F^{-1}(p)} tf(t)dt}{\mathbb{E}X}, \quad p \in [0, 1].$$

a. Give an interpretation of the Lorenz curve: what does $L(p)$ mean?
b. Give expressions of the Lorenz curve for X uniformly and exponentially distributed.

Exercise 1.7 Prove Equation (1.14). Hint: split the summation in a term running from 0 to $\lfloor \lambda \rfloor$ and a term from $\lfloor \lambda \rfloor + 1$ to ∞. This derivation is from Crow [46].

Exercise 1.8 From the Poisson distribution N two new distributions N_1 and N_2 are formed: each point in N is assigned independently to N_1 according to an alternative distribution with success probability p, otherwise it is assigned to N_2. Show that N_1 and N_2 have independent Poisson distributions.

Exercise 1.9 Let X and Y be independent exponentially distributed, with parameters λ and μ. Show that $\mathbb{P}(X < Y) = \lambda/(\lambda + \mu)$.

Exercise 1.10 Let $X \sim \exp(\mu_1)$ and $Y \sim \exp(\mu_2)$, with $\mu_1 \neq \mu_2$. Let $Z \in \{0, 1\}$ with $\mathbb{P}(Z = 1) = p$ and X, Y and Z independent. Define $U = ZX + (1 - Z)Y$ and $V = pX + (1 - p)Y$. (Note that U has a hyper-exponential distribution with $k = 2$.)
a. Compute $\mathbb{E}U$, $\mathbb{E}V$, $\sigma^2(U)$, and $\sigma^2(V)$.
b. Compute the distribution functions and densities of U and V.

Exercise 1.11 Let X have a hyper-exponential distribution.
a. Show that $c^2(X) \geq 1$.
b. For given $\beta > 0$ and $c > 1$ construct X with $k = 2$ such that $\mathbb{E}X = \beta$ and $c^2(X) = c$.

Exercise 1.12 a. Show (1.15).
b. Show (1.16).
c. Determine the hazard rate of a gamma distribution with 2 phases. Is it increasing or decreasing? Explain intuitively your answer.

Exercise 1.13 Let $X \sim \text{Uniform}[0, 1]$ and $Y \sim \text{Uniform}[a, b]$ with $a < b$.
a. Express Y as a function of X.
b. Compute $\mathbb{E}Y$ and $\sigma^2(Y)$ by using the answer to a and the expressions for X given in Section 1.6.8, and by calculating them directly.

Exercise 1.14 Prove Equation (1.20).

Exercise 1.15 Consider a hospital with two parallel ORs, each available for 8 hours. We have 14 operations to plan, all with expectation 1 hour. 8 of these have standard deviation 0, the 6 others 15 minutes.
a. We plan the operations with no variance on one OR, the other operations on the other OR. Estimate the expected number of operating rooms that exceed the planned finish time.
b. We plan 4 operations with no variance on each OR, and 3 operations with variance. Estimate the expected number of operating rooms that exceed the planned finish time.
c. Interpret the difference in findings between a and b.
For the total durations normal distributions can be used.

Exercise 1.16 Consider the expected excess as defined in Section 1.2.
a. Calculate the expected excess for a uniform random variable.
b. Calculate the expected excess for an exponential random variable.

Exercise 1.17 Consider the excess distribution $(X-t)^+$ and its expectation as defined in Section 1.2.
a. Formulate the excess distribution for a normal random variable.
b. Calculate the expected excess for a standard normal random variable.
c. Show Equation (1.17).

Exercise 1.18 a. Sum in Excel 50 realizations of a uniform distribution on $[-0.1, 0.1]$. Repeat this 500 times and plot the histogram of the empirical distribution.
b. Compare it to the normal distribution with the same average and standard deviation.
c. Multiply in Excel 50 realizations of a uniform distribution on $[0.9, 1.1]$. Repeat this 500 times and plot the histogram of the empirical distribution.
d. Derive μ and σ from Equation (1.19).
e. Compare the histogram to that of the lognormal distribution with the same average and standard deviation.

Exercise 1.19 Let $N_\lambda \sim \text{Poisson}(\lambda)$, and $X_\lambda \sim N(\lambda, \lambda)$.
a. Motivate, using the Central Limit Theorem, that X_λ is a good approximation for N_λ when λ gets big.
b. Determine, using some appropriate software tool, the minimal λ for which

$$\max_k |\mathbb{P}(N_\lambda \leq k) - \mathbb{P}(X_\lambda \leq k)| \leq 0.01.$$

Exercise 1.20 This exercise is related to Example 1.8.1.
a. For $Z = \min\{X, a\}$ and X exponential, show that $\mathbb{E}Z = \mathbb{E}X\mathbb{P}(X < a)$.
b. Use the answer to a. to show that $\mathbb{E}Z = \mathbb{E}X\mathbb{P}(X < Y)$ for $Z = \min\{X, Y\}$ and X exponential.

Exercise 1.21 In a call center the number of calls that arrive during a day has a Poisson distribution. At the end of a certain day there have been 5327 calls. Give a 95% confidence interval for the parameter of the distribution.

Exercise 1.22 During a day 4 operations are planned in a certain operation room in a hospital, each having a $N(1.5, 0.25)$ distributed duration. The operation room is

reserved for 7 hours. There is no lost time between operations and at the beginning of the day.
a. Use a simulation tool to estimate the probability that the operations take longer than the reserved time.
b. Verify this using a calculation.
c. Use a simulation tool to estimate the average time that operations exceed the planned time (finishing early is counted as 0).

Exercise 1.23 A bank employee processes mortgage requests involving a number of steps. Arrivals occur during working hours according to a Poisson process with rate 0.5. There are three processing steps, all having a uniformly distributed duration. Steps 1 and 3 are executed for every request, step 2 only for 30% of them. The upper and lower bound of the distribution are, in minutes: 30/40, 20/40, 40/60, respectively. Denote with S the time the employee works on an arbitrary request.
a. Compute $\mathbb{E}S$.
b. Estimate $\mathbb{E}S^2$ using simulation.
c. Estimate the expected time between arrival of a request and the time it has been processed using Theorem 5.3.2.

Exercise 1.24 The numbers of arrivals to a service center are noted during 100 days. The number at day i is given by x_i. We have $\sum_{i=1}^{100} x_i = 1763$ and $\sum_{i=1}^{100}(x_i - 17.63)^2 = 2351$.
a. Give a 95% confidence interval for the expected number of arrivals on a day.
b. Somebody thinks that the number of arrivals per day has a Poisson distribution. Is this likely to be the case? Motivate your answer.

Exercise 1.25 Consider n numbers $x_1, \ldots, x_n$.
a. Show that $\alpha = \sum_i x_i/n$ minimizes $\sum_i (x_i - \alpha)^2$.
b. Which number minimizes $\sum_i |x_i - \alpha|$?
c. And which number minimizes $\sum_i [p(x_i - \alpha)^+ + (1-p)(\alpha - x_i)^+]$, for $p \in (0,1)$?
Assume that $x_1, \ldots, x_n$ are i.i.d. realizations of some r.v. X.
d. Rephrase the results in terms of unbiased estimators of functions of X.

Exercise 1.26 Let 0.13, 0.47 and 0.67 be 3 realizations of a uniform distribution on [0,1]. On the basis of these numbers, calculate 3 realizations of an exponential distribution with rate 2 using the method described in Section 1.9.1.

Exercise 1.27 Random variables with a density f and a finite support $[m, M]$ can also be simulated as follows. Obtain a sample u from a uniform distribution on $[m, M]$,

and v from $[0,1]$. If $v < f(u)/\max_{x\in[m,M]} f(x)$ then accept the sample; otherwise reapeat this procedure.
a. Show that this algorithm gives sample according to the right distribution.
b. Use $1 - \mathbb{P}(X \in [\mu - 5\sigma, \mu + 5\sigma]) \approx 10^{-7}$ for X normally distributed to construct a simulation algorithm for the normal distribution.
c. Calculate the expected number of draws from a uniformly distributed random variable to generate 1 random number and think of a way to speed up the algorithm.
d. Implement this algorithm in a suitable computer language.

Exercise 1.28 Consider a random variable X with hazard rate $\lambda(t)$, with the property that there is a $\lambda > 0$ such that $\lambda(t) \leq \lambda$ for all t. Now construct a random variable Y as follows. Sample according to an exponential distribution with rate λ. Let the result be t. Now we draw according to a Bernoulli distribution with parameter $\lambda(t)/\lambda$. In the case of success $Y = t$. Otherwise, we sample again according to an exponential distribution with rate λ. For the result s, we draw according to a Bernoulli distribution with parameter $\lambda(t+s)/\lambda$. In the case of success $Y = t + s$. Otherwise, we draw again from the exponential distribution, etc.
Show that X and Y have the same distribution. (Hint: show that Y has hazard rate $\lambda(t)$.)
Note that this procedure gives a way to simulate distributions which have a bounded, known hazard rate.

Chapter 2

Customer arrivals and the Poisson Process

In this chapter we study customer arrival processes, especially the Poisson process. We give an informal introduction, based on an intuitive understanding of the Poisson distribution as modeling arrivals coming from a large population.

2.1 Motivation

Suppose we have a population of size K, where each individual has probability p_K of generating a request for service. Then the number of requests N_K has a binomial distribution:

$$\mathbb{P}(N_K = n) = \binom{K}{n} p_K^n (1 - p_K)^{K-n}.$$

It is intuitively clear that $\mathbb{E}N_K = \sum_{n=1}^{K} n\mathbb{P}(N_K = n) = p_K K$. Now increase K, while keeping the expected number of requests constant, i.e., $\mathbb{E}N_K = p_K K = \lambda$. Thus $p_K = \lambda/K$. Then

$$\lim_{K\to\infty} \mathbb{P}(N_K = n) = \lim_{K\to\infty} \binom{K}{n} p_K^n (1 - p_K)^{K-n} = \lim_{K\to\infty} \binom{K}{n} (\frac{\lambda}{K})^n (1 - \frac{\lambda}{K})^{K-n} =$$

$$\frac{\lambda^n}{n!} \lim_{K\to\infty} (1 - \frac{\lambda}{K})^K \frac{K!}{(K-n)!(K-\lambda)^n} = \frac{\lambda^n}{n!} e^{-\lambda} = \mathbb{P}(N = n)$$

for N having a Poisson distribution with parameter λ. Thus the Poisson distribution can be used to model the number of service requests coming from a large group of

potential users of the service. Its parameter λ represents the expected number of requests.

Now we consider also the *time* at which arrivals occur. Assume that they can occur in the interval $[0,T]$. For every request its arrival time is generated according to a probability distribution on $[0,T]$, independent from any other request. Now split $[0,T]$ in two intervals $[0,t]$ and $[t,T]$. Now let each request determine its arrival time according to a uniform distribution on $[0,T]$, thus every instant in $[0,T]$ is equally likely. Then an arrival occurs in $[0,t]$ ($[t,T]$) with probability t/T ($(T-t)/T$). (This is the homogeneous case, in Section 2.4 we discuss the general case.) Denote with $N(t,t')$ (or simply $N(t')$ if $t=0$) the number of arrivals in $[t,t']$, for $0 \leq t < t' \leq T$. Assume that $\mathbb{E}N(T) = \lambda T$, thus on average λ arrivals occur per time unit. Then, according to Exercise 1.8, $N(t)$ and $N(t,T)$ have independent Poisson distributions with parameters λt and $\lambda(T-t)$.

Both facts, derived from practically logical customer behavior, are used as a definition of the Poisson process.

2.2 The homogeneous Poisson process

Consider events, typically arrivals to some service center, that occur at random moments in $[0,\infty)$. Let $N(t)$, for every $t \in [0,\infty)$, be a random variable that counts the number of events in $[0,t]$. Then we call $N(t)$ a *counting process*. We also define $N(s,t) = N(t) - N(s)$, the number of arrivals in $(s,t]$, for $0 \leq s < t$.

Definition 2.2.1 *The counting process $N(t)$ on $[0,\infty)$ is called a (homogeneous) Poisson process with rate λ if:*
- $N(s,t)$ has a Poisson distribution with expectation $\lambda(t-s)$ for all $0 \leq s < t$;
- $N(s,t)$ and $N(s',t')$ are stochastically independent for all $0 \leq s < t \leq s' < t'$.

Now we consider interarrival times. Let X_1 be the time until the first arrival of a request. Let $0 \leq t \leq T$. Then

$$\mathbb{P}(X_1 > t) = P(N(t) = 0) = e^{-t\lambda}.$$

Thus X_1 has an exponential distribution. The same holds for X_2, the time between the first and second arrival:

$$\mathbb{P}(X_2 > t | X_1 = s) = P(N(s,s+t) = 0) = e^{-t\lambda}.$$

Note that $\mathbb{P}(X_2 > t|X_1 = s)$ does not depend on s, and thus X_1 and X_2 are also independent. This argument can be repeated for all other interarrival times. Thus all interarrival times of a homogeneous Poisson process have independent exponentially distributed interarrival times with the same parameter. This is the most often used definition of the Poisson process.

Definition 2.2.2 *The counting process $N(t)$ on $[0, \infty)$ is called a (homogeneous) Poisson process with rate λ and interarrival times $X_1, X_2, \ldots$ if all X_i are independent and identically exponentially distributed with parameter λ.*

At the end of this section we will show that both definitions are equivalent.

Definition 2.2.2 is very useful when we try to simulate a Poisson process: we simple generate realizations of the exponential sojourn times using the method described in Section 1.9.1.

There is a third definition of the Poisson process based on the concept of *rates*. In Section 1.5 we saw that the exponential distribution is defined by the fact that the hazard rate is constant. Thus we can see the Poisson process as a stream of points generated at a constant rate. In the next definition we use the notion "small order of", see Definition 1.5.1.

Definition 2.2.3 *The counting process $N(t)$ on $[0, \infty)$ is called a (homogeneous) Poisson process with rate λ if:*
- $N(s, t)$ and $N(s', t')$ are stochastically independent for all $0 \leq s < t \leq s' < t'$;
- $\mathbb{P}(N(t, t + h) = 1) = \lambda h + o(h)$, $\mathbb{P}(N(t, t + h) > 1) = o(h)$ for all $t \geq 0$ and $h > 0$.

Theorem 2.2.4 *Definitions 2.2.1, 2.2.2, and 2.2.3 are equivalent.*

Proof It has already been argued that Definition 2.2.2 follows from Definition 2.2.1. Vice versa, the time until the kth arrival from s on has a gamma or Erlang distribution. Using its distribution function it can be seen that $N(s, t)$ has a Poisson distribution. The independence follows from the memoryless property of the exponential distribution.

Definition 2.2.3 suggests yet another interpretation of the Poisson process. Every h time units a Bernoulli experiment is executed with success probability λh. If successful, an arrival occurs, otherwise nothing happens. In the limit, as $h \to 0$, this also gives a Poisson process (see Exercise 2.3).

The first part of Definition 2.2.3 follows from Definition 2.2.1, the second part from Definition 2.2.2. For the reverse we refer to Theorem 5.1 of Ross [130]. □

2.3 Merging and splitting

In this section we consider merging and splitting of Poisson processes. We start with merging: what can we say about the sum N of two independent Poisson processes N_1 and N_2? To show that N is again a Poisson process we check Definition 2.2.1. The first point of the definition is satisfied because of Exercise 1.4. The second point follows directly from the independence.

Now we consider splitting. With splitting we mean that from a Poisson process N two new processes N_1 and N_2 are formed: each point in N is assigned independently to N_1 according to an alternative distribution with success probability p, otherwise it is assigned to N_2. We claim that N_1 and N_2 are independent Poisson processes. This can be shown by checking the conditions of Definition 2.2.1, see Exercise 1.8.

2.4 The inhomogeneous Poisson process

As in Section 2.1, consider two intervals $[0,t]$ and $[t,T]$. For each arrival occuring in $[0,T]$ it was assumed that the arrival moment was determined according to a uniform distribution on $[0,T]$. Here we abandon this assumption, instead we assume that the arrival time is determined according to a distribution with a piece-wise continuous density f on $[0,T]$. Define $\gamma = \mathbb{E}N(T)$. Then $N(s,t)$ has a Poisson distribution with parameter $\gamma \int_s^t f(u)du$, and arrivals in disjunct intervals are again independent.

Define $\lambda(t) = f(t)\gamma$. The function $\lambda(t)$ is called the rate function, and it has the following interpretation: $\mathbb{E}N(s,t) = \int_s^t \lambda(u)du$, and $\frac{dN(t)}{dt} = \lambda(t)$ for all t (for which it exists). These observations lead to the following definition.

Definition 2.4.1 *The counting process $N(t)$ on $[0,\infty)$ is called an inhomogeneous Poisson process with rate function $\lambda(t)$ if:*
- $N(s,t)$ has a Poisson distribution with expectation $\int_s^t \lambda(u)du$ for all $0 \leq s < t$;
- $N(s,t)$ and $N(s',t')$ are stochastically independent for all $0 \leq s < t \leq s' < t'$.

Note that if $\lambda(t)$ is constant then we have a homogeneous Poisson process.

Let us now study the interarrival times. The time until the next arrival after a fixed point in time is characterized by the rate function $\lambda(t)$, which should now be interpreted as the hazard rate (see Section 1.5, in which we used the same notation). Indeed, with X_1 the time until the first arrival,

$$\mathbb{P}(X_1 > t) = \mathbb{P}(N(t) = 0) = e^{-\int_0^t \lambda(s)ds},$$

which is, according to Equation (1.12), equivalent to saying that $\lambda(t)$ is the hazard rate of X_1. Thus X_1 can have any distribution, depending on the rate function $\lambda(t)$.

Let us now consider the second interarrival time X_2. We have

$$\mathbb{P}(X_2 > t | X_1 = s) = P(N(s, s+t) = 0) = e^{-\int_s^{s+t} \lambda(u)du}.$$

This clearly depends on s, and thus X_1 and X_2 are dependent in general. For this reason we cannot formulate a definition equivalent to Definition 2.2.2 for the inhomogeneous Poisson process.

A definition using rates, equivalent to Definition 2.2.3, is more easily formulated.

Definition 2.4.2 *The counting process $N(t)$ on $[0, \infty)$ is called an inhomogeneous Poisson process with rate $\lambda(t)$ if:*
- $N(s,t)$ and $N(s',t')$ are stochastically independent for all $0 \leq s < t \leq s' < t'$;
- $\mathbb{P}(N(t, t+h) = 1) = \lambda(t)h + o(h)$, $\mathbb{P}(N(t, t+h) > 1) = o(h)$ for all $t \geq 0$ and $h > 0$.

Definition 2.4.2 can be used when we want to simulate inhomogeneous Poisson processes with bounded rates. This works as follows. Let the rate of the Poisson process be bounded by the number λ, that is, $\lambda(t) \leq \lambda$. Now we simulate a Poisson process with rate λ, and we take a point t of this simulation as a point in the inhomogeneous Poisson process with probability $\lambda(t)/\lambda$. This gives a process with rate $\lambda(t)$ (see also exercise 1.28).

Theorem 2.4.3 *Definitions 2.4.1 and 2.4.2 are equivalent.*

Proof Definition 2.4.2 follows from Definition 2.4.1 using properties of the Poisson distribution. The reverse follows from Theorem 1.3.1 of Tijms [152]. □

Definition 2.4.2 is equivalent to the results in Section 1.5.1 in the sense that for both cases, events governed by rates running in parallel and in series, the probability of two events in time h is $o(h)$.

2.5 Parameter estimation and forecasting

Suppose we observe a homogeneous Poisson process on $[0, \infty)$ with an (unknown) rate λ. Then $N(t)/t$ is an unbiased estimator of λ. We also have $N(t) = \sum_{s=1}^{t} N(s-1, s)$, with all $N(s-1, s)$ i.i.d. (independent and identically distributed). Thus the law of large numbers applies, and $N(t)/t \to \lambda$.

Consider that we have realizations $x_1, \ldots, x_n$ of the numbers of arrivals of consecutive time periods. To check whether they are likely to come from a homogeneous Poisson process we can calculate the sample variance

$$s_n^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{x}_n)^2.$$

If $s_n^2 \approx \hat{x}_n$ then it is well possible that the numbers come from Poisson distributions with the same rate because for a Poisson distribution expectation and variance are equal. However, it often occurs that $s_n^2 > \hat{x}_n$. We call this *overdispersion* with respect to the Poisson process. There are two possible reasons for this: either the numbers of arrivals are not Poisson, or the arrivals are Poisson but with different parameters.

Example 2.5.1 Traffic accidents occur according to a Poisson process, but they regularly involve more than one person. These *batch arrivals*, as seen by the ambulances and hospital emergency departments, cause overdispersion. At the same time, the parameter of the Poisson process depends on the day of the week, weather conditions, and so forth. This also causes overdispersion.

Most real customer arrival processes are inhomogeneous Poisson processes. They are usually modeled with piecewise-constant rate functions, because the expected numbers of arrivals are estimated for intervals of a fixed length in which the arrival rate is assumed to be constant. It is observed that these arrival processes often have weekly and daily patterns that change little over time. Forecasting amounts to estimating the number in each interval, thus the parameters of its Poisson distributions. Often this is done at the daily level, sometimes we have to drill down to the interval level. In this chapter we focus on the daily level, in Chapter 17 on call centers we will discuss the extension to the 15-minute level. In Figure 2.1 typical data is shown, from a call center. On top we see 4 years of monthly data, on the bottom 4 weeks of daily data. We see that subsequent years and weeks show a similar pattern, both are forms of what is called *seasonality*. We also see that week 4 is exceptional: it starts with zero volume. This particular day was Easter Monday, and the call center was closed. The following Tuesday the volume was exceptionally high.

There are many forecasting methods described in the scientific literature. To be able to deal with customer demand, features of these models should include non-stationary long-term trend, two forms of seasonality and events such as holidays. Few forecasting methods allow for all these features. A surprisingly simple and effective one with which everything can be modeled is linear regression, with dummy

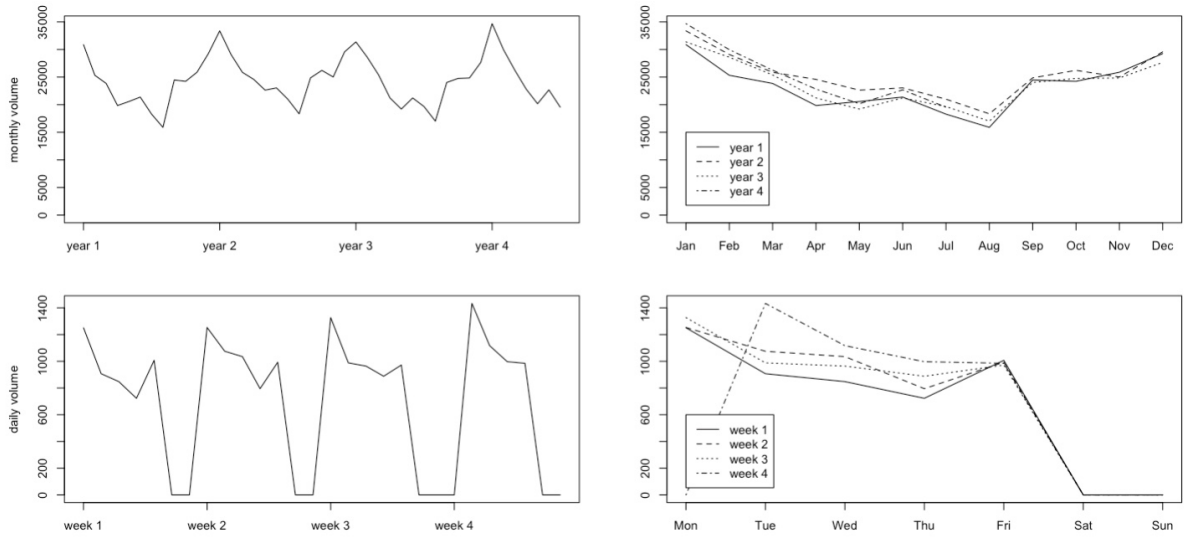


Figure 2.1: Example of intra-year and intra-week fluctuations

variables for the seasonal parameters (for each day/week in the season a 0/1 variable) and the events, and a low-degree polynomial for the long-term trend. For short-term forecasting the seasonal parameters and the events are most important, long-term forecasting (several months and beyond) is always unreliable because nobody knows what will happen to the trend. Seasonal parameters are best modeled as multiplicative in the trend, because the amplitude of the weekly pattern scales with the weekly volume, which by itself is determined by the trend. Therefore it makes sense to first take the logs because this makes the multiplicative effects additive.

This leads to the following mathematical model. The rate at day t is given by

$$\lambda_t = b(t)s^y_{w(t)}s^w_{d(t)}\prod_{n=1}^{E}(1+e_n h_n(t)), \tag{2.1}$$

with
- $b(t)$ some low-order polynomial of t;
- s^y the intra-year seasonal factors, one for every week, and $w(t)$ the week of day t;
- s^w the 7 intra-week seasonal factors, and $d(t)$ the weekday of day t;
- e_n representing the impact of event $n \in \{1, ..., E\}$ and $h_n(t)$ a 0/1 vector with $h_n(t) = 1$ if n happens on t.

The realization v_t at t has a Poisson distribution with parameter λ_t. The goal is to construct an estimate $\hat{\lambda}_t$ of λ_t. As explained above, we do this with a linear model for $\log v_t$ which approximates all logarithms of the coefficients.

Remark 2.5.2 When volumes are small the impact of the Poisson distribution is relatively big because $\sigma(N) = \sqrt(\lambda)$ for N Poisson distributed with expectation λ. In this case it certainly makes sense to stabilize the standard deviations first by using the transform $2\sqrt{v_t + 3/8}$, the so-called *Anscombe transform*. Additionally, it avoids the problem of taking a logarithm when the volume is 0.

Note that seasonal factors might change over time, that trend is not completely polynomial, and that the impact of events might not be fully predictable. Because of this all parameters are actually random. Modeling this would make the model unnecessarily complicated, for this reason we stick to the current model.

Of course we need to measure the error we make in our estimations. The most common error measurement is the *mean squared error* (MSE) defined by

$$T^{-1} \sum_{t=1}^{T} (\hat{\lambda}_t - v_t)^2,$$

for a forecast over T time periods. We can also take its root, the *root mean squared error* (RMSE), which has the advantage that the error is in the same unit as the realization (e.g., orders instead of orders^2). Although the MSE has many nice mathematical properties, it is hard to interpret by practitioners. Therefore several error measures based on absolute errors are introduced. The most popular one is the *mean absolute percentage error* (MAPE), defined by

$$T^{-1} \sum_{t=1}^{T} \frac{|\hat{\lambda}_t - v_t|}{v_t}.$$

The MAPE has two disadvantages:
- it is undefined if one of the v_ts is 0;
- it can be dominated by realizations with small volumes and high percentage errors.

A solution for both is weighing with the volume, leading to the *weighted absolute percentage error* (WAPE), which is equal to

$$\frac{\sum_{t=1}^{T} |\hat{\lambda}_t - v_t|}{\sum_{t=1}^{T} v_t}.$$

There are two reason why error measurements such as MSE or WAPE are always greater than 0: we make an error in estimating λ_t, and on top of that there is Poisson "noise". The goal of forecasting is to reduce the error in forecasting λ_t, the Poisson error cannot be avoided. To evaluate the quality of our forecast it would be good to

quantify the unavoidable Poisson error. To find the minimal APE we need to compute $\mathbb{E}|N_\lambda - \lambda|/\lambda$ with $N_\lambda \sim$ Poisson(λ). A very good approximation, based on the normal distribution, is given by $\sqrt{2/(\lambda\pi)}$. This follows easily from Equation (1.18). The exact formula is given by Crow [46]. Note that the minimal APE decreases as a function of λ.

Remark 2.5.3 Linear regression minimizes the MSE. This seems in contradiction with the fact that we propose to use an error measure based on absolute values. There is an alternative method that minimizes the MAPE: *quantile regression*. For points (x_i, y_i), $\sum_i[(1-\tau)(a+bx_i - y_i)^+ + \tau(y_i - a - bx_i)^+]$ is minimized by a τ-quantile (see Exercise 1.25), which is the median for $\tau = 0.5$. The quantile can be found using linear programming (see Exercise 2.9).

On the other hand, when we transform using logarithms, minimizing the MSE of the transformed numbers is equivalent to minimizing the geometric mean of the actuals which is close to the median (see, e.g., Section 3.2 of [78]). Thus minimizing the WAPE is close to minimizing the MSE of the logarithms.

2.6 Other arrival processes

Many arrivals streams in practice can be modeled accurately by the (inhomogeneous) Poisson process. However, there are a number of generalizations worth mentioning. We already encountered the Poisson process where each point is actually a batch of arrivals.

Another generalization starts from Definition 2.2.2. The interarrival times are again i.i.d., but not exponentially distributed anymore. Such a process is called a *renewal process*. Although mathematically interesting, there are few applications of this type of process. One exception is the process where the interarrival times are deterministic. This is typically the situation in which arrivals to a system are planned, such as production orders or patients in a clinic. This last example is particularly interesting, because patients often do not arrive exactly on time, but often a few minutes early and sometimes a little late. This adds a random shift to each arrival moment, making the analysis of systems with this type of processes particularly difficult. *Discrete-event simulation* is about the only useful solution technique. It is the subject of the next chapter.

2.7 Further reading

Almost every book on probability or stochastic models introduces the Poisson process. Chapter 1 of Tijms [152] gives an excellent introduction to the many properties of the Poisson process.

A recent accessible book on forecasting is Hyndman & Athanasopoulos [78], which can also be read online. More details on call center forecasting can be found in Ibrahim et al. [79] (theoretical) and Koole [99] (practical). The easiest way to obtain more information and references on the Anscombe transform is by reading its Wikipedia page. Koenker & Hallock [96] introduce quantile regression in a very accessible way.

Renewal processes are extensively discussed in Ross [130] and Tijms [152].

2.8 Exercises

Exercise 2.1 A department of a bank processes mortgage applications. There are 15 employees, each one is supposed to be able to handle 3 applications per day. On average 42 requests arrive at the beginning of each day. Applications have to be dealt with the day they arrive, when necessary in overtime.
a. Do you think that the Poisson distribution is a good choice for modeling the number of applications per day? Motivate your answer.
b. Give a formula for the probability that overtime is necessary and the expected number of applications that are processed during overtime.
c. Calculate these numbers using some appropriate software tool.

Exercise 2.2 Consider a process N on $[0,\infty)$ with the following properties:
- $N(s,t)$ has a Poisson distribution with expectation $\lambda(t-s)$ for all $0 \leq s < t$;
- given $N(s,t) = k$, the arrivals in $[s,t]$ are distributed according to k i.i.d. uniform distributions on $[s,t]$.
Show that this is an alternative definition of a Poisson process.

Exercise 2.3 Consider counting processes N_h, $h > 0$, where at each point kh, $k \in \mathbb{N}$, an arrival occurs with probability λh. Show that N_h converges, as $h \to 0$, to a Poisson process with rate λ.

Exercise 2.4 Consider a Poisson distribution X with parameter 10. Give its expectation and bounds l and u such that $\mathbb{P}(l \leq X \leq u) \approx 0.9$.

a. Do the same for Poisson distributions with parameters 100, 1000, and 10000.
b. Use the law of large numbers to explain your findings.
c. What are the implications when predicting future arrival count to a service facility?

Exercise 2.5 Let X have a Poisson distribution with parameter λ. Let Y have a normal distribition with $\mathbb{E}Y = \mathbb{E}X$ and $\sigma^2(Y) = \sigma^2(X)$.
a. What is $\sigma^2(Y)$?
b. Make an Excel sheet with the values of $\mathbb{P}(Y \leq n) - \mathbb{P}(X \leq n)$ for all relevant values of $n \in \mathbb{N}_0$.
c. Vary Y and determine values of λ for which Y is a good approximation of X.
d. Design and implement an Excel function POISSONINV() that generates random outcome of the Poisson distribution.

Exercise 2.6 The arrival process to the "First Cardiac Aid" department (FCA) of a hospital is modeled as an inhomogeneous Poisson process with the following rate: from 8.00 to 22.00 it is 1.5 per hour, from 22.00 to 8.00 0.5 per hour.
a. What is the expected number of patients per day (from midnight to midnight)?
b. Patients stay exactly 6 hours. Give for each point in time the expected number of patients.
c. What is the distribution of the number of patients at a certain point in time? Motivate your answer.
d. 50% of the patients stay only 4 hours, the other 50% stay 8 hours. What is the distribution and its parameter(s) of the number of patients at 13.00?

Exercise 2.7 Consider the following data with a trend, intra-week seasonality, and an event at day 8 and 14:

$$121, 105, 97, 111, 118, 31, 21, 151, 115, 118, 105, 130, 40, 20, 131, 95, 108, 100, 127, 34, 15$$

a. Apply linear regression to this model with and without Anscombe and log transforms. Calculate the in-sample MSE, MAPE and WAPE.
b. Make a 4-week forecast assuming that the event will occur again on day 30 and 40.

Exercise 2.8 a. Show that the formula for the WAPE is indeed the weighted average percentage error.
b. Consider the following actuals and forecasts for two weeks:

$$116, 101, 102, 107, 102, 8, 11, 116, 120, 105, 82, 102, 6, 8$$

$$100, 100, 100, 100, 100, 10, 10, 100, 100, 100, 100, 100, 10, 10$$

Calculate MSE, RMSE, MAPE, and WAPE.
c. Calculate the unavoidable minimal MAPE and WAPE assuming these numbers are Poissonian. Is it likely that there was a forecasting error?
d. Repeat this exercise with data that you generated, both with and without forecasting error.

Exercise 2.9 This exercise is about quantile regression.
a. Formulate an LP model that minimizes $\sum_{i=1}^{n}[(1-\tau)(a+bx_i-y_i)^+ + \tau(y_i-a-bx_i)^+]$ for given (x_i, y_i), $i = 1, \ldots, n$, and $\tau \in (0,1)$.
b. Let $n = 10$, take $x_i = i$ and $y_i \sim N(i,1)$. Implement the LP model and determine a and b such that $\sum_i |a + bx_i - y_i|$ is minimized.
c. Use standard software to determine a' and b' such that $\sum_i (a' + b'x_i - y_i)^2$ is minimized. Explain the differences.

Chapter 3

Regenerative Processes

In this chapter we discuss stochastic processes, regenerative processes and discrete-event simulation.

3.1 Stochastic processes

A stochastic process is a collection of random variables X_t, $t \geq 0$. A realization or trajectory of the process is a function from $[0, \infty)$ to $\mathbb{R}$ (or $\mathbb{R}^m$). X_t is often called the *state* of the process at time t. It is convenient to introduce the notation $\pi_t(A) = \mathbb{P}(X_t \in A)$ for some set A.

Example 3.1.1 Consider some service system with arrivals and departures. As X_t we can take the number of customers in the system at time t. An alternative would be to take the total workload in the system.

We are interested in calculating performance measures such as $T^{-1}\mathbb{E}\int_0^T f(X_s)ds$ and $\mathbb{E}f(X_T)$. A number of different techniques exist depending on the structure of X_t. Special cases are for example Markov chains, for which special methods exist, and sometimes even closed-form expressions can be derived. A widely used technique that can be used for a very general class of models is *discrete-event simulation*. As can be expected from the name this technique works for models that have discrete events, usually called discrete-event systems.

3.2 Discrete-event simulation

In this section we assume $X_t \subset \mathbb{N}_0^m$. Thus every trajectory is a piece-wise constant function, the process makes discrete jumps. For this reason such a process is called a discrete-event system. Discrete-event simulation is about generating trajectories of discrete-event systems. This is done by starting at time 0 and then constructing a trajectory by sampling one by one events in the system. Usually there is an obvious way to do this.

Example 3.2.1 Consider a service system with a single server, Poisson arrivals and i.i.d. service times. At time 0 the system is empty. In computer memory 4 numbers are stored: the number of customers in the system x, the time of the next arrival a, the time of the next departure b (if $x > 0$), and the current time t. By sampling from the interarrival and service time distributions (using the method of Section 1.9.1) we generate future events. Then the time t is augmented to $\min\{a, b\}$, x is increased of decreased by 1 according to the type of event that attained the minimum, and an interarrival time or service time is sampled (unless a departure leaves an empty queue behind). Now time is increased again, and so forth. This way we simulate a whole trajectory, a realization of the stochastic process.

If the performance measure is of the form $\mathbb{E}f(X_T)$ or $T^{-1}\mathbb{E}\int_0^T f(X_s)ds$ then it suffices to simulate the stochastic process up to T. In this case the ideas from Section 1.9 can be used to derive a confidence interval for the measure we try to estimate. Evidently, we need multiple runs from 0 to T to perform such an analysis. Many simulation packages exist for executing this type of discrete-event simulations.

When we are interested in calculating $T^{-1}\mathbb{E}\int_0^T f(X_s)ds$ for T big then we often see that the trajectories all give approximately the same value: the bigger T, the smaller the variation of the outcome. To explain this phenomenon we need renewal theory, the subject of the next section.

3.3 Renewal theory

Consider some stochastic process X_t, $t \geq 0$. We are interested in calculating

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t f(X_s)ds, \tag{3.1}$$

some long-run average performance measure of this process. Under what conditions does this number exist? Is it a number or a random variable? How to calculate it efficiently? These are the questions that we answer in this section.

In full generality it is impossible to answer the questions we posed above. For this reason we assume the following framework in which the bigger part of the models that we are interested in fit. Assume that there are random variables T_i, with $0 = T_0 \leq T_1 \leq T_2 \leq \cdots$, such that $\{X_t, T_i \leq t \leq T_{i+1}\}$ i.i.d. for all $i \geq 0$. Then the T_i are called renewal points, X_t is a regenerative process and the following theorem holds.

Theorem 3.3.1 *If* $0 < \mathbb{E}(T_1 - T_0) < \infty$ *and* $\mathbb{E}\int_{T_0}^{T_1} |f(X_s)|ds < \infty$ *then*

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t f(X_s)ds = \frac{\mathbb{E}\int_{T_0}^{T_1} f(X_s)ds}{\mathbb{E}(T_1 - T_0)}.$$

Proof Take $f = f^+ - f^-$, $f^+, f^- \geq 0$, and consider $N_t = \max_n\{T_n \leq t\}$. Now

$$\frac{1}{t}\int_0^t f^+(X_s)ds \geq \frac{1}{t}\int_0^{T_{N_t}} f^+(X_s)ds = \frac{N_t}{t}\frac{\sum_{k=0}^{N_t-1}\int_{T_k}^{T_{k+1}} f^+(X_s)ds}{N_t}. \tag{3.2}$$

If $t \to \infty$, then so does N_t. Then, using the Law of Large Numbers, it follows that

$$\sum_{k=0}^{N_t-1}\int_{T_k}^{T_{k+1}} f^+(X_s)ds/N_t \to \mathbb{E}\int_{T_0}^{T_1} f^+(X_s)ds.$$

From the same law it also follows that

$$\frac{T_{N_t}}{N_t} = \frac{\sum_{k=0}^{N_t-1}(T_{k+1} - T_k)}{N_t} \to \mathbb{E}(T_1 - T_0).$$

Similarly we can show that $T_{N_t+1}/N_t \to \mathbb{E}(T_1 - T_0)$. Because

$$\frac{N_t}{T_{N_t+1}} \leq \frac{N_t}{t} \leq \frac{N_t}{T_{N_t}}$$

we find $N_t/t \to 1/\mathbb{E}(T_1 - T_0)$. Taking the limit for $t \to \infty$ in (3.2) leads to

$$\lim_{t\to\infty} \frac{1}{t}\int_0^t f^+(X_s)ds \geq \frac{\mathbb{E}\int_{T_0}^{T_1} f^+(X_s)ds}{\mathbb{E}(T_1 - T_0)}.$$

To obtain the other inequality we use

$$\frac{1}{t}\int_0^{T_{N_t+1}} f^+(X_s)ds \geq \frac{1}{t}\int_0^t f^+(X_s)ds.$$

This gives the result for f^+. The same arguments can be applied to f^-. □

It is important to note that

$$\frac{\mathbb{E}\int_{T_0}^{T_1} f(X_s)ds}{\mathbb{E}(T_1 - T_0)} \neq \mathbb{E}\frac{\int_{T_0}^{T_1} f(X_s)ds}{(T_1 - T_0)}.$$

Example 3.3.2 Consider a process that alternates between the states 0 and 1. The times it stays in 0 are i.i.d., the first length is A. The times the process stays in 1 are also i.i.d., with r.v. S. This is a suitable model for the repair process of a component (see Section 14.5), and when A is exponentially distributed then it is equivalent to the $M|G|1|1$ queue (see Section 5.1 for this notation). Assume that $0 < \mathbb{E}A + \mathbb{E}S < \infty$. The long-run fraction of time that the system is in state 1 is given by $\mathbb{E}S/(\mathbb{E}A + \mathbb{E}S)$. This follows from Theorem 3.3.1, with $f(x) = \mathbb{I}\{x = 1\}$.

It should also be observed that $\lim_{t\to\infty} \frac{1}{t}\int_0^t f(X_s)ds$ is a number, not a random variable. This explains the convergence of long replications to the same outcome.

3.4 Simulating long-run averages

Now suppose we want to calculate (3.1) for a regenerative process for which we have no closed-form expression for $\mathbb{E}\int_{T_0}^{T_1} f(X_s)ds$ and/or $\mathbb{E}(T_1 - T_0)$. Renewal theory suggests simulating repeatedly busy periods and estimating the performance from that. Unfortunately, most simulation software tools are not well fit to do this. On the other hand, running a system indefinitely is also impossible: it would require an infinite running time. The solution is as follows. Instead of simulating X_t from 0 to ∞ and measuring $\mathbb{E}\int_0^\infty f(X_s)ds$ we simulate X_t from 0 to t_1 and we measure $\mathbb{E}\int_{t_0}^{t_1} f(X_s)ds$ for well-chosen constants t_0 and t_1, $0 < t_0 < t_1$. The choice of t_0 is of particular importance. Due to the start-up of the simulation the stochastic process does not show long-run behavior for small t. These *transient* effects disappear in the long-run average. However, when we simulate up to a constant t_1, this transient effect plays a role. By choosing t_0 large enough this should be avoided as much as possible, without taking t_0 too large to avoid spending too much of our computing time on simulating the warming-up period. The samples of $\mathbb{E}\int_{t_0}^{t_1} f(X_s)ds$ can be analyzed in the usual way.

Example 3.4.1 We simulated the system of Example 3.2.1 with arrival rate 1 and exponential service times with mean 0.8. The system starts initially empty. When simulating 2000 runs from 0 to 100 the average number of customers in the system was for our particular run 3.33.

Simulating 1000 runs from 0 to 200 gives 3.64. Simulating 1000 runs from 50 to 250 finally gives 3.89. Thus we see an increase in average value as we increase the simulation length and as we introduce a warm-up period. Queueing theory (Theorem 5.3.1) tells us that the long-run average number of customers is exactly 4.

3.5 The long-run average and limiting distributions

Taking $f(X_t) = \mathbb{I}\{X_t \in A\}$ for some set A is an interesting special case of Theorem 3.3.1, because

$$\bar{\pi}_\infty(A) = \lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{I}\{X_s \in A\} ds$$

can be interpreted as the long-run fraction of time that the system is in the set A. Evidently

$$0 \leq \mathbb{E} \int_{T_0}^{T_1} \mathbb{I}\{X_s \in A\} ds \leq \mathbb{E} \int_{T_0}^{T_1} ds = \mathbb{E}(T_1 - T_0),$$

thus the long-run average distribution exists and is unique if $0 < \mathbb{E}(T_1 - T_0) < \infty$.

Sometimes we are not interested in the long-run average distribution, but in the limiting distribution

$$\pi_\infty(A) = \lim_{t\to\infty} \pi_t(A) = \lim_{t\to\infty} \mathbb{P}(X_t \in A) = \lim_{t\to\infty} \mathbb{E}\mathbb{I}\{X_t \in A\}.$$

Unfortunately, this distribution does not always exist. For example, take a process that alternates between state 0 and 1 and stays in each state exactly 1 time unit. Then $\lim_{t\to\infty} \frac{1}{t} \int_0^t \mathbb{I}\{X_s = 1\} ds = 0.5$ (see Example 3.3.2), but the limiting probability $\pi_\infty(1)$ does not exist, because $\pi_t(1)$ alternates between 0 and 1 and does not converge.

We need a condition on the cycle length distribution $T_1 - T_0$ to make π_t converge. We assume that $T_1 - T_0$ is *nonlattice*, which means that $T_1 - T_0$ is not concentrated on a set of the form $\{\delta, 2\delta, \ldots\}$.

Theorem 3.5.1 *Consider a regenerative process X_t with $0 < \mathbb{E}(T_1 - T_0) < \infty$ and $T_1 - T_0$ nonlattice. Then $\pi_\infty(A)$ exists for all A and is given by*

$$\pi_\infty(A) = \lim_{t\to\infty} \mathbb{P}(X_t \in A) = \frac{\mathbb{E} \int_{T_0}^{T_1} \mathbb{I}\{X_s \in A\} ds}{\mathbb{E}(T_1 - T_0)}.$$

Thus under the nonlattice condition the time-average distribution and the limiting distribution are equal. For a proof of Theorem 3.5.1 we refer to Asmussen [10, Theorem VI.1.2].

3.6 Poisson arrivals see time averages

In Part III we often consider systems in which customers arrive according to a Poisson process. For this reason it is important to ask ourselves the question: how do arriving customers perceive the system? Consider a regenerative process X_t and a Poisson process $N(t)$. X_t models some process for which $N(t)$ is the arrival process. For this reason, $X(s)$, $s \leq t$, is a function of the points of $N(s)$ with $s \leq t$. The points of N after t do not influence X before t. Thanks to the second property of Definition 2.2.1 $N(t)$ is independent of $N(t, \infty)$. Combining all gives that $\{X_s, s \leq t\}$ is independent of $N(t, \infty)$.

Let Y_t be equal to X_{t^-}, conditioned on the fact that an arrival occurs at t (with t^- we mean the moment just before the arrival at t):

$$\mathbb{P}(Y_t \in A) = \lim_{h \to 0} \mathbb{P}(X_{t-h} \in A | N(t-h, t) = 1).$$

Define $\bar{\pi}_t$ by

$$\bar{\pi}_t(A) = \frac{1}{t} \int_0^t \mathbb{P}(X_s \in A) ds.$$

Note that for $\bar{\pi}_\infty$, as defined in the previous section, holds:

$$\bar{\pi}_\infty(A) = \lim_{t \to \infty} \bar{\pi}_t(A),$$

We also define $\bar{\alpha}_t$ and $\bar{\alpha}_\infty$ by

$$\bar{\alpha}_t(A) = \frac{1}{t} \int_0^t \mathbb{P}(Y_s \in A) ds \text{ and } \bar{\alpha}_\infty(A) = \lim_{t \to \infty} \bar{\alpha}_t(A).$$

The limiting average distribution $\bar{\pi}_\infty$ exists under the conditions of Theorem 3.3.1. The values of $\bar{\alpha}_t$ and $\bar{\alpha}_\infty$ follow from the following theorem.

Theorem 3.6.1 *In the case of Poisson arrivals $\bar{\alpha}_t = \bar{\pi}_t$ and $\bar{\alpha}_\infty = \bar{\pi}_\infty$, thus Poisson arrivals see time averages (generally known as the* PASTA *property).*

Proof The random variables $N(t-h)$ and $N(t-h, t)$ are independent, and therefore so are X_{t-h} and $N(t-h, t)$. Thus $\mathbb{P}(Y_t \in A) = \lim_{h \to 0} \mathbb{P}(X_{t-h} \in A | N(t-h, t) = 1) = \lim_{h \to 0} \mathbb{P}(X_{t-h} \in A) = \mathbb{P}(X_t \in A)$. Integrating, averaging, and taking the limit as $t \to \infty$ gives the required results. □

This result is very convenient when analyzing all kinds of systems with Poisson arrivals. When analyzing certain properties of customers (such as the waiting time upon arrival) we can base ourselves on the long-run average distribution: PASTA tells us that the time-averages are equal to the customer-averages. Note that (under the nonlattice condition) Theorem 3.5.1 tells us that the time-average distribution is equal to the limiting distribution. Thus an arbitrary arrival also sees the limiting distribution.

Example 3.6.2 Consider a service facility with a single server, Poisson arrivals, and customers leave when the server is occupied. The occupancy of the server can be analyzed using the model of Example 3.3.2, with $A \sim \exp(\lambda)$. We have

$$\mathbb{P}(\text{arbitrary arriving customer blocked}) = \bar{\alpha}_\infty(1) = \bar{\pi}_\infty(1) = \frac{\mathbb{E}S}{\lambda^{-1} + \mathbb{E}S}.$$

To show that Poisson arrivals are crucial to obtain $\bar{\alpha}_\infty = \bar{\pi}_\infty$, we consider the same model but with fixed interarrival times b and service times s, with $b > s > 0$. Then

$$\mathbb{P}(\text{arbitrary arriving customer blocked}) = 0 = \bar{\alpha}_\infty(1) \neq \bar{\pi}_\infty(1) = \frac{s}{b}.$$

3.7 The waiting-time paradox

Consider a regenerative process X_t. We are interested in the expected time until the next renewal point for an arbitrary outside observer. For a regenerative process on $[0, t]$, we define an outside observer as someone who observes the state of the system at an arbitrary instant ξ in $[0, t]$, that is, $\xi \sim \text{Uniform}[0, t]$. But arrivals in a Poisson process are, if they are not ordered, distributed as independent uniform distributions, according to Exercise 2.2. Thus the outside observer behaves as a Poisson arrival, and thus Theorem 3.6.1 applies also to the outside observer, who sees as a result time-average behavior.

In fact, an outside observer is a special case of the situation of Section 3.6: X_t and $N(s)$ are independent for all t and s. Thus, the expected time until the next renewal point for our outside observer is equal to the long-run average time until the next renewal.

Naively one could think that this number is equal to $\mathbb{E}S/2$ with $S \sim T_1 - T_0$, but this is only true for S deterministic. Let T_i be the time of the ith renewal. Define the process X_t as the time until the next renewal. Then we have $X_t = T_i - t$ for $T_{i-1} < t \leq T_i$.

Now X_t is indeed a regenerative process, with T_i the ith moment that X_t gets 0. A typical realization of X_t is given in Figure 3.1. The upward jumps are distributed according to S.

Figure 3.1: A typical realization of X_t.

Using Theorem 3.3.1 we find that

$$\lim_{t\to\infty}\frac{1}{t}\int_0^t X_s ds = \frac{\mathbb{E}\int_{T_0}^{T_1} X_s ds}{\mathbb{E}(T_1-T_0)} = \frac{\mathbb{E}\int_0^S s\,ds}{\mathbb{E}S} = \frac{\int\int_0^t s\,ds\,dF_S(t)}{\mathbb{E}S} = \frac{\mathbb{E}S^2}{2\mathbb{E}S}.$$

The fact that $\mathbb{E}\int_{T_0}^{T_1} X_s ds = \mathbb{E}S^2/2$ can also be seen directly: it is the surface of a right-angled triangle with legs S.

Indeed, we see that the waiting time $\mathbb{E}S^2/(2\mathbb{E}S) = \mathbb{E}S/2$ if and only if S is deterministic. For example, if S is exponential, then $\mathbb{E}S^2/(2\mathbb{E}S) = \mathbb{E}S$.

This result is often used to explain the fact that when we go at an arbitrary moment to the bus stop we experience an expected waiting time longer than half the expected interarrival time. For this reason this result is known under the name *waiting-time paradox*. Note that it is assumed that the interarrival times are independent, which is a somewhat unrealistic assumption.

Note also that for the current model

$$\mathbb{E}\frac{\int_{T_0}^{T_1} f(X_s)ds}{(T_1-T_0)} = \mathbb{E}S.$$

This nicely illustrates what happens if we calculate the expectation of the quotient instead of the quotient of the expectations.

This waiting-time paradox is also known under the name *inspection paradox*.

3.8 Cost equations and Little's Law

The previous sections dealt with the limiting state distribution perceived by an arriving customer and an outside observer such as the manager of the system. There is another set of interesting relations linking the views of customers and system managers, called *cost equations*.

The idea is as follows. Consider a process X_t and customers that arrive during a period $[0, T]$ according to some process $N(t)$, and leave according to another process $M(t)$. Note that $N(t)$ is not necessarily a Poisson process, and that the system is empty at t if $N(t) = M(t)$. Let A_k be the arrival time of the kth arriving customer, and D_k its departure time. Of course $0 \leq A_1 \leq A_2 \leq \cdots$, but the D_k need not be ordered. Evidently $A_k \leq D_k$. Each customer incurs costs while in the system, $C_k(t)$ for customer k at t. We assume $C_k(t) = 0$ if $t \leq A_k$ or $t \geq D_k$. Now there are two ways to calculate the expected costs: we can look at the expected amount the system receives, or at the expected number of arrivals times the average cost per customer. These must be equal. To formalize this, define

$$\lambda = \lim_{T\to\infty} N(T)/T,\ H(t) = \sum_{k=1}^{\infty} C_k(t), \text{ and } G_k = \int_0^{\infty} C_k(t)dt.$$

In the next theorem we will assume that arrivals and departures are functions of X_t, that is, from the evolution of X_t the processes $N(t)$ and $M(t)$ can be constructed. Then the renewals of X_t are also renewals of $N(t)$ and $M(t)$.

Theorem 3.8.1 *For every realization of X_t for which $N(T) = M(T)$ we have*

$$\int_0^T H(t)dt = \sum_{k=1}^{N(T)} G_k;$$

if X_t is a regenerative process with $0 < \mathbb{E}(T_1 - T_0) < \infty$ and $N(T_1) = M(T_1)$, then

$$\lim_{T\to\infty} \frac{\int_0^T H(t)dt}{T} = \lambda \lim_{n\to\infty} \frac{\sum_{k=1}^n G_k}{n}.$$

Proof The first statement follows from the fact that the system is empty at T, and thus $C_k(t) = 0$ for $t > T$ and $k \leq N(T)$:

$$\int_0^T \sum_{k=1}^{\infty} C_k(t)dt = \int_0^T \sum_{k=1}^{N(T)} C_k(t)dt = \sum_{k=1}^{N(T)} \int_0^T C_k(t)dt = \sum_{k=1}^{N(T)} \int_0^{\infty} C_k(t)dt.$$

The proof of the second statement is similar to that of Theorem 3.3.1. Consider $C_k^+(t) = \max\{C_k(t), 0\}$, and define $H^+(t)$ and G_k^+ accordingly. Then it is readily seen that for all T

$$\sum_{k=1}^{M(T)} G_k^+ \leq \int_0^T H^+(t)dt \leq \sum_{k=1}^{N(T)} G_k^+.$$

This is equal to

$$\frac{M(T)}{T}\frac{1}{M(T)}\sum_{k=1}^{M(T)} G_k^+ \leq \frac{1}{T}\int_0^T H^+(t)dt \leq \frac{N(T)}{T}\frac{1}{N(T)}\sum_{k=1}^{N(T)} G_k^+.$$

From the fact that X_t is regenerative it follows that $\lim_{T\to\infty} M(T)/T = \lim_{T\to\infty} N(T)/T = \lambda$ and that $N(T)$ and $M(T) \to \infty$ as $T \to \infty$. Hence

$$\lim_{T\to\infty}\frac{\int_0^T H^+(t)dt}{T} = \lim_{T\to\infty}\frac{N(T)}{T}\frac{\sum_{k=1}^{N(T)} G_k^+}{N(T)} = \lambda\lim_{n\to\infty}\frac{\sum_{k=1}^{n} G_k^+}{n},$$

for positive costs. A similar argument applies to $C_k^-(t) = \max\{-C_k(t), 0\}$. Summing gives the required result. □

The best known cost equation is Little's Law. Consider a customer who pays one unit for every time unit it stays in a system, thus $C_k(t) = 1$ for $A_k \leq t \leq D_k$. Then we have, with w the average time a customer spends in a system and l the long-run average number of customers in the system:

$$l = \lambda w. \tag{3.3}$$

Another useful inequality is that of a single-server system where every customer pays 1 unit when in service. Then, with S the service-time distribution:

$$\text{long-run average fraction of time server busy} = \lambda\mathbb{E}S. \tag{3.4}$$

3.9 Further reading

Çinlar [39] is the standard reference for stochastic processes. Renewal theory is extensively discussed in Ross [130] and Tijms [152]. A more technical reference is Asmussen [10].

There are many excellent books dedicated to simulation. We name Nelson [119], Ross [129], Rubinstein [134], Kleijnen [91], Law & Kelton [103], and Kelton et al. [87]. The latter is at the same time an introduction to the simulation package Arena.

El-Taha & Stidham [55] is completely dedicated to cost equations; the approach in Section 3.8 is inspired by [55, Section 6.4], especially Theorem 6.8.

3.10 Exercises

Exercise 3.1 Consider Example 3.6.2 with Poisson arrivals and general i.i.d. service times.
a. Give renewal points of this process.
Now assume that the interarrival times are not exponentially distributed anymore, but i.i.d.
b. Give again renewal points.

Exercise 3.2 A person takes a bus each morning to go to work. If she catches the bus within t minutes after arriving at the bus stop she gets to work on time, otherwise she is late. Busses arrive according to a renewal process (see Section 2.6) with interarrival distribution S.
a. Give an expression for the probability that she arrives on time.
b. Calculate this for $t = 10$ and S exponentially distributed with average 8 minutes.

Exercise 3.3 In call centers, when the shift of an employee ends, it is customary that they finish the call they are currently working on. Call durations are approximately lognormal.
a. Find a method to calculate the average amount of work employees are doing after the end of the working time. Numerical calculation or simulation can be used.
b. Approximate this time for call durations with an average and standard deviation of both 3 minutes.

Exercise 3.4 Consider a model with Poisson arrivals, s servers, constant service times, and arrivals that find all servers busy are lost (the $M|G|s|s$ system).
a. Simulate this model using Arena for some well-chosen parameters and give an estimate for the blocking probability.
b. Motivate your choice of the number of replications and the lengths of the simulation and the warming-up period.
c. Give a confidence interval for the blocking probability.

Exercise 3.5 An emergency department in a hospital has capacity for 2 trauma patients. When the capacity is reached new patients are brought to another hospital. Arrivals occur according to a Poisson process, on average 4.2 per day are accepted. 16% of the time both beds are occupied.
a. What was the demand? Which percentage is sent elsewhere?
In reality there is a daily pattern in the arrivals and sometimes arrivals occur in small groups (trauma patients are often the result of traffic accidents).
b. What do you think that the influence of these facts will be on the outcomes?

Chapter 4

Markov Chains

Markov chains are the most often used class of stochastic processes. They are also fundamental to the study of queueing models.

There are several types of Markov chains between which we have to distinguish. A first distinction is between continuous and discrete time. We concentrate on continuous-time Markov chains, because most of the applications we consider evolve in continuous time. However, discrete-time Markov chains are conceptually simpler, therefore we pay attention to them first.

4.1 Discrete-time Markov chains

A discrete-time Markov chain is a special type of stochastic process X_t. This process takes values in some finite or countable state space $\mathcal{X}$. Often we take $\mathcal{X} = \{0, \ldots, n\}$ or $\{0, 1, \ldots\}$. A discrete-time Markov chain changes state only at the time instants $\{1, 2, \ldots\}$, according to the following rule: when at $t \in \mathbb{N}$ the state $X_t = x$, then $X_{t+1} = y$ with probability $p(x, y)$, independent of the states visited before t. Thus $\mathbb{P}(X_{t+1} = y | X_t = x) = p(x, y)$. These probabilities are called the *one-step transition probabilities*.

This system is easy to simulate. We simply keep in memory the current state x, and generate the next state according to the distribution $p(x, \cdot)$. This way we can generate trajectories of the process X_t. But we can do better if $|\mathcal{X}| < \infty$: in this case we can calculate the distribution of X_t for each $t \in \mathbb{N}$. As in Section 3.5, we write $\pi_t(x) = \mathbb{P}(X_t = x)$. Then

$$\pi_{t+1}(y) = \sum_{x \in \mathcal{X}} \pi_t(x) p(x, y), \tag{4.1}$$

or, in matrix notation, $\pi'_{t+1} = \pi'_t P$ (with u' the transpose of u and the matrix P with entries $p(x,y)$). Recursively, this amounts to $\pi'_t = \pi'_0 P^t$. Thus computing π_t only requires a number of matrix multiplications. From these distributions we can compute performance measures such as $\mathbb{E}f(X_T)$ and $T^{-1}\mathbb{E}\sum_{t=1}^{T} f(X_t)$, the discrete-time equivalent of (3.1).

Next we study the behavior for the case $T \to \infty$. Define (similar to the definitions in Section 3.5)

$$\pi_\infty(x) = \lim_{t\to\infty} \pi_t(x)$$

and

$$\bar{\pi}_\infty(x) = \lim_{t\to\infty} \bar{\pi}_t(x) = \lim_{t\to\infty} \frac{1}{t}\mathbb{E}\sum_{s=1}^{t} \mathbb{I}(X_s = x).$$

We will apply Theorem 3.3.1. The renewal moments in our Markov chain are the instants at which we reach a certain state x^*. A necessary condition for $\mathbb{E}(T_1 - T_0) < \infty$ is that from x^* we can always go back to x^* in a finite number of transitions. An even stronger assumption, but which is usually satisfied in practical applications, is the following.

Assumption 4.1.1 *Every state can be reached from every other state, that is, for arbitrary $x, y \in \mathcal{X}$ there is a path with positive probability, which means that there are $x_1, \ldots, x_k$ with $x = x_0$ and $y = x_k$ such that $p(x_0, x_1) \cdots p(x_{k-1}, x_k) > 0$.*

Thus every state can now serve as renewal point. Let T_x be the time, starting in x, to return to x. It follows immediately from Theorem 3.3.1 that $\bar{\pi}_\infty(x) = [\mathbb{E}T_x]^{-1}$ if $\mathbb{E}T_x < \infty$. It can be shown that if Assumption 4.1.1 holds and $|\mathcal{X}| < \infty$ then $\mathbb{E}T_x < \infty$. If $|\mathcal{X}| = \infty$ then it can also occur that $\mathbb{E}T_x = \infty$. Note however that also in this situation $\bar{\pi}_\infty(x) = [\mathbb{E}T_x]^{-1}$ holds.

For the computation of the actual value of $\bar{\pi}_\infty$ we cannot use renewal theory. We derive directly an equation for the value of $\bar{\pi}_\infty$. The expected number of times up to t that the chain moves from x to y is given by $t\bar{\pi}_t(x)p(x,y)$. Consider some set $\mathcal{Y} \subset \mathcal{X}$. The difference between the number of times that the chain goes out of $\mathcal{Y}$ or into $\mathcal{Y}$ is at most 1. Equating the flow in and out of $\mathcal{Y}$, dividing by t and taking $t \to \infty$ leads to the following theorem.

Theorem 4.1.2 *The numbers $\bar{\pi}_\infty(x)$ must satisfy*

$$\sum_{x\in\mathcal{Y}} \bar{\pi}_\infty(x) \sum_{y\in\mathcal{Y}^c} p(x,y) = \sum_{x\in\mathcal{Y}^c} \bar{\pi}_\infty(x) \sum_{y\in\mathcal{Y}} p(x,y) \tag{4.2}$$

for all $\mathcal{Y} \subset \mathcal{X}$ *and*

$$\sum_{x \in \mathcal{X}} \bar{\pi}_\infty(x) = 1. \tag{4.3}$$

If we take $|\mathcal{Y}| = 1$, then we get a system of equations which is known as the *equilibrium equations* (take $\mathcal{Y} = x$ and add $\bar{\pi}_\infty(x)p(x,x)$ to both sides):

$$\bar{\pi}_\infty(x) = \sum_{y \in \mathcal{X}} \bar{\pi}_\infty(y)p(y,x). \tag{4.4}$$

In certain situations however other choices of $\mathcal{Y}$ can be useful. Note also that if we find a solution of (4.4), then it is a solution for all possible $\mathcal{Y}$, by summing (4.4) over all $x \in \mathcal{Y}$.

Markov chain theory tells us that, under Assumption 4.1.1, Equations (4.2)-(4.3) have a unique solution if and only if $\mathbb{E}T_x < \infty$ for all x. In fact, either $\mathbb{E}T_x < \infty$ for all x or $\mathbb{E}T_x = \infty$ for all x. The latter case can only occur if $|\mathcal{X}| = \infty$.

To obtain the existence of π_∞ we need an extra condition, the equivalent of the nonlattice condition that was formulated in Section 3.5. This condition is simply that the greatest common divisor of all paths from x^* to x^* should be 1.

In Markov chains, the distribution $\bar{\pi}_\infty$ has another interesting interpretation. Assume that $\pi_0 = \bar{\pi}_\infty$. Then, according to Equation (4.4), $\pi_1 = \bar{\pi}_\infty$, and recursively, $\pi_t = \bar{\pi}_\infty$ for all t. For this reason we call $\bar{\pi}_\infty$ also the *stationary distribution*.

4.2 Continuous-time Markov chains

Let us define continuous-time Markov chains. They are again defined on some finite or countable set $\mathcal{X}$, the state space. The time that this process stays in a state is exponentially distributed, with parameter $\Lambda(x)$ in state x. When this time expires a transition is made to a new state y with probability $p(x,y)$. Once in y this process starts again.

Define $\lambda(x,y) = p(x,y)\Lambda(x)$. We can see $\lambda(x,y)$ as the rate at which the system moves from x to y. The state changes according to the first transition to occur. Then the time until the first transition is indeed exponentially distributed with parameter $\Lambda(x) = \sum_{y \in \mathcal{X}} \lambda(x,y)$, because the minimum of a number of exponential random variables is again exponential, and the probability of a transition to y is equal to $p(x,y) = \lambda(x,y)/\Lambda(x)$ (see Section 1.6.5).

A continuous-time Markov chain is easy to simulate: in x, one samples from an exponentially distributed random variable to determine the next transition epoch,

and the next state is determined according to the distributed $p(x,\cdot)$. An alternative method is sampling from an exponential distribution for each of the possible transitions, using the rates $\lambda(x,y)$. The latter choice is sometimes more intuitive.

Example 4.2.1 Consider a queueing system with a single server, Poisson arrivals with rate λ and exponential service time distributions, rate μ. Then $\Lambda(x) = \lambda + \mu$ if $x > 0$, $\Lambda(0) = \lambda$. Also $p(0,1) = 1$, and for $x > 0$ $p(x,x+1) = \lambda/(\lambda+\mu)$ and $p(x,x-1) = \mu/(\lambda+\mu)$. Thus we can sample first the time in each state and then the transition. It is more intuitive, with the actual system in mind, to sample the interarrival and service time distributions, using $\lambda(x,x+1) = \lambda$ and $\lambda(x,x-1) = \mu$ (the latter for $x > 0$).

The *distribution* of X_t is harder to determine than in the deterministic case; we will discuss this in Section 4.6, in the context of time-inhomogeneous chains. Here we are interested in the behavior of the Markov chain as the time goes to ∞. To be able to apply renewal theory we make the following assumption.

Assumption 4.2.2 *Every state can be reached from every other state, that is, for arbitrary $x, y \in \mathcal{X}$ there are $x_1, \ldots, x_k$ with $x = x_0$ and $y = x_k$ such that $\lambda(x_0,x_1)\cdots\lambda(x_{k-1},x_k) > 0$.*

Define $\pi_t(x)$, $\pi_\infty(x)$, $\bar{\pi}_t(x)$, and $\bar{\pi}_\infty(x)$ as in Section 3.5. The nonlattice condition is satisfied because exponential distributions are nonlattice. Therefore Theorem 3.5.1 can be applied and we find $\pi_\infty(x) = \bar{\pi}_\infty(x)$.

Let us for the moment delay questions about the existence of this limit. The expected number of times up to t that the chain moves from x to y is given by $\bar{\pi}_t(x)\lambda(x,y)$. Then, equivalent to Theorem 4.1.2, we get:

Theorem 4.2.3 *The numbers $\pi_\infty(x)$ must satisfy*

$$\sum_{x\in\mathcal{Y}} \pi_\infty(x) \sum_{y\in\mathcal{Y}^c} \lambda(x,y) = \sum_{x\in\mathcal{Y}^c} \pi_\infty(x) \sum_{y\in\mathcal{Y}} \lambda(x,y) \tag{4.5}$$

for all $\mathcal{Y} \subset \mathcal{X}$ and

$$\sum_{x\in\mathcal{X}} \pi_\infty(x) = 1. \tag{4.6}$$

Under Assumption 4.2.2 the following holds.

Theorem 4.2.4 *If $|\mathcal{X}| < \infty$, then a unique solution of (4.5)-(4.6) exists;*
if $|\mathcal{X}| = \infty$, then either a unique solution of (4.5)-(4.6) exists, or $\pi(x) = 0$ for all $x \in \mathcal{X}$ is the unique solution to (4.5) with $|\sum_{x\in\mathcal{X}} \pi(x)| < \infty$.

Consider X_{t+h} for h small. Then we find, using the interpretation of $\lambda(x,y)$ as the hazard rate of going from x to y (using Equation (1.13)):

$$\pi_{t+h}(x) = \sum_{y\in\mathcal{X}} \pi_t(y)[\lambda(y,x)h + o(h)] + \pi_t(x)[1 - \Lambda(x)h + o(h)]. \tag{4.7}$$

Now assume that $\pi_t = \pi_\infty$ with π_∞ a solution of Equation (4.5). Plugging this in gives $\pi_{t+h}(x) = \pi_t(x) + o(h)$, and therefore $\frac{d}{dt}\pi_t(x) = 0$. Thus if $X_t \sim \pi_\infty$ for some t, then $X_s \sim \pi_\infty$ for all $s > t$. Thus π_∞ is also the *stationary distribution*, equivalent to what we found for the discrete-time case.

4.3 Birth-death processes

A special class of continuous-time Markov chains are those where $\mathcal{X} = \{0,\dots,n\}$ or $\{0,1,\dots\}$, and the only non-zero transition rates are $\lambda(x,x+1)$ for all $x < n$ (with n possibly ∞) and $\lambda(x,x-1)$ for all $x > 0$. In fact, we assume that $\lambda(x,x-1) > 0$ for all $x \in \mathcal{X}/\{0\}$, thus state 0 can be reached from any state. We also assume that $\lambda(x,x+1) > 0$ for all $x \in \mathcal{X}/\{n\}$ (or $x \in \mathcal{X}$ if $\mathcal{X}$ is countable), thus every two states are *communicating*, i.e., Assumption 4.2.2 holds.

Such a chain where in one step only neighboring states can be reached is called a *birth-death process*. Birth-death processes are easy to solve. To do so, define $\lambda_x = \lambda(x,x+1)$ and $\mu_x = \lambda(x,x-1)$. If we take, for $x > 0$, $Y = \{0,\dots,x-1\}$ in (4.5), then we find $\pi_\infty(x-1)\lambda_{x-1} = \pi_\infty(x)\mu_x$, and thus

$$\pi_\infty(x) = \frac{\lambda_{x-1}}{\mu_x}\pi_\infty(x-1) = \frac{\lambda_0\cdots\lambda_{x-1}}{\mu_1\cdots\mu_x}\pi_\infty(0). \tag{4.8}$$

All that remains is determining $\pi_\infty(0)$. This can be done using (4.6). Indeed,

$$\pi_\infty(0) = \Big[1 + \sum_{x=1}^{n} \frac{\lambda_0\cdots\lambda_{x-1}}{\mu_1\cdots\mu_x}\Big]^{-1}, \tag{4.9}$$

with $n = \infty$ if $|\mathcal{X}| = \infty$. Note that the convergence of the sum decides whether or not a stationary distribution exists. In many cases (often queueing models) there are explicit solutions for $\pi_\infty(x)$. We give one example.

Example 4.3.1 The system with $\lambda_x = \lambda$ and $\mu_x = \mu$, $|\mathcal{X}| = \infty$, is called the $M|M|1$ queue (for further details, see Chapter 5). Define $\rho = \lambda/\mu$. Then $\pi_\infty(x) = \rho^x\pi_\infty(0)$, by (4.8). The

probability $\pi_\infty(0)$ is given by $\pi_\infty(0) = [\sum_{x=0}^{\infty} \rho^x]^{-1}$. The sum converges if and only if $\rho < 1$. Under this condition a stationary distribution exists with $\pi_\infty(x) = (1-\rho)\rho^x$; if $\rho \geq 1$ then the Markov chain has no stationary distribution. In queueing theory we say that the queue is *unstable* in this situation. Indeed, if we interpret λ as the arrival rate of customers, and μ as the service rate of the server, then $\lambda < \mu$ signifies that there are less arrivals than the server can handle, and thus the queue will always empty after some finite time. If $\lambda \geq \mu$ this is not the case. (In fact, if $\lambda = \mu$ then it takes on average an infinite amount of time to reach state 0; if $\lambda > \mu$ state 0 might not be reached again at all.) See also Exercise 4.2 for the discrete-time equivalent.

4.4 The Markov property

In Section 4.2 we defined a continuous-time Markov chain or Markov process by its state space $\mathcal{X}$ and its transition rates $\lambda(x,y)$. However, starting from a practical problem, it is not always clear how to choose $\mathcal{X}$ and λ. A characterization of Markov processes that helps us understand better what a Markov process actually is, is the so-called *Markov property*. We say that the process X_t satisfies the Markov property if for all $t_1 < \cdots < t_n$ the following holds:

$$\mathbb{P}(X_{t_n} = x_n | X_{t_1} = x_1, \ldots, X_{t_{n-1}} = x_{n-1}) = \mathbb{P}(X_{t_n} = x_n | X_{t_{n-1}} = x_{n-1}).$$

In words: the evolution of a Markov process from some point in time t_{n-1} on does not depend on the history but only on the current state $X_{t_{n-1}}$. It can be seen as a memoryless property.

It is easily seen that the Markov property holds for a given Markov process, due to the transition machanism and the memoryless property of the exponential distribution (see Section 1.6.5). This helps us modeling a system as a Markov process: we should choose states and transition rates such that the future behavior depends only on the current state.

Example 4.4.1 Consider a service center with a single server and a queue for which we are interested in the use of the server. We could take $\mathcal{X} = \{0,1\}$, corresponding to the states of the server, but this information is not enough to describe future behavior: we should also know the length of the queue. Thus $\mathcal{X} = \mathbb{N}_0$ is a good choice, representing the number of customers in the system.

4.5 Beyond PASTA

In Section 3.6 we showed that 'Poisson arrivals see time averages' (PASTA) in general renewal processes. This concept evidently holds also for the special case of continuous-time Markov chains. In the case of Markov chains we can go a step further: we can derive the distribution perceived by arrivals (or other events) that do not occur according to a Poisson process. For this, define (as in Section 3.6) the distribution $\alpha_t(x)$ of Y_t, the state of the Markov chain, conditioned on the fact that an arrival occured at t, just before the arrival:

$$\alpha_t(x) = \mathbb{P}(Y_t = x) = \lim_{h \to 0} \mathbb{P}(X_{t-h} = x | N(t-h, t) = 1).$$

We assume that there are two types of transition rates: $\lambda(x, y) = \lambda'(x, y) + \lambda''(x, y)$ with $\lambda'(x, y)$ representing the arrivals and $\lambda''(x, y)$ the other transitions. Define $\Lambda'(x) = \sum_y \lambda'(x, y)$, and let N now represent event of the λ'-type. $\alpha_t(x)$ can be seen as the fraction of arrivals that occurs in x. Thus $\alpha_t(x)$ is given by:

$$\alpha_t(x) = \lim_{h \to 0} \mathbb{P}(X_{t-h} = x | N(t-h, t] = 1) = \lim_{h \to 0} \frac{\mathbb{P}(X_{t-h} = x, N(t-h, t] = 1)}{\mathbb{P}(N(t-h, t] = 1)} =$$

$$\lim_{h \to 0} \frac{\pi_{t-h}(x)\Lambda'(x)h + o(h)}{\sum_{y \in \mathcal{X}} \pi_{t-h}(y)\Lambda'(y)h + o(h)} = \frac{\pi_t(x)\Lambda'(x)}{\sum_{y \in \mathcal{X}} \pi_t(y)\Lambda'(y)}.$$

If $\Lambda'(x)$ is constant then we find $\alpha_t(x) = \pi_t(x)$ as was expected from Theorem 3.6.1. Taking the limit for $t \to \infty$ gives $\alpha_\infty(x) = \pi_\infty(x)$.

4.6 Time-inhomogeneous chains

In this section we consider time-inhomogeneous Markov chains. For these chains we derive methods to compute the distribution of the process at T. A special case is the distribution of X_T for the time-homogeneous continuous-time Markov chain, which we delayed until this section.

Deriving results for discrete-time chains is easy. We make the chain time-inhomogeneous by letting P depend on t, thus P_t is the transition matrix at t. Now we get, following Section 4.1, $\pi'_{t+1} = \pi'_t P_{t+1}$, and recursively $\pi'_t = \pi'_0 P_1 \cdots P_t$.

Continuous-time chains are more challenging. We assume that the transition rates are piecewise constant. Now consider a time interval of length T for which the rates are constant. For simplicity we assume that the interval starts at 0. We will construct

a method to approximate π_T starting from π_0. The only additional condition we need is that $\Lambda(x)$ is uniformly bounded. Note that this is always the case if $|\mathcal{X}| < \infty$.

Choose a constant Λ such that $\Lambda(x) \leq \Lambda$ for all $x \in \mathcal{X}$. Now add the possibility of "dummy" transitions from a state to itself. Take $\lambda(x,x) = \Lambda - \sum_{y \neq x} \lambda(x,y)$. Then the time between each two jumps is exponentially distributed with parameter Λ, independent of the current state. The total number of jumps between 0 and T has a Poisson distribution with rate ΛT. Conditioned on the number of jumps the transition process is a discrete-time Markov chain with transition probabilities $\hat{p}(x,y) = \lambda(x,y)/\Lambda$ for all y including x. Thus, if the number of jumps in $[0,T]$ is k, then the distribution at T will be $\pi_0' \hat{P}^k$. This leads to the following formula:

$$\pi_T' = \sum_{k=0}^{\infty} \frac{(\Lambda T)^k}{k!} \mathrm{e}^{-\Lambda T} \pi_0' \hat{P}^k. \tag{4.10}$$

If we want to use this in practice, we have to limit the summation to some upper bound K. This also limits the number of states that can be reached for any state x, which is helpful in case $|\mathcal{X}| = \infty$.

The process that we just described is called *uniformization*.

Example 4.6.1 Consider again the single-server queue of Example 4.3.1. Suppose that the system is initially empty. As an example, take $\lambda = 1$ for all t and $\mu = 1/2$ for $t \in [0,10]$, and $\mu = 2$ for $t > 10$, and let us see how $\pi_t(0)$ evolves over time. Computations show that $\pi_{10}(0) \approx 0.033$, $\pi_{15}(0) \approx 0.31$ and $\pi_{20}(0) \approx 0.44$. Note that until $t = 10$ the departures do not counterbalance the arrivals, explaining why $\pi_{10}(0)$ is that small. From time 10 on $\pi_t(0)$ increases steadily to the limit 0.5.

4.7 Further reading

Most books on probability theory, Operations Research, or queueing theory contain one or more chapters on Markov chains. Next to that there are many books entirely devoted to the subject. From this enormous literature we note the Chapters 4 and 7 of Ross [130] and Chapters 3 and 4 of Tijms [152].

Information on the inventor of Markov chains, A.A. Markov, can be found on Wikipedia.

4.8 Exercises

Exercise 4.1 Consider a finite-state discrete-time Markov chain.
a. Explain how to compute π_{64} with as few matrix multiplications as possible.
b. Do the same for π_{21}.

Exercise 4.2 Consider a Markov chain with $\mathcal{X} = \{0, 1, \ldots\}$. For $0 < q < 1$ and all $x \geq 0$ we take $p(x, x+1) = q$ and $p(x+1, x) = 1 - q$, $p(0, 0) = 1 - q$.
a. Find all solutions of Equation (4.4).
b. Determine for which q there is a solution that also satisfies Equation (4.3).
c. Give an intuitive interpretation of your findings.

Exercise 4.3 Consider an arbitrary Markov chain with up to say 10 states.
a. Make an Excel sheet in which you can calculate $\pi_1, \pi_2, ..., \pi_{100}$.
b. Try different examples with a slow and fast convergence to stationarity.
c. Define the distance between π_k and π_{100} in some appropriate way and make a plot as a function of k.

Exercise 4.4 Consider a company with a central telephone switch that is connected to the public network by N outgoing lines. This means that there can be no more than N calls in parallel. Calls arrive according to a Poisson process of rate λ, each call has an exponential duration with parameter μ. We model the number of busy lines as a birth-death process.
a. Give the transition rates of this birth-death process.
b. Give a formula for the probability that all lines are occupied.
c. Calculate this number for $N = 3$, $\lambda = 1$ and $\mu = 0.5$.
d. What do you think that is unrealistic about this model?

Exercise 4.5 We model a hospital ward as follows. Patients arrive according to a Poisson process, and each has an exponentially distributed length of stay. We assume that there is enough capacity to handle all patients (thus, theoretically, an infinite number of beds). We model the occupancy as a birth-death process.
a. Give the transition rates of this birth-death process.
b. Give a formula for the long-run probability that x beds are occupied.
c. Give the coefficient of variation of the number of occupied beds.
d. Given the answer to c, what is your conclusion with respect to the size of hospital wards?

Exercise 4.6 Two machines are maintained by a single repairman. The repair time is exponential with rate μ, each machine fails (when functioning) with a rate λ. When both machines are down one is being repaired, the other is waiting for repair. We model the number of functioning machines as a continuous-time Markov chain.
a. Give the transition rates.
b. Give the expected number of functioning machines and the probability that both machines are not functioning.
c. What is the state distribution perceived by a machine going down?
d. What is the expected time between failure of a machine and the moment the repair is finished?

Exercise 4.7 Consider a system with 3 machines and 2 repairmen. Machines fail independently with rate λ. Each repairman repairs machines at rate μ. When one machine is down then only one repairman can work, when 2 or more machines are down both work.
a. Model this system as a birth-death process.
b. Calculate the stationary distribution and use this to derive the long-run expected number of machines that are functioning.
c. Derive the long-run distribution at moments that a machine fails.
d. Use this to derive the distribution of the long-run average time a machine waits before it is taken into service and calculate its expectation.

Exercise 4.8 Consider a Markov chain with $\mathcal{X} = \{0,1\}$, $\lambda(0,1) = \lambda(1,0) = 1$, and $\pi_0(0) = 1$.
a. Show that $\pi_t(1) = (1 - \exp(-2t))/2$.
Now assume that $\lambda(0,1)$ and $\lambda(1,0)$ are different.
b. Derive, using Equation (4.7), an expression for $\frac{d}{dt}\pi_t(1)$.
c. Find an expression for $\pi_t(1)$.

Chapter 5

Queueing Models

In this chapter we study queueing models. Queueing models are characterized by the fact that customers or jobs compete for the same service(s). The focus in this chapter is on analytic solutions for time-stationary models. This is because there are few results for non-stationary models. Usually one has to rely on simulation or computational Markov chain methods in that case.

The first question for any queueing model is whether it is well-dimensioned: does the processing capacity exceed the expected load per unit of time? If this is the case then the model is called *stable*. However, most of the phenomena that we are interested in deal with consequences of randomness in arrival and service processes. E.g., stable call centers without fluctuations have zero waiting times and in hospitals without fluctuations the number of occupied beds is always the same. This is evidently not the case: randomness is predominant in queueing.

5.1 Classification

A rough classification of the most common models is as follows. A queueing model can have one or more nodes, one or more types of customers, and each node can have one or more servers. A model with multiple nodes is called a *queueing network*. In what follows we discuss first single-node single-type models, then single-node multi-type models, and finally queueing networks.

For the single-node single-type models we use the well-known Kendall notation, which is of the form $A|B|c|d$. Here A represents the arrival process, B the service time distribution, c the number of servers, and d the total number of places in the queueing system. A and B are usually either M ("Markovian", i.e., Poisson arrivals

or exponential service times), D ("deterministic", i.e., constant interarrival or service times), or G ("general" interarrival or service times, sometimes denoted as GI to stress that the interarrivals or departures are mutually independent). Of course c and d are integers, with $c \leq d$. If $d = \infty$ it is often skipped.

Another aspect of queueing models is the *queueing discipline*. Usually we assume that customers at each node are served in the order of arrival, i.e., first-come-first-served (FCFS). In a single server system a customer that is served earlier than another customer leaves the system earlier as well; in this case FCFS is equivalent to first-in-first-out, FIFO. Other well-known disciplines are LIFO (last-in-first-out) and PS (processor sharing), which means that every customer gets an equal part of the service capacity.

5.2 Notation and queueing basics

Over the years a form of standard notation for queueing systems has evolved, which has the following ingredients:
- λ denotes the parameter of the Poisson arrival process;
- S is the service time distribution;
- μ is the parameter of the service time distribution in case it is exponential;
- $\beta = \mathbb{E}S$ is the expected service time (thus $\beta = 1/\mu$ in case of exponentiality);
- s is the number of servers.

In what follows next we use the following notation for waiting times and queue lengths:
- W_Q is the time that an arbitrary customer spends waiting before service, in a stationary situation;
- W is the time that an arbitrary customer spends in the system, while waiting and while being served;
- L_Q is the limiting number of customers in the queue;
- L is the limiting number of customers in the system;
- π is the stationary distribution of the number of customers in the system. In contrast to Chapter 4 we write π instead of π_∞.

Above we introduced notation for performance measures in a stationary situation. However, before we can study stationary behavior, we should determine whether or not the system eventually reaches equilibrium. In terms of discrete-time Markov chains (Section 4.1), this is equivalent to saying that $\mathbb{E}T_x < \infty$. In a queueing context, where all customers wait until they get serviced, stationarity is equivalent to requiring that, on average, per unti of time less work arrives than the server(s) can handle.

This is the case if $\lambda < s/\beta$. Thus $\lambda\beta/s < 1$ is the stability condition. We usually write $\rho = \lambda\beta/s$. Note that when delayed customer leave, then there is no stability issue.

Example 5.2.1 The $M|M|1|\infty$ or $M|M|1$ queue is stable if and only if $\lambda < \mu$; see also Example 4.3.1. The $M|M|1|1$ queue (a two-state continuous-time Markov chain) is always stable.

In Section 3.8 a number of useful relations were derived using so-called *cost equations*. The best known cost equation is Little's law, which states that $l = \mathbb{E}L = \lambda\mathbb{E}W = \lambda w$ (Equation (3.3)) and $\mathbb{E}L_Q = \lambda\mathbb{E}W_Q$ for regenerative processes.

Another important property that we will regularly use is *PASTA*, which stands for "Poisson arrivals see time averages". See Section 3.6.

5.3 Single-server single-type queues

In this section we study the $M|M|1$ and the $M|G|1$ queues. The main results for the $M|M|1$ queue are:

Theorem 5.3.1 ($M|M|1$ queue) *The following results hold for the $M|M|1$ queue with $\rho = \frac{\lambda}{\mu} < 1$: The stationary distribution π is geometric and given by*

$$\pi(j) = (1-\rho)\rho^j, \tag{5.1}$$

$$\mathbb{E}W_Q = \frac{\rho}{\mu(1-\rho)}, \quad \mathbb{E}L_Q = \frac{\rho^2}{1-\rho}, \tag{5.2}$$

$$\mathbb{E}W = \frac{1}{\mu(1-\rho)}, \quad \mathbb{E}L = \frac{\rho}{1-\rho}, \tag{5.3}$$

and

$$\mathbb{P}(W_Q > t) = \rho \mathrm{e}^{-(1-\rho)\mu t}.$$

Theorem 5.3.1 has some interesting consequences. We have $\pi(0) = 1-\rho$, thus the server is busy a fraction ρ of the time. Hence we would expect that $\mathbb{E}L - \mathbb{E}L_Q = \rho$, which is indeed the case. When the server is busy then arriving customers are delayed, and thus, using "Poisson arrivals see time averages" (PASTA): $\mathbb{P}(W_Q > 0) = \rho$. Note that

$$\mathbb{P}(W_Q > t | W_Q > 0) = \frac{\rho \mathrm{e}^{-(1-\rho)\mu t}}{\rho} = \mathrm{e}^{-(1-\rho)\mu t},$$

which is the tail of an exponential distribution. Thus given that you have to wait, your remaining waiting time is exponential. This means for example that the remaining expected waiting time never changes! (See Section 1.6.5 for properties of the exponential distribution.)

It is also interesting to note that $\mathbb{E}L$ and $\mathbb{E}L_Q$ are dimensionless, as they only depend on ρ. Thus if time is scaled, i.e., λ and μ are mutiplied by the same number, then the average queue length does not change.

Proof of Theorem 5.3.1 The expression for $\pi(j)$ has been obtained in Example 4.3.1. The next four expression follow from $\mathbb{E}L = \sum_{j=0}^{\infty} j\pi(j)$, $\mathbb{E}L_Q = \sum_{j=1}^{\infty}(j-1)\pi(j)$, and Little's law, which states that $\mathbb{E}L_{(Q)} = \lambda\mathbb{E}W_{(Q)}$. For the expression of the waiting time distribution we refer to the discussion of the $M|M|s$ queue. □

Quite often arrival processes are (approximately) Poisson; rarely however service times are exponentially distributed. In what follows we derive the expected waiting time in the $M|G|1$ queue. This will be used to study the influence of randomness in the service times on the behavior of the queue. The arrival rate is again λ, service times are i.i.d., denoted with the r.v. S. In accordance with the exponential case we define $\rho = \lambda\mathbb{E}S$.

To be able to study the importance of randomness in what follows we will use the squared coefficient of variation of S. Recall from Section 1.2 that the squared coefficient of variation $c^2(X)$ of a distribution X is defined by $c^2(X) = \mathbb{E}(X - \mathbb{E}X)^2/(\mathbb{E}X)^2$.

Theorem 5.3.2 ($M|G|1$ queue) *For the $M|G|1$ queue with $\rho = \lambda\mathbb{E}S < 1$ holds:*

$$\mathbb{E}W_Q = \frac{\lambda\mathbb{E}S^2}{2(1-\lambda\mathbb{E}S)} = \frac{\rho\mathbb{E}S(1+c^2(S))}{2(1-\rho)}, \quad \mathbb{E}L_Q = \frac{\lambda^2\mathbb{E}S^2}{2(1-\lambda\mathbb{E}S)} = \frac{\rho^2(1+c^2(S))}{2(1-\rho)},$$
$$\mathbb{E}W = \mathbb{E}W_Q + \mathbb{E}S, \quad \mathbb{E}L = \mathbb{E}L_Q + \rho. \tag{5.4}$$

Formula (5.4) is the celebrated *Pollaczek-Khintchine formula*. We see that the expected waiting time depends only on the first two moments of S.

Proof of Theorem 5.3.2 First we calculate the total expected amount of work that an arbitrary customer has in queue over time, i.e., the contribution of a customer to the workload process. When looking at Figure 5.1, this corresponds to the surface between the arrival and the departure of the customer, the shaded area. We call this variable U. Then

$$\mathbb{E}U = \mathbb{E}\Big(SW_Q + \int_0^S (S-x)dx\Big) = \mathbb{E}S\mathbb{E}W_Q + \frac{\mathbb{E}S^2}{2}.$$

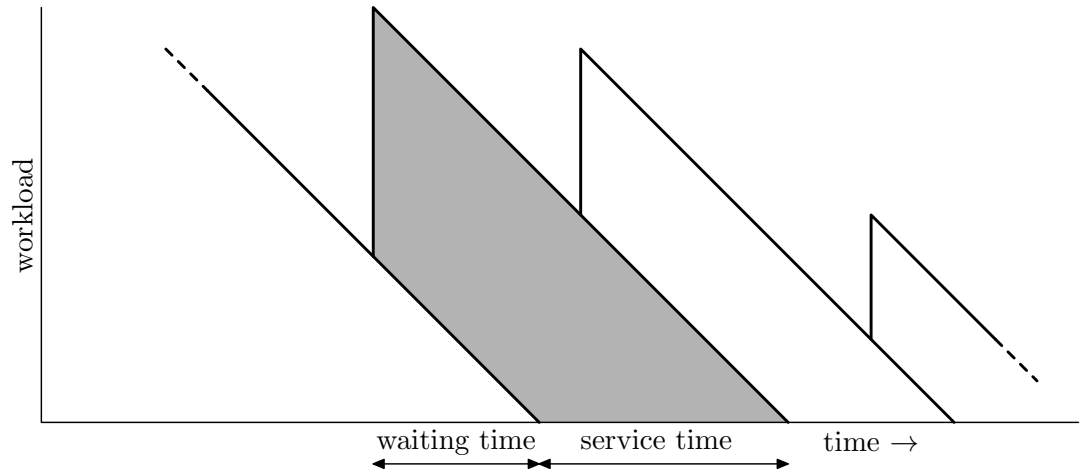


Figure 5.1: The workload process.

Let V be the stationary amount of work in the system. Then there is the "cost equation" $\mathbb{E}V = \lambda \mathbb{E}U$. Thus

$$\mathbb{E}V = \lambda \mathbb{E}S \mathbb{E}W_Q + \frac{\lambda \mathbb{E}S^2}{2}.$$

But PASTA tells us that $\mathbb{E}V = \mathbb{E}W_Q$, and finally we find $\mathbb{E}W_Q = \frac{\lambda \mathbb{E}S^2}{2(1-\lambda \mathbb{E}S)}$. The second expression for $\mathbb{E}W_Q$ is obtained as follows:

$$\frac{\lambda \mathbb{E}S^2}{2(1-\lambda \mathbb{E}S)} = \frac{\lambda (\mathbb{E}S)^2 (1+c^2(S))}{2(1-\lambda \mathbb{E}S)} = \frac{\rho \mathbb{E}S (1+c^2(S))}{2(1-\rho)}.$$

The expression for $\mathbb{E}L_Q$ is derived using Little's law. By an argument using a cost equation (see Section 3.8) it can be shown that ρ is the fraction of the time that the server is busy, which gives $\mathbb{E}L = \rho + \mathbb{E}L_Q$. The expression for $\mathbb{E}W$ follows again from Little's law. □

Remark 5.3.3 Higher moments of W_Q (and therefore the whole distribution) can be derived from the Laplace transform of W_Q, which is the Pollaczek-Khintchine formula in its general form.

Let us interpret Theorem 5.3.2. If S is deterministic, then $c^2(S) = 0$; if S is exponential, then $c^2(S) = 1$. Thus the $M|M|1$ has an expected waiting time which is twice as high as the $M|D|1$ queue (where the D stand for deterministic) with the same ρ. In Figure 5.2 we see $\mathbb{E}W_Q$ as a function of λ for different values of $c^2(S)$.

Note that $\mathbb{E}L_Q$ is again dimensionless, as it depends only on ρ and $c^2(S)$.

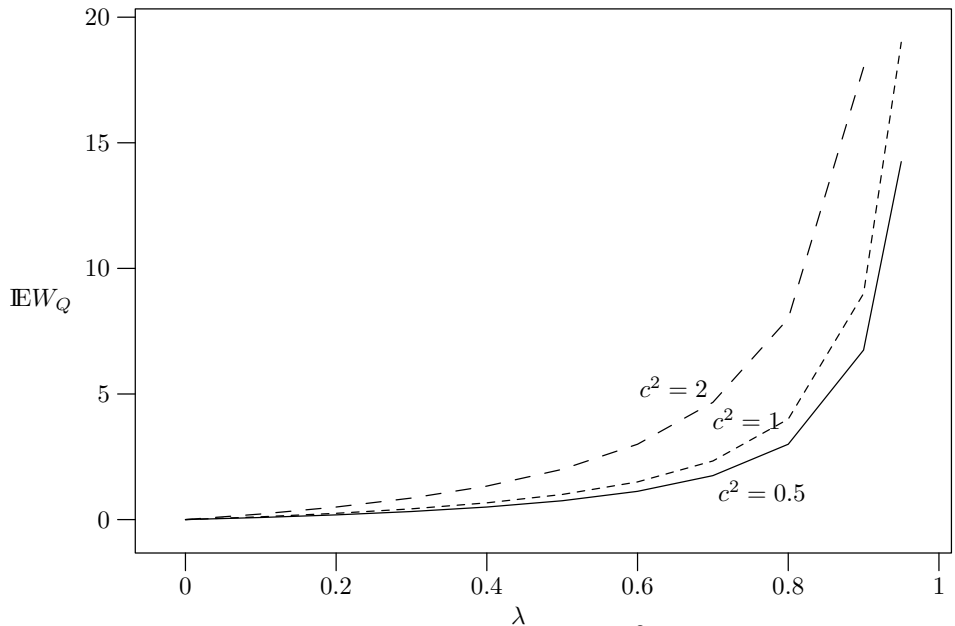


Figure 5.2: $\mathbb{E}W_Q$ as a function of λ and $c^2(S)$, for $\mathbb{E}S = 1$.

Example 5.3.4 Printers are typical examples of $M|G|1$ queues, and we all know that the size of printer jobs has a high variability. This leads to typical behavior of queues with a high $c^2(S)$ and a relatively low ρ: often we find no customers at all, but if there is a queue, then it is often very long.

Remark 5.3.5 There is no known expression for the $G|G|1$ queue. However, equation (5.4) can be used as the basis for an approximation of the expected waiting time of the $GI|G|1$ queue, which is:

$$\mathbb{E}W_Q \approx \frac{\rho \mathbb{E}S(c^2(A) + c^2(S))}{2(1-\rho)},$$

where A is the inter-arrival time and $\rho = \mathbb{E}S/\mathbb{E}A$. Note that the formula is exact for A exponential. This approximation is known as *Kingman's formula*.

Remark 5.3.6 (The $M|G|1$ queue with processor sharing) Above we saw how uncertainty plays a role in the $M|G|1$ queue. An obvious way to reduce waiting times is reducing $c^2(S)$. If we assume the service times as given, and there are no ways to obtain information about the realizations of S, then changing the queueing discipline is an option.

So far we discussed the $M|G|1$ queue with the FIFO discipline. In the case of service times with a high variability this can lead to long delays because of customers demanding long service times blocking the server. A possible solution is changing the queueing discipline, for

example to processor sharing (PS). Because under this discipline every customer is taken in service immediately at its arrival instant, we compare W and not W_Q for FIFO and PS.

First note that for the $M|M|1$ system the queue length is independent of the queueing discipline, which we denote by $\mathbb{E}L(\text{FIFO}) = \mathbb{E}L(\text{PS})$. From this it follows that $\mathbb{E}W(\text{FIFO}) = \mathbb{E}W(\text{PS})$, by Little's law.

It is well known that $\mathbb{E}L(\text{PS})$ is insensitive to higher moments of S (e.g., Section 4.4 of Kleinrock [93]), thus for the $M|G|1$ queue with the PS discipline holds

$$\mathbb{E}L(\text{PS}) = \frac{\lambda \mathbb{E}S}{1 - \lambda \mathbb{E}S} \text{ and } \mathbb{E}W(\text{PS}) = \frac{\mathbb{E}S}{1 - \lambda \mathbb{E}S}.$$

From this it follows directly that

$$\mathbb{E}W(\text{PS}) \leq \mathbb{E}W(\text{FIFO}) \iff c^2(S) \geq 1.$$

Sometimes it is not possible to change the queueing discipline to PS; in some of these cases it is possible to break the service center up in smaller components. Then the $M|G|1$ queue is replaced by an $M|G|s$ queue with the same joint service capacity. Unfortunately there is no expression for the expected waiting time in the $M|G|s$ queue; for individual cases it should be examined whether splitting up the service capacity is an improvement or not. This only works for highly loaded systems and very highly variable service times, as the system works not at its full capacity if not all components are busy.

Example 5.3.7 Coming back to the printer example, it can be argued that it is advantageous to install multiple small printers instead of a single big one; of course all users should have access to all printers.

5.4 Multi-server single-type queues

The $M|M|s$ queue, with arrival rate λ and s servers that each can serve at rate μ, can be modeled as a birth-death process on $\{0, 1, \ldots\}$ (see Section 4.3), where state j represents the number of customers in the system (including the customers in service). This queue is also known as the Erlang C model.

In the sequel we use the following notation: $a = \lambda/\mu$, the offered load (in Erlang), and $\rho = a/s$, the load (in Erlang per server).

Theorem 5.4.1 ($M|M|s$ queue) *The following results hold for the $M|M|s$ queue with $\rho < 1$: The stationary distribution π is given by*

$$\pi(j) = \begin{cases} \dfrac{a^j}{j!}\,\pi(0) & \text{if } j < s, \\ \dfrac{a^j}{s!s^{j-s}}\,\pi(0) & \text{otherwise,} \end{cases}$$

with

$$\pi(0)^{-1} = \sum_{j=0}^{s-1} \frac{a^j}{j!} + \frac{a^s}{(s-1)!\,(s-a)};$$

$$\mathbb{E}W_Q = \frac{C(s,a)}{s\mu - \lambda}, \quad \mathbb{E}L_Q = \frac{\rho C(s,a)}{1-\rho},$$

and

$$\mathbb{P}(W_Q > t) = C(s,a)\mathrm{e}^{-(s\mu-\lambda)t},$$

with

$$C(s,a) = \sum_{j=s}^{\infty} \pi(j) = \frac{a^s}{s!\,(1-a/s)} \left[\sum_{j=0}^{s-1} \frac{a^j}{j!} + \frac{a^s}{s!\,(1-a/s)} \right]^{-1};$$

$$\mathbb{E}W = \mathbb{E}W_Q + 1/\mu, \quad \mathbb{E}L = \mathbb{E}L_Q + a.$$

Note that the constant $C(s,a)$ can be interpreted as the probability of delay: $C(s,a) = \mathbb{P}(W_Q > 0)$. The r.v. $W_Q|W_Q > 0$, the waiting time distribution given that one has to wait, has an exponential distribution with the overcapacity $s\mu - \lambda$ as rate. From this it follows that

$$\sigma^2(W_Q) = \frac{C(s,a)(2 - C(s,a))}{(s\mu - \lambda)^2}. \tag{5.5}$$

Proof of Theorem 5.4.1 Note that the $M|M|s$ queue is a birth-death process. Thus we can use the theory developed in Section 4.3. The transition rates are as follows: $\lambda_j = \lambda$ and $\mu_j = \min\{s,j\}\mu$, $j \geq 0$. The equilibrium equations are given by $\lambda\pi(0) = \mu\pi(1)$ and $(\lambda + \min\{s,i\}\mu)\pi(i) = \lambda\pi(i-1) + \min\{s,i+1\}\mu\pi(i+1)$ for $i > 0$. Summing these equalities for $i = 0$ up to j gives: $\lambda\pi(j) = \min\{s,j+1\}\mu\pi(j+1)$ for all j. From this it follows that:

$$\pi(j) = \begin{cases} \dfrac{a^j}{j!}\,\pi(0) & \text{if } j < s \\ \dfrac{a^j}{s!s^{j-s}}\,\pi(0) & \text{otherwise} \end{cases}$$

The value of $\pi(0)$ can be derived from $\sum_{j=0}^{\infty} \pi(j) = 1$, giving

$$\pi(0)^{-1} = \sum_{j=0}^{s-1} \frac{a^j}{j!} + \frac{a^s}{s!\,(1-a/s)}.$$

Now we calculate the waiting time distribution W_Q. Thanks to PASTA we can interpret W_Q as the time, starting at time 0 with the stationary queue length distribution L, until at least

one of the servers stops working on the jobs that were initially present. This is the moment that the customer arriving at 0 would go into service. The probability $\mathbb{P}(W_Q > t|L = j + s)$ is then equal to the probability that there are less than $j + 1$ departures in t time units. The probability of k departures for $k < j + 1$ is given by the probability that a Poisson distributed random variable with parameter $s\mu t$ has outcome k. Thus

$$\mathbb{P}(W_Q > t|L = j + s) = \mathrm{e}^{-s\mu t} \sum_{k=0}^{j} \frac{(s\mu t)^k}{k!}.$$

We are only interested in $W_Q|W_Q > 0$. Note that W_Q has an atom at 0 of size $1 - \mathbb{P}(W_Q > 0) = 1 - C(s, a)$. For this reason we are interested in the distribution of $(L|W_Q > 0) = (L|L \geq s)$. This distribution is geometric with parameter ρ: $\mathbb{P}(L = s + j|L \geq s) = (1 - \rho)\rho^j$, $j \geq 0$.

Now, using Equation (1.8),

$$\mathbb{P}(W_Q > t|W_Q > 0) = \sum_{j=0}^{\infty} \mathbb{P}(W_Q > t|L = j + s, W_Q > 0)\mathbb{P}(L = j + s|W_Q > 0) =$$

$$\sum_{j=0}^{\infty} \mathbb{P}(W_Q > t|L = j + s)\mathbb{P}(L = j + s|W_Q > 0) = (1 - \rho)\mathrm{e}^{-s\mu t} \sum_{j=0}^{\infty} \sum_{k=0}^{j} \frac{(s\mu t)^k}{k!}\rho^j =$$

$$(1 - \rho)\mathrm{e}^{-s\mu t} \sum_{k=0}^{\infty} \frac{(s\mu t)^k}{k!} \sum_{j=k}^{\infty} \rho^j = \mathrm{e}^{-s\mu t} \sum_{k=0}^{\infty} \frac{(\rho s\mu t)^k}{k!} = \mathrm{e}^{-(1-\rho)s\mu t} = \mathrm{e}^{-(s\mu-\lambda)t}.$$

Putting all together we find that

$$\mathbb{P}(W_Q > t) = C(s, a)\mathrm{e}^{-(s\mu-\lambda)t}.$$

From the interpretation it is immediately clear that

$$\mathbb{E}W_Q = \frac{C(s, a)}{s\mu - \lambda}.$$

By Little's law, $\mathbb{E}L_Q = \lambda\mathbb{E}W_Q$, we also find

$$\mathbb{E}L_Q = \frac{\lambda C(s, a)}{s\mu - \lambda} = \frac{\rho C(s, a)}{1 - \rho}.$$

Note that $\mathbb{E}L_Q$ depends only on λ and μ through the quotient $a = s\rho$.

From a cost equation (see Equation (3.4)) it follows that $\mathbb{E}W = \mathbb{E}W_Q + 1/\mu$. □

Many implementations of the Erlang C formula can be found on the web. See, e.g., www.math.vu.nl/∼koole/obp/ErlangC.

Remark 5.4.2 (The $M|G|s$ queue) For the $M|G|s$ queue no closed-form expressions exist for the average waiting time or other performance measures. Many approximations and numerical methods exist in the literature.

Up to now we studied the $M|M|s$ queue, which is a delay model: customers that find all servers occupied are delayed until service capacity is available. We continue with the $M|G|s|s$ queue, which is a blocking model. We derive its stationary distribution, leading to the surprising fact that this is only a function of the mean of the service time.

Theorem 5.4.3 ($M|G|s|s$ queue) *The following results hold for the $M|G|s|s$ queue with s possibly ∞: The stationary distribution π is given by*

$$\pi(i) = \frac{(\lambda \mathbb{E}S)^i / i!}{\sum_{j=0}^{s} (\lambda \mathbb{E}S)^j / j!}; \tag{5.6}$$

$$\mathbb{E}L = (1 - \pi(s))\lambda \mathbb{E}S. \tag{5.7}$$

Note that for $s = \infty$ the expression simplifies to $\pi(i) = (\lambda \mathbb{E}S)^i / i!\, \mathrm{e}^{-\lambda \mathbb{E}S}$, a Poisson distribution. Note also that $\pi(s)$ for $s < \infty$ represents the blocking probability, thanks to PASTA. The model is also known as the Erlang B model, and the blocking probability $\pi(s)$ is sometimes written as $B(s, a)$, with $a = \lambda \mathbb{E}S$ the offered load. The number $aB(s, a)$ represents the load that is rejected, and $a(1 - B(s, a))$ is the load that enters the system, which is equal to the expected number of occupied servers.

Proof of Theorem 5.4.3 The proof of this result is not straightforward, except for $s = 1$ and ∞, or if the service time is exponentially distributed. For the general case we refer to the literature. For $s = 1$ the result follows readily from renewal theory (see Section 3.3). We continue with the $M|G|\infty(|\infty)$ system. Consider a Poisson process with rate λ on $(-\infty, \infty)$ and consider the $M|G|\infty$ queue at time 0. An arrival that occured at $-t$ for some $t > 0$ is still present at 0 with probability $\mathbb{P}(S > t)$. Thus the total number of customers still present at 0 has a Poisson distribution with parameter $\int_0^\infty \lambda \mathbb{P}(S > t)dt = \lambda \mathbb{E}S$. This corresponds with Equation (5.6).

Equation (5.7) follows from the fact that $(1 - \pi(s))\lambda$ is the average number of admitted customers per unit of time. It also follows directly from $\mathbb{E}L = \sum_{j=0}^{s-1} \lambda \pi(j)$. □

Example 5.4.4 The possible connections between two telephone exchanges can well be modeled by a loss system. Although it is generally thought that the length of telephone calls can be well approximated by an exponential distribution, it is of no influence to the availability of free connections, due to the insensitivity of the loss model.

An important property of the Erlang B model is that it shows economies of scale: when the load and number of servers are increased by the same percentage, then the blocking probability decreases. Similarly, when the servers of two parallel Erlang B models start pooling and effectively become a single Erlang B system, then that system performs better in the sense that the total occupancy is higher. These properties of the Erlang B models are formalized in the following theorem.

Theorem 5.4.5 *For the Erlang B model holds for every $a > 0$ that $B(s, sa)$ is strictly decreasing in s, $s \in \mathbb{N}$;*
for every $\lambda_1, \lambda_2, \beta_1, \beta_2 > 0$ and $s_1, s_2 \in \mathbb{N}$

$$(\lambda_1\beta_1 + \lambda_2\beta_2)B(s_1 + s_2, \lambda_1\beta_1 + \lambda_2\beta_2) \leq \lambda_1\beta_1 B(s_1, \lambda_1\beta_1) + \lambda_2\beta_2 B(s_2, \lambda_2\beta_2). \quad (5.8)$$

Proof The proof of the first expression involves analytical properties of the blocking probability; see the appendix of Smith & Whitt [144]. Equation (5.8) is equivalent to

$$(\lambda_1\beta_1 + \lambda_2\beta_2)(1 - B(s_1 + s_2, \lambda_1\beta_1 + \lambda_2\beta_2)) \geq \lambda_1\beta_1(1 - B(s_1, \lambda_1\beta_1)) + \lambda_2\beta_2(1 - B(s_2, \lambda_2\beta_2)).$$

Using the fact that $aB(s, a)$ is the expected number of occupied servers in the Erlang B model, we have to show that pooling increases the overall server occupation. This can be done using a *coupling argument*, which consists of comparing realizations of the processes in such a way that the pooled system always has a higher occupation. For details, see [144]. □

A further property is that $B(s, sa)$ is not only decreasing, but also *convex* in s. This means that increasing in size pays off less as the size increases: there are *diminishing returns*. There is ample numerical evidence for the correctness of this statement, but a formal proof is lacking.

Remark 5.4.6 (The $M|G|s|N$ system) Some models combine blocking and delay, such as the $M|G|s|N$ queueing systems with $s < N < \infty$. As for the $M|G|s$ queue there is no closed-form solution for the standard performance measures. The stationary distribution of the $M|M|s|N$ can be obtained by analyzing the corresponding birth-death process:

$$\pi(j) = \begin{cases} \dfrac{a^j}{j!}\,\pi(0) & \text{if } 0 \leq j < s \\ \dfrac{a^j}{s!s^{j-s}}\,\pi(0) & \text{if } s \leq j \leq N \end{cases}$$

with

$$\pi(0)^{-1} = \sum_{j=0}^{s-1} \frac{a^j}{j!} + \sum_{j=s}^{N} \frac{a^j}{s! s^{j-s}}.$$

For the $M|M|1|N$ this simplifies to

$$\pi(j) = \frac{\rho^j}{1 + \cdots + \rho^N}.$$

The waiting time distribution is a mixture of gamma distributions, tail probabilities of the waiting time can therefore be calculated easily.

Up to now we discussed models with Poisson arrivals. The motivation for Poisson arrivals is an almost infinite pool of possible customers, who all have a very small arrival rate. Thus the number of customers in service has no influence on the arrival rate.

If the number of potential customers is small, then the number of customers in service or at the queue influences the arrival rate. To be precise, if there are in total n customers in the model, and there are in total j customers at the queue or in service, then the arrival rate is $\lambda(n-j)$. Thus each customer joins the service facility after an exponentially distributed time with parameter λ.

If we assume exponential service times, then the state of the system is completely described by the number of customers in queue. Therefore the model is a birth-death process. On the other hand, the model can also be seen as a two-station queueing network. Thus the results of the next section on networks of queues can also be utilized.

We give the formulas for two situations: where there are enough waiting places next to the $s < n$ servers, and the situations where there are not. We indicate these systems by adding a fifth entry to the Kendall notation, indicating the size of the population.

In the literature finite source models are also called Engset models, in contrast with the Erlang models that have Poisson arrivals. (See also Remark 5.6.6.)

Theorem 5.4.7 (Engset models) *The $M|M|s|\infty|n$ or, equivalently, $M|M|s|n|n$ model (the* Engset delay model*) has as steady state probabilities*

$$\pi(j) = \binom{n}{j} \left(\frac{\lambda}{\mu}\right)^j \pi(0) \tag{5.9}$$

if $0 \leq j \leq s$ *and*

$$\pi(j) = \frac{n!}{(n-j)!s!s^{j-s}} \left(\frac{\lambda}{\mu}\right)^j \pi(0) \tag{5.10}$$

if $j > s$, *with*

$$\pi(0)^{-1} = \sum_{j=0}^{s} \binom{n}{j} \left(\frac{\lambda}{\mu}\right)^j + \sum_{j=s+1}^{n} \frac{n!}{(n-j)!s!s^{j-s}} \left(\frac{\lambda}{\mu}\right)^j; \tag{5.11}$$

The $M|G|s|s|n$ *model (the* Engset blocking model*) has as steady state probabilities*

$$\pi(j) = \binom{n}{j} \left(\lambda \mathbb{E}S\right)^j \pi(0) \tag{5.12}$$

for $0 \leq j \leq s$ *with*

$$\pi(0)^{-1} = \sum_{j=0}^{s} \binom{n}{j} \left(\lambda \mathbb{E}S\right)^j. \tag{5.13}$$

The states with $j > s$ *have* $\pi(j) = 0$.

Proof The results for exponential service times follow using the theory on birth-death processes. For the insensitivity of the $M|G|s|s|n$ blocking model we refer to the literature. □

If $s = n$ then all customers have their "private" server and behave independently, each with stationary probability of $\lambda/(\lambda+\mu)$ of being in service. Therefore the stationary distribution in this case is given by

$$\pi(j) = \binom{n}{j} \left(\frac{\lambda}{\lambda+\mu}\right)^j \left(\frac{\mu}{\lambda+\mu}\right)^{n-j},$$

the binomial distribution. This can also be derived from (5.12) and (5.13), using the fact that $\sum_{j=0}^{n} \binom{n}{j} (\frac{\lambda}{\mu})^j = (1+\frac{\lambda}{\mu})^n$. In the same spirit as for $s = n$, the stationary distribution for $s < n$ can be rewritten as

$$\pi(j) = \frac{\binom{n}{j} (\frac{\lambda}{\lambda+\mu})^j (\frac{\mu}{\lambda+\mu})^{n-j}}{\sum_{i=0}^{s} \binom{n}{i} (\frac{\lambda}{\lambda+\mu})^i (\frac{\mu}{\lambda+\mu})^{n-i}}.$$

This is a binomial distribution cut off at level s.

Note that like the Erlang loss model the finite source loss model is insensitive for the service time distribution.

5.5 Single-server multi-type queues

Now suppose that some knowledge on S is available, i.e., on arrival we know to which class the customer belongs. Customers in each class have their own service time distribution and arrive according to a Poisson process. In this subsection we study the influence of priorities between the classes on the waiting times.

Suppose we have P classes of customers, with service times S_p and arrival rate λ_p, $p = 1, \ldots, P$. Then the service time S of an arbitrary customer is equal to S_p with probability λ_p/λ, with $\lambda = \sum_p \lambda_p$. If customers are served using a FIFO discipline, then the Pollaczek-Khinthine formula (5.4) still holds with $\mathbb{E}S = \sum_p (\lambda_p/\lambda)\mathbb{E}S_p$ and $\mathbb{E}S^2 = \sum_p (\lambda_p/\lambda)\mathbb{E}S_p^2$.

Instead of FIFO we study the head-of-the-line (HOL) discipline, which is defined as follows: customers within a class are served in a FIFO manner, and when the server has finished serving a customer then a waiting customer from the class with the lowest class number is selected.

Define $W_Q(p)$ and $L_Q(p)$ as the waiting times and queue lengths of class p. Define also $\rho_i = \lambda_i \mathbb{E}S_i$ and $\sigma_p = \sum_{i=1}^{p} \rho_i$.

Theorem 5.5.1 (priority queue)

$$\mathbb{E}W_Q(p) = \frac{\mathbb{E}R}{(1-\sigma_p)(1-\sigma_{p-1})}. \tag{5.14}$$

$$\mathbb{E}W_Q(HOL) = \sum_{p=1}^{P} \frac{\lambda_p \mathbb{E}R}{\lambda(1-\sigma_p)(1-\sigma_{p-1})}. \tag{5.15}$$

Proof Above we found the following formula for the $M|G|1$ queue:

$$\mathbb{E}W_Q = \lambda \mathbb{E}S \mathbb{E}W_Q + \frac{\lambda \mathbb{E}S^2}{2}.$$

Using Little's law gives $\mathbb{E}W_Q = \mathbb{E}S\mathbb{E}L_Q + \lambda \mathbb{E}S^2/2$. This term can be explained as follows: $\mathbb{E}S\mathbb{E}L_Q$ takes into account the customers ahead in the queue, $\lambda \mathbb{E}S^2/2$ is the remaining expected service time of the customer currently in service. This can also be shown as follows. Let R be the remaining service time of the customer currently in service. In Section 3.7 an expression was derived for the remaining time until a renewal in a renewal process. This corresponds to R given that the server is busy:

$$\mathbb{E}(R|R>0) = \frac{\mathbb{E}S^2}{2\mathbb{E}S}.$$

Note that $\mathbb{P}(R > 0) = \rho$, following from a cost equation, see Equation (3.4). Using Equation (1.9) we find indeed

$$\mathbb{E}R = \mathbb{E}(R|R > 0)\mathbb{P}(R > 0) + \mathbb{E}(R|R = 0)\mathbb{P}(R = 0) = \frac{\rho \mathbb{E}S^2}{2\mathbb{E}S} = \frac{\lambda \mathbb{E}S^2}{2}.$$

In what follows we introduce the different classes of customers, and we rewrite the above expression for each class of customers.

Define $N_i(p)$ as the number of customers of class i that arrive during $W_Q(p)$. Note that every customer is served eventually (we assume a stable system), and thus R does not depend on the service discipline. Therefore $\mathbb{E}R = \lambda \mathbb{E}S^2/2$. We have:

$$\mathbb{E}W_Q(p) = \mathbb{E}R + \sum_{i=1}^{p} \mathbb{E}L_Q(i)\mathbb{E}S_i + \sum_{i=1}^{p-1} \mathbb{E}N_i(p)\mathbb{E}S_i.$$

By Little's law $\mathbb{E}L_Q(i) = \lambda_i \mathbb{E}W_Q(i)$, by the Poisson arrivals $\mathbb{E}N_i(p) = \lambda_i \mathbb{E}W_Q(p)$. Thus

$$\mathbb{E}W_Q(p) = \mathbb{E}R + \sum_{i=1}^{p} \lambda_i \mathbb{E}S_i \mathbb{E}W_Q(i) + \sum_{i=1}^{p-1} \lambda_i \mathbb{E}S_i \mathbb{E}W_Q(p).$$

Then the last formula is equivalent to

$$(1 - \sigma_p)\mathbb{E}W_Q(p) = \mathbb{E}R + \sum_{i=1}^{p-1} \lambda_i \mathbb{E}S_i \mathbb{E}W_Q(i). \tag{5.16}$$

By induction to p we can now show Equation (5.14). For $p = 1$ it follows right away from Equation (5.16). Let us now assume that Equation (5.14) holds up to p. Now substract (5.14) for p from the same equation for $p + 1$. This leads to

$$(1 - \sigma_{p+1})\mathbb{E}W_Q(p + 1) = (1 - \sigma_p)\mathbb{E}W_Q(p) + \lambda_p \mathbb{E}S_p \mathbb{E}W_Q(p).$$

Filling in the formula for $\mathbb{E}W_Q(p)$ quickly leads to the equation for $\mathbb{E}W_Q(p + 1)$. This proves the waiting time expression for each class separately. An arbitrary customer belongs to class p with probability λ_p/λ. From this the expected waiting under HOL, Equation (5.15), follows.

□

Example 5.5.2 As an example, we study the case $P = 2$, and we will analyze in which cases FIFO or HOL is better. From the above it follows that:

$$\mathbb{E}W_Q(1) = \frac{\mathbb{E}R}{(1 - \rho_1)}, \ \mathbb{E}W_Q(2) = \frac{\mathbb{E}R}{(1 - \rho_1 - \rho_2)(1 - \rho_1)}.$$

Then:

$$\mathbb{E}W_Q(HOL) = \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbb{E}W_Q(1) + \frac{\lambda_2}{\lambda_1+\lambda_2}\mathbb{E}W_Q(2) = \frac{\mathbb{E}R(\lambda_1(1-\rho_1-\rho_2)+\lambda_2)}{(\lambda_1+\lambda_2)(1-\rho_1-\rho_2)(1-\rho_1)}.$$

We already saw that

$$\mathbb{E}W_Q(FIFO) = \frac{\mathbb{E}R}{1-\rho_1-\rho_2}.$$

A simple computation shows that $\mathbb{E}W_Q(HOL) \leq \mathbb{E}W_Q(FIFO)$ if and only if $\mathbb{E}S_1 \leq \mathbb{E}S_2$. Thus if shorter jobs get priority the average waiting time decreases. Of course, the customers with long waiting times have to pay for this.

In the example we saw that scheduling customers with short service times first decreases the average waiting time. Thus the queueing discipline that schedules the waiting job with the shortest processing time first, the *shortest-job-first* (SJF) discipline, minimizes the average waiting time. The waiting time can be computed from the above formula, by making a class of each possible service time. This gives:

$$\mathbb{E}[W_Q(SJF)|S=x] = \frac{\mathbb{E}R}{(1-\lambda\int_0^x t f_S(t)dt)^2},$$

with f_S the density of S. For $x=0$ we find $\mathbb{E}[W_Q(SJF)|S=0] = \mathbb{E}R$, which makes sense: a customer requiring 0 service gets priority over all waiting customers. On the other hand we find $\mathbb{E}[W_Q(SJF)|S=\infty] = \mathbb{E}R/(1-\rho)^2$, the time until the first moment the system gets empty. The expected waiting time is given by

$$\mathbb{E}W_Q(SJF) = \int_{x=0}^{\infty}\mathbb{E}[W_Q(SJF)|S=x]f_S(x)dx = \int_{x=0}^{\infty}\frac{\mathbb{E}Rf_S(x)}{(1-\lambda\int_0^x t f_S(t)dt)^2}dx. \quad (5.17)$$

This expression is not easy to solve for specific examples; using Maple we found for exponential service times with $\mu=1$ and various values of λ the results of Table 5.1.

λ	0.1	0.5	0.75	0.9
$\mathbb{E}W_Q(SJF)$	0.10	0.71	1.55	3.20
$\mathbb{E}W_Q(FIFO)$	0.11	1.00	3.00	9.00

Table 5.1: Waiting times for exponential service times with $\mu=1$.

SJF is optimal even if the customers do not arrive according to a Poisson process. Only in special cases we can compute the waiting times.

Remark 5.5.3 There can be many reasons to use HOL. If one is interested in (weighted) waiting times, then the problem is to order the classes such that the expected costs are minimal. Thus: reorder $1, \ldots, P$ such that

$$\mathbb{E}C_Q(HOL) = \sum_{i=1}^{P} \frac{c_p \lambda_p \mathbb{E}R}{\lambda(1-\sigma_p)(1-\sigma_{p-1})}$$

is minimal. By exchanging the order 1 by 1 (as in the example) it can be shown that the classes should be ordered such that $c_1/\mathbb{E}S_1 \geq \cdots \geq c_P/\mathbb{E}S_P$. For exponential service times this translates to $c_1\mu_1 \geq \cdots \geq c_P\mu_P$, which is the reason why this policy is commonly known as the μc rule.

Name	arrival process	service time distribution	# of servers	queue	page
Pollaczek-Khinthine $(M\|G\|1)$	Poisson	general	1	yes	66
Kingman's formula $(G\|G\|1)$	general	general	1	yes	68
Erlang C $(M\|M\|s)$	Poisson	exponential	s	yes	69
Erlang B $(M\|G\|s\|s)$	Poisson	general	s	no	72
Engset delay $(M\|M\|s\|n\|n)$	finite source	exponential	s	yes	74
Engset blocking $(M\|G\|s\|s\|n)$	finite source	general	s	no	74
multi-type priority queue	Poisson	general	1	yes	76

Table 5.2: Overview of queueing models

5.6 Queueing networks

So far we studied queueing systems with a single service station. In this section we will make a start with the study of networks of queues. First we consider a tandem system consisting of a number of ∞-capacity single-server queues with exponential service times and Poisson input at the first queue. Let there be V queues, server j with rate μ_j, and arrival rate λ at the first queue. See Figure 5.3 for a 2-queue example.

Theorem 5.6.1 (tandem queueing system) *For a tandem queueing system with V queues and $\lambda < \mu_j$ for all j, the input to each queue is Poisson and the marginal queue lengths are each independent, with joint steady state distribution*

$$\mathbb{P}(N_i = n_i, i = 1, \ldots, V) = \prod_{i=1}^{V} \Big(1 - \frac{\lambda}{\mu_i}\Big)\Big(\frac{\lambda}{\mu_i}\Big)^{n_i}. \tag{5.18}$$

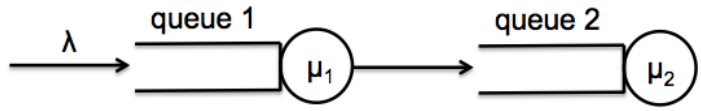


Figure 5.3: A tandem system with 2 queues.

Proof We give the proof for $V = 2$. Let us first consider *time-reversibility*. Consider a Markov process N with rates $\lambda(i,j)$ and stationary distribution π. Now we define the reversed process $\tilde{N}$ by taking its transition rates $\tilde{\lambda}$ such that

$$\pi(i)\lambda(i,j) = \pi(j)\tilde{\lambda}(j,i). \tag{5.19}$$

It can be seen that $\tilde{\pi}$, the stationary distribution of $\tilde{N}$, is equal to π. From (5.19) it follows that N moves as often from i to j as $\tilde{N}$ moves from j to i. This means that under stationarity $\{N(t), t \in \mathbb{R}\}$ and $\{\tilde{N}(s-t), t \in \mathbb{R}\}$ are stochastically indistinguishable. If $\lambda(i,j) = \tilde{\lambda}(i,j)$ for all i and j, then we say that N is time-reversible. This we use for the study of two $M|M|1$ queues in series.

For the $M|M|1$ queue it follows directly from the stationary distribution given in (5.1) that $\pi(i)\lambda(i,j) = \pi(j)\lambda(j,i)$. Thus the $M|M|1$ queue is time-reversible. (In fact, any birth-death process is.) Therefore, the departures before t in N are the arrivals after t in $\tilde{N}$. The arrivals in $\tilde{N}$ are Poisson with rate λ, and thus so are the departures in N. Furthermore, these arrivals in $\tilde{N}$ after t are independent of $\tilde{N}(t)$, and thus the departures in N before t are independent of $N(t)$. In conclusion: the departure process of an $M|M|1$ queue forms a Poisson process and past departures are independent of the current state. Now going to a system of two $M|M|1$ queues in tandem, we observe that the state of the second queue N_2 at t depends on the departures at queue 1 before t. Thus $N_1(t)$ and $N_2(t)$ are independent, and thus

$$\mathbb{P}(N_1(t) = n_1, N_2(t) = n_2) = \mathbb{P}(N_1(t) = n_1)\mathbb{P}(N_2(t) = n_2) =$$

$$\Big(1 - \frac{\lambda}{\mu_1}\Big)\Big(\frac{\lambda}{\mu_1}\Big)^{n_1}\Big(1 - \frac{\lambda}{\mu_2}\Big)\Big(\frac{\lambda}{\mu_2}\Big)^{n_2}.$$

□

Theorem 5.6.1 is a very useful result: expected queue lengths and thus also expected waiting times can be calculated from it. However, we cannot derive results for waiting time distributions from it, because for example the length of the second queue at $t + s$ depends on the length of the first queue at t. The independence holds only if the queues are considered at the same moment.

When there is feedback in the system the traffic between stations is no longer according to a Poisson process. Surprisingly enough the generalization of equation (5.18) holds as well when there is feedback in the system. We will show this for

networks of general birth-death processes with Jackson routing. This is defined as follows. Assume we have V stations, and the service rate at station p is equal to $\mu_p(k)$ if there are k customers present. Of course $\mu_p(0) = 0$. There are outside arrivals at station p according to a Poisson process with rate λ_p. If a customer leaves station i then it joins station j with probability r_{ij}. We assume that $\sum_{j=1}^{V} r_{ij} = 1 - r_{i0} \leq 1$, where we see station 0 as the outside of the network.

Evidently on average more customers arrive at station p during a unit of time than λ_p, due to the feedback. We determine this arrival rate, notated with γ_i. It is given by:

$$\gamma_i = \lambda_i + \sum_{j=1}^{V} \gamma_j r_{ji}. \tag{5.20}$$

We call these the *routing equations*. If this system has as unique solution $\gamma_i = 0$ if $\lambda_i = 0$ for all i then we call the network open: all customers eventually leave the system. If $r_{i0} = 0$ for all i then the network is closed, in this case λ_i needs to be 0 for all i. otherwise the system is unstable.

Theorem 5.6.2 (open queueing network) *The joint steady state distribution of an open queueing network is given by*

$$\mathbb{P}(N_1 = n_1, \ldots, N_V = n_V) = \prod_{i=1}^{V} \pi_i(n_i). \tag{5.21}$$

with

$$\pi_i(n_i) = \mathbb{P}(N_i = n_i) = \Big(\sum_{n=0}^{\infty} \prod_{j=1}^{n} \frac{\gamma_i}{\mu_i(j)} \Big)^{-1} \prod_{j=1}^{n_i} \frac{\gamma_i}{\mu_i(j)}, \tag{5.22}$$

if the denominator of the last expression exists for all i.

Proof It suffices to show that $\pi(n) = \mathbb{P}(N_1 = n_1, \ldots, N_V = n_V)$ satisfies the balance equations, given by

$$\pi(n)\Big(\sum_{i=1}^{V} \lambda_i + \sum_{i=1}^{V} \mu_i(n_i) \Big) = \sum_{i=1}^{V} \lambda_i \mathbb{I}\{n_i > 0\} \pi(n - e_i) +$$

$$\sum_{i=1}^{V} \mu_i(n_i + 1) r_{i0} \pi(n + e_i) + \sum_{j=1}^{V} \sum_{i=1}^{V} \mu_j(n_j + 1) r_{ji} \mathbb{I}\{n_i > 0\} \pi(n + e_j - e_i).$$

However, this is a tedious task. When doing it, one realizes that in fact the balance equations can be decomposed in $V+1$ equations, which all hold separately. There are:

$$\pi(n)\mu_i(n_i) = \lambda_i \pi(n-e_i) + \sum_{j=1}^{V} \mu_j(n_j+1) r_{ji} \pi(n+e_j-e_i),\ i=1,\ldots,V,$$

and

$$\pi(n) \sum_{i=1}^{V} \lambda_i = \sum_{i=1}^{V} \mu_i(n_i+1) r_{i0} \pi(n+e_i).$$

These equalities are called the *station balance* equations. Note that we left out the indicator, as the corresponding stationary probabilities are 0. Using the routing equations and filling in $\pi(n)$ the proof that these equations hold is relatively simple:

$$\pi(n)\mu_i(n_i) = \pi(n-e_i)\gamma_i = \pi(n-e_i)\Big(\lambda_i + \sum_{j=1}^{V} r_{ji}\gamma_j\Big) =$$

$$\pi(n-e_i)\lambda_i + \pi(n-e_i+e_j) \sum_{j=1}^{V} r_{ji}\mu_j(n_j+1)$$

for station $1,\ldots,V$ and

$$\pi(n) \sum_{i=1}^{V} \lambda_i = \pi(n) \sum_{i=1}^{V} \Big(\gamma_i - \sum_{j=1}^{V} \gamma_j r_{ji}\Big) = \pi(n) \sum_{i=1}^{V} \gamma_i r_{i0} = \sum_{i=1}^{V} \pi(n+e_i)\mu_i(n_i+1) r_{i0}$$

for station 0. □

For the special case of single-server queues the marginal distributions simplify to the expression for the $M|M|1$ queue (see Theorem 5.3.1) and expressions for expected waiting times and queue lengths can be given.

Theorem 5.6.3 (open network of single-server queues) *If $\gamma_i < \mu_i$ for all i, then we have for the open queueing network consisting of single-server queues:*

$$\mathbb{P}(N_i = n_i, i=1,\ldots,V) = \prod_{i=1}^{V} (1-\frac{\gamma_i}{\mu_i}) \Big(\frac{\gamma_i}{\mu_i}\Big)^{n_i},$$

$$\mathbb{E}W_Q = \frac{\sum_{j=1}^{V} \frac{\gamma_j^2}{\mu_j(\mu_j-\gamma_j)}}{\sum_{j=1}^{V} \lambda_j}, \quad \mathbb{E}L_Q = \sum_{j=1}^{V} \frac{\gamma_j^2}{\mu_j(\mu_j-\gamma_j)},$$

$$\mathbb{E}W = \frac{\sum_{j=1}^{V} \frac{\gamma_j}{\mu_j-\gamma_j}}{\sum_{j=1}^{V} \lambda_j}, \text{ and } \mathbb{E}L = \sum_{j=1}^{V} \frac{\gamma_j}{\mu_j-\gamma_j}.$$

Remark 5.6.4 As we did for single queue models, we can approximate network performance under general service time assumptions, see again Chapter 5 of [67] for an overview. The crucial point is that the output of an $M|G|1$ is not Poisson; therefore the network becomes a network of $G|G|1$ queues. The output of one station is the input to the next, therefore we approximate as well the squared coefficient of variation c_d^2 of the output process:

$$c_d^2 = (1-\rho^2)c^2(A) + \rho^2 c^2(S).$$

We will not go into detail about the approximation; just note that the value for $\rho \approx 0$ and $\rho \approx 1$ is correct.

Now consider a tandem system, in which customers visit all queues one by one. Assume that queue i has service time S_i, and queue 1 has interarrival times A_1 with $c^2(A_1) = c_{a1}^2$ as squared coefficient of variation. Then we use as estimation for the squared coefficient of variation of the input to queue i the number

$$c_{a,i+1}^2 = (1-\rho_i^2)c_{a,i}^2 + \rho_i^2 c^2(S_i).$$

Now together with the approximation of the waiting time we can find the expected lead time and work-in-process. Experiments on Page 190 however show that the approximation is not that accurate.

Next we consider closed systems: thus $\lambda_i = 0$ for all i, and also $\sum_{j=1}^V r_{ij} = 1$, to avoid that the system empties. We define again γ_i by

$$\gamma_i = \sum_{j=1}^V \gamma_j r_{ji}.$$

Note that if γ_i is a solution to this equation, then so is $c\gamma_i$. To make the solution unique we assume that we take γ such that $\sum_i \gamma_i = 1$. (We will see later on that we only need that $\gamma_i > 0$.) The rest of the analysis seems to go as for the open networks, leading to the same stationary distribution. However, here we forget that this time the system is not irreducible: we have to condition on the number of customers M in the system. This leads to the following theorem.

Theorem 5.6.5 (closed queueing network) *The stationary distribution of a closed queueing network with M customers in the system is:*

$$\mathbb{P}(N_1 = n_1, \ldots, N_V = n_V) = \Big(\sum_{m:\sum_i m_i = M} \prod_{i=1}^V \pi_i(m_i) \Big)^{-1} \prod_{i=1}^V \pi_i(n_i) \tag{5.23}$$

if $n_1 + \cdots + n_V = M$, 0 otherwise, with $\pi_i(n_i)$ as in Theorem 5.6.2.

Due to the form of (5.22) it is easily seen that multiplying γ with a constant gives the same solution.

Example 5.6.6 We consider a two-station model, where one is a single server queue with ∞ capacity (with service rate μ), the other is an ∞-server queue (with service rate α). The ∞-server queue models a finite customer source, as such this model is known under the name *Engset delay model*, and often considered as a single station model. Here we use the theory of closed networks to derive its stationary distribution. As solution to the routing equations we take $\gamma_1 = \gamma_2 = 1$. We already know that $\pi_1(i) = (1-\gamma_1/\mu)(\gamma_1/\mu)^i$; it can be easily verified that $\pi_2(i) = \exp(-\gamma_2/\alpha)(\gamma_2/\alpha)^i/i!$. Thus the stationary distribution is

$$\mathbb{P}(N_1 = n_1, N_2 = n_2 = M - n_1) = C\prod_{i=1}^{V}\pi_i(n_i) =$$

$$C\Big(1-\frac{\gamma_1}{\mu}\Big)\Big(\frac{\gamma_1}{\mu}\Big)^{n_1} e^{\frac{-\gamma_2}{\alpha}} \frac{\frac{\gamma_2}{\alpha}^{M-n_1}}{(M-n_1)!} = \tilde{C}\frac{\Big(\frac{\alpha}{\mu}\Big)^{n_1}}{(M-n_1)!}.$$

The normalizing constant is given by

$$\tilde{C}^{-1} = \sum_{n=0}^{M}\frac{\Big(\frac{\alpha}{\mu}\Big)^n}{(M-n)!} = \Big(\frac{\alpha}{\mu}\Big)^M\sum_{n=0}^{M}\frac{\Big(\frac{\mu}{\alpha}\Big)^n}{n!} \approx \Big(\frac{\alpha}{\mu}\Big)^M\exp\Big(\frac{\mu}{\alpha}\Big).$$

Thus $\tilde{C} \approx (\mu/\alpha)^M\exp(-\mu/\alpha)$.

Equation (5.23) is less useful then it appears to be at first sight, because the normalizing constant is very hard to calculate. Thus computing expected queue lengths and waiting times on the basis of this formula is computationally not feasible for reasonably sized networks.

A solution to this is a recursive method to compute the mean waiting times, called *mean value analysis*. We write it out for a network of single server queues. Denote with $W^M(j)$ the sojourn time of an arbitrary customer at queue j, and with $L^M(j)$ the queue length. The following result, known as the *Arrival Theorem*, is crucial to our analysis: A customer in the network arriving at a queue sees the network as if he is not in the network. For open networks, this does not induce a change; for closed networks with M customers, this means that a customer who changes queue sees the network in equilibrium as if there are $M-1$ customers. This gives

$$\mathbb{E}W^M(j) = \frac{1+\mathbb{E}L^{M-1}(j)}{\mu_j}.$$

A cost equation shows that $\mathbb{E}L^M(j) = \delta_j^M \mathbb{E}W^M(j)$, with δ_j^M the throughput of queue j, i.e., the number of customers that are served on average by queue j. We call $\delta^M = \sum_j \delta_j^M$ the system throughput; from the routing mechanism it is clear that $\delta_j^M = \gamma_j \delta^M$, if we assume that $\sum_j \gamma_j = 1$. Then

$$M = \sum_{i=1}^{V} L^M(i) = \delta^M \sum_{i=1}^{V} \gamma_i \mathbb{E}W^M(i).$$

Now we have all ingredients to write $\mathbb{E}W^M(j)$ in a recursive way:

$$\mathbb{E}W^M(j) = \frac{1 + \mathbb{E}L^{M-1}(j)}{\mu_j} = \frac{1 + \delta^{M-1}\gamma_j \mathbb{E}W^{M-1}(j)}{\mu_j} = \frac{1}{\mu_j} + \frac{(M-1)\gamma_j \mathbb{E}W^{M-1}(j)}{\mu_j \sum_{i=1}^{V} \gamma_i \mathbb{E}W^{M-1}(i)}.$$

This results in a $O(MV)$ algorithm to compute the average waiting times.

5.7 Further reading

Standard books on queueing theory are Cooper [44], Kleinrock [94], and Gross & Harris [70]. See also Chapter 8 of Ross [130] and parts of Tijms [152]. The latter book also discusses the *uniformization method*, a computational method for computing performance measures of transient Markov chains that can also be applied to (small) queueing models. More advanced books are Walrand [159], Cohen [42] and Kleinrock [93]. A practice-oriented introduction is King [89].

Handbook 2 on stochastic models [74] contains chapters on queueing by Cooper and Walrand, and Handbook 3 on computing [41] has a chapter by Mitrani [116] on queueing models of computer systems.

Most of the results of this chapter can be found in any of the above references; the part on priority queueing is based on Chapter 3 of Kleinrock [93]. Mitrani [116] also discusses priority queueing and processor sharing. Kelly [86] is the standard reference to reversibility, but the basic ideas can be found in many books (e.g., [159, 130]). Approximations for the $M|G|s$ queue can be found in Sze [148], see also Tijms et al. [153].

More information about the Scandinavian queueing pioneers Erlang and Engset can be found on Wikipedia. See also Myskja [117].

5.8 Exercises

Exercise 5.1 Calculate the expected waiting time in the $M|G|1$ queue for all 12 parameter combination of $\lambda = 0.5, 0.8, 0.9$ and $\beta = \mathbb{E}S = 1$, with S deterministic, exponential, hyperexponential, and gamma with shape parameter 2. A hyperexponential distribution is a random mixture of exponential distributions; choose the parameters as you like (but do not take the trivial case with both exponentials having the same parameter).

Exercise 5.2 A printer can be modeled as an $M|G|1$ queue. For a specific printer it was determined that the arrival rate was 1 and that the average service time was 0.5, and that the printing time of an arbitrary document has approximately an exponential distribution.
a. Calculate the expected time between the moment a printer job is submitted and the moment the printer finishes printing it (the "system time").
It is found that the time to print a job is much longer in reality. The answer lies in printer failures: about 1% of the jobs cause the printer to get jammed. Repair takes on average 30 time units. By lack of data the repair time is assumed to be exponentially distributed. For the sake of the calculation the repair time is added to the printing time.
b. Calculate the first and second moment of this new printing time.
c. Calculate the expected system time if repairs are included.

Exercise 5.3 Consider an $M|G|1$ queue with arrival rate 0.5 and service time distribution $S = X + Y$, with X and Y independent and both exponentially distributed with rates 1 and 2, respectively.
a. Calculate $\mathbb{E}S$, $\mathbb{E}S^2$, $\sigma^2(S)$ and $c^2(S)$.
b. Calculate the expected waiting time and the expected sojourn time for the $M|G|1$ queue.

Exercise 5.4 Consider a system of two parallel $M|D|1$ queues. Both have load 80%, but one has service times of length 1 and the other of length 10.
a. Calculate the system times in both systems, and the expected overall system time.
The manager of the system considers merging both queues to obtain economies of scale. We approximate the resulting $M|D|2$ queue by a single $M|D|1$ queue with double service speed. Customer are treated in the order of arrival.
b. Characterize the arrival process and the service time distribution of the resulting queueing system.

c. Calculate the system time in this new $M|G|1$ queue.
d. Compare the results found under a and c and give an intuitive explanation. How would you redesign the system as to obtain the lowest possible average system time?

Exercise 5.5 In Remark 5.3.5 an approximation is given for the $G|G|1$ queue. Use this approximation to answer the following questions.
a. Give an approximation for the waiting time in the $D|M|1$ queue.
b. Compute it for $\lambda = 0.5, 0.8, 0.9$ and $\beta = \mathbb{E}S = 1$.
Let A be the interarrival time in a $G|M|1$ queue.
c. Give the distribution of A for the following *compound Poisson process*: batches of orders arrive according to a Poisson process, each batch having a geometrically distributed number of customer orders.
d. Compute $\mathbb{E}A^2$.
e. Give the approximation for this situation and compute it for $\lambda = 0.25, 0.4, 0.45$, $\beta = \mathbb{E}S = 1$, and average batch size 2.

Exercise 5.6 Consider an $M|M|s$ queue with $\mu = 0.2$.
a. Compute, using a tool such as www.math.vu.nl/∼koole/obp/ErlangC, the number of servers needed to assure that $\mathbb{P}(W_Q \leq 0.5) \geq 0.8$, for $\lambda = 1, 10$, and 100.
b. Give the overcapacity in each of the cases.
c. Give a definition of productivity of the servers and compute it for the three cases.

Exercise 5.7 Consider the $M|M|2$ queue, choose some arbitrary λ.
a. Give a formula for $C(s,a)$, and calculate it for $\rho = 0.5, 0.75$, and 0.95.
b. Give a formula for $\mathbb{P}(W_Q > t)$, and calculate it for the same parameter values, for some $t > 0$.

Exercise 5.8 Recall that $B(s,a)$ is the blocking probability in the Erlang B queueing system.
a. Let $N \sim \text{Poisson}(a)$. Show that $B(s,a) = \mathbb{P}(N = s)/\mathbb{P}(N \leq s)$.
b. Use this to write a simple Erlang B calculator in R, python or Excel with the help of an appropriate Poisson function.
c. Show that $C(s,a) = sB(s,a)/(s - a(1 - B(s,a)))$.
d. Use this to write a simple Erlang C calculator.
e. Compare the results with existing calculators.

Exercise 5.9 Prove (5.5).

Exercise 5.10 Use an Erlang B calculator (www.math.vu.nl/∼koole/obp/ErlangB for example) to answer the following questions. A hospital has two wards with both 10 beds and an offered load of 9 Erlang. Patients arrive according to a Poisson process.
a. Give the probability that a patient is rejected because no bed is available.
b. Both wards are merged into a single ward. What is now the rejection probability?
c. Another ward with the same parameters is merged. What is now the rejection probability?
d. Consider hospital wards with the same load (offered load divided by size), but varying size. What do you think about the signs of the first and second derivative of the rejecting probability as a function of the size? How would you call these properties in economic terms?

Exercise 5.11 A small town has a single ambulance for dealing with all emergencies in the area (24 hours a day, 7 days a week). Research shows that emergency calls arrive according to a Poisson process. Emergency handling time consists roughly of driving time to the site of the accident, handling time on the site, and driving time to the hospital.
a. Define the variables involved in this system.
b. Give a formula for the probability that the ambulance is busy at the moment an emergency call arrives. Did you make any assumptions?
c. Derive an approximation for the expected time between an emergency call and the moment the ambulance arrives at the site of the accident. Did you make any (additional) assumptions?
d. Indicate how you could answer questions b and c for the case of 2 ambulances.

Exercise 5.12 Consider the Engset model $M|M|1|3|3$.
a. Calculate the stationary distribution.
b. Calculate the distribution as it is seen by arriving customers.

Exercise 5.13 Consider a tandem of two exponential single-server queues with rates 1 and 2 and Poisson arrivals.
a. What is the maximum arrival rate?
b. Calculate the expected waiting time at queue 1 divided by the total expected waiting time, as a function of the arrival rate.
c. What do you conclude from that?

Exercise 5.14 Consider a tandem of infinite server queues with Poisson arrivals and exponential service times.

a. Calculate the stationary distribution and the distribution of the total number of customers.
b. What do you conclude from that concerning the $M|G|\infty$ queue?

Exercise 5.15 A production system consists of three machines in tandem with infinite buffer space. The second process step fails in 20% of the cases. For this reason there is a quality check (taking no time) after the second step that sends all parts for which the second step failed back to the queue at the second step for processing. Service times are assumed to be exponential, with rates that you can choose arbitrarily.
a. Model this production system as a queueing network.
b. For an arbitrary arrival rate, solve the routing equations.
c. What is the maximum production rate of this system?
d. For a Poisson order arrival process at 80% of this maximum, what are the expected waiting and response times?

Exercise 5.16 a. Calculate the stationary distribution of an open queueing network consisting of two single-server queues in tandem.
b. Calculate the stationary distribution of a closed queueing network consisting of two single-server queues in a cycle.
c. Simplify this expression as much as possible, and determine the most likely state(s) as a function of the parameters.

Exercise 5.17 Consider a cycle of two single-server queues with rates 1 and 2 and 5 customers in the system.
a. Compute the stationary distribution and use that to compute the expected waiting time at queue 1.
b. Compute the same number again using mean value analysis.

Chapter 6

Inventory Models

In this chapter we study mathematical models for determining order sizes and moments. Although this usually results in inventory, a better name than the usual *inventory models* would be *order models*. We stick to the standard terminology. Inventory models are often used in distribution and sales environments, but sometimes they can also be used in other application areas: see for example Chapter 18 on revenue management. In this chapter we discuss the basic inventory models. Randomness plays again an important role. Next to that we pay attention to robustness.

6.1 Objectives and notation

The main characteristic of inventory models is demand that can be met by ordering items and keeping them on stock. We make a difference between single-order models and long-term problems for which orders are placed at regular intervals. Single-order models are direct applications of probability theory; long-term inventory models are examples of stochastic processes. The crucial difference with queueing models (see Chapter 5) is that the lead time, the time between the order moment and the delivery, does not depend on the order size. Thus we can assume that items are treated in batches. Capacity restrictions play a minor role. Instead, in inventory models there is a focus on costs.

Costs involve, in the first place, holding costs and sometimes order costs. Holding costs are usual linear in the number of items in inventory and the time that items stay in inventory; order costs can be linear in the order size, but often there is also a fixed component. We will denote the holding costs per item per unit of time with h, the order price per item with k, and the fixed order costs with K. In the case of long-term

models it can either be the case that there are regular order moments or that orders can be placed at any time. We denote the stochastic demand in an interval with D, and the demand during the lead time L with D_L. Every item has a selling price p. The distribution function of D will be denoted by F_D, and D_L has distribution function F_L. Usually demand is of a discrete nature, and only an integer number of items can be ordered. Sometimes the goods are of a continuous nature, and often this is a good approximation of the discrete case. For goods of a continuous nature we assume that D has a density f_D and D_L density f_L.

It might occur that there is demand that cannot be met immediately because the inventory is 0. At that moment two things can happen: the item is *backordered*, or the sale is lost. The objective can either be cost minimization under constraints on the fraction of back orders or lost sales, or cost minimization where costs for back orders or lost sales are included. In the former we use α to denote the maximal fraction of lost sales or back orders as part of total demand, in the latter case we denote with q the costs per back order or lost sale.

In the next sections we discuss all possible models with the above characteristics. We start in Section 6.2 with single-order models, the so-called *newsvendor* or *newsboy problem*. Then we discuss a continuous-review deterministic demand model with fixed order costs leading to the central *Economic Order Quantity*. In that section we also discuss periodic-review models. The consecutive section discusses the extensions to stochastic demand. We conclude with a section treating possible extensions of the models.

We summarize the notation. The input parameters of our models are:
- h: holding costs per item per unit of time;
- k: order price per item;
- p: selling price per item;
- K: fixed order costs;
- L: lead time;
- D: demand (if relevant in an interval, usually of length 1);
- λ: the average demand per time unit, $\lambda = \mathbb{E}D$;
- D_L: demand during the lead time L;
- F_D, F_L: distribution function of D, D_L;
- f_D, f_L: density of D, D_L;
- α: maximal fraction of lost sales or back orders;
- q: costs per back order or lost sale (in which case it equals $p - k$).

We assume that all variables are non-negative.

Different types of inventory policies exist. The best known are the (s, S) and

(r, Q) policies. The (s, S) policies are used for periodic-review models, the (r, Q) for continuous-review models. The meaning of the letters is as follows:
- s: the inventory level below which or at which an order is placed at an order instant;
- S: the order upto level (that is, the difference between S and the current inventory level is ordered);
- r: the inventory level at which an order is placed the moment it is reached;
- Q: the order quantity.

Example 6.1.1 Let $s = r = 5$, $Q = 20$ and $S = 30$. In the (r, Q) model or order of size 20 is placed the moment only 5 items are left over. In the (s, S) model we wait for the first order moment after the moment the inventory dropped to 5 (these moments have to be specified). Let this level be at 4; then $30 - 4 = 26$ items are ordered. In the (s, Q) model always 20 items are ordered at an order instant at which the current inventory is 5 or lower.

6.2 Single-order model

We discuss in this section the single-order model. We use the notation introduced above: demand D with distribution function F_D, costs q per order that cannot be met, and costs h for each left-over items. We have order costs K and a current inventory of y. S is the inventory after ordering, the *order up to level*. The actual order size is thus $S - y$. Assume $S > y$. Then the total costs $C(S)$ are:

$$C(S) = K + h\mathbb{E}(S - D)^+ + q\mathbb{E}(D - S)^+. \tag{6.1}$$

We could interpret $h\mathbb{E}(S - D)^+ + q\mathbb{E}(D - S)^+$ as the price we have to pay for the randomness in demand. If D were deterministic and $S = D$, then this term would completely disappear.

In the next theorem we calculate the optimal value of S.

Theorem 6.2.1 (Single-order model)
i) The optimal order size S^* that minimizes total holding and backorder costs in the single-order model in case $K = 0$ and no initial inventory is given by

$$S^* = F_D^{-1}\Big(\frac{q}{q+h}\Big) \tag{6.2}$$

in the case of D continuous and

$$S^* = \arg\min_S\{F_D(S) \geq \frac{q}{q+h}, S \text{ integer}\}$$

in the case of D discrete.
ii) The optimal order size S^* that minimizes total holding and backorder costs in the single-order model in case $K = 0$ for initial inventory y is given by $(S^* - y)^+$;
iii) If $K > 0$ and $y < S^*$, then it is optimal to order $S^* - y$ if $C(S^*) < C(y) - K$, otherwise no order should be placed;
iv) In the case of minimizing holding costs with the fraction of lost sales below or equal to α the optimal order size S^* is given by the solution of

$$\mathbb{E}(D - S^*)^+ = \alpha\mathbb{E}D$$

in the case of D continuous and

$$S^* = \arg\min_S\{\mathbb{E}(D - S)^+ \leq \alpha\mathbb{E}D, S \text{ integer}\}$$

in the case of D discrete.

Proof If the inventory after ordering but before sales is equal to S, then the sum of inventory and lost sales costs are equal to $C(S)$. Let us consider first the case that D is continuous, thus f_D exists. Then

$$\frac{d\mathbb{E}(S-D)^+}{dS} = \frac{d}{dS}\int_0^S (S-x)f_D(x)dx = \frac{d}{dS}S\int_0^S f_D(x)dx - \frac{d}{dS}\int_0^S xf_D(x)dx =$$

$$\frac{d}{dS}SF_D(S) - Sf_D(S) = F_D(S).$$

Similarly,

$$\frac{d\mathbb{E}(D-S)^+}{dS} = -(1 - F_D(S)).$$

Differentiating again gives

$$\frac{d^2\mathbb{E}(S-D)^+}{(dS)^2} = \frac{d^2\mathbb{E}(D-S)^+}{(dS)^2} = f_D(S) \geq 0.$$

Thus $C(S)$ is convex, and the global minimum can be obtained by solving $\frac{dC(S)}{dS} = 0$ which leads to $hF_D(S) - q(1 - F_D(S)) = 0$, resulting in (6.2). To arrive at this situation $(S^* - y)^+$ items should be ordered. If $K > 0$ then the same order quantity is optimal, but then total costs with ordering $C(S^*)$ should be compared with $C(y) - K$, which corresponds to not ordering.

If D is discrete then $\mathbb{E}(S+1-D)^+ - \mathbb{E}(S-D)^+ = F_D(S)$ for S integer. Again, $C(S) : \mathbb{N} \to \mathbb{R}$ is convex, and $C(S+1) - C(S) = hF_D(S) - q(1 - F_D(S))$. For the convexity it follows that the optimal order quantity S^* is given by $S^* = \arg\min_S\{C(S+1) - C(S) \geq 0\} = \arg\min_S\{F_D(S) \geq q/(q+h)\}$. For point ii) and iii) the same arguments apply as for the continuous case.

Concerning iv) it should be noted that holding costs are increasing in S, and thus S^* should be the minimal value for which the constraint is satisfied. □

Equation (6.2) has a nice interpretation. Consider some S for which

$$hF_D(S) < q(1 - F_D(S)). \tag{6.3}$$

Should we increase S or decrease S? If we increase S by δ, then $F_D(S)$ is the probability that the additional δ are left-over. Thus the marginal left-over costs are $\delta h F_D(S)$. On the other hand, the back-order costs are reduced by ordering more. $1 - F_D(S)$ is the probability that back-orders will occur if the order size is S, thus $-\delta q(1 - F_D(S))$ are the marginal back-order costs. If (6.3) holds then we can reduce costs by increasing S. We do this until we found S^* for which

$$hF_D(S^*) = q(1 - F_D(S^*),$$

which is equivalent to Equation (6.2).

The best-known single-period or single-order model is the *newsvendor* or *newsboy* problem. For the newsvendor there are order costs k per item, a selling price p and *salvage value* v, the amount received for left-over items. When the newsvendor orders S then his profit is given by:

$$P(S) = (p-k)\mathbb{E}\min\{D, S\} + (v-k)\mathbb{E}(S-D)^+.$$

Now take $q = p - k$, the profit per item sold, and $h = k - v$, the amount paid for left-over items. Then $P(S) = q[\mathbb{E}D - \mathbb{E}(D-S)^+] - h\mathbb{E}(S-D)^+ = q\mathbb{E}D - C(S)$.

Maximizing $P(S)$ corresponds to minimizing $C(S)$. Thus Theorem 6.2.1 applies also to this model, and the optimal order level is given by

$$S^* = F_D^{-1}\Big(\frac{p-k}{p-v}\Big).$$

6.3 Multi-order deterministic-demand models

Now we continue with multi-order models, starting with the deterministic-demand continuous-review model. We also discuss the differences with the periodic-review model. In the next section we deal with models with stochastic demand.

The most famous result from inventory theory is the *Economic Order Quantity* (EOQ). One of the reasons for its popularity is its simplicity. In the standard version it concerns a deterministic-demand continuous-time continuous-product model without lost sales or back orders and with fixed order costs $K > 0$, where the objective is to minimize the long-run average sum of order and inventory costs. Note that because shortselling is not allowed there is no need for parameters related to backordering, purchasing, and selling: all demand needs to be satisfied immediately. Thus the only relevant parameters are λ, h, and K.

Note the difference between physical and economic inventory: economic inventory includes orders that are not yet delivered.

Theorem 6.3.1 (EOQ model) *In the deterministic-demand model with continuous review and lead time L the optimal policy orders a quantity Q^* (the so-called* Economic Order Quantity*) given by*

$$Q^* = \sqrt{\frac{2K\lambda}{h}} \tag{6.4}$$

at (economic) inventory level λL; the total average inventory and order costs are given by $C(Q^*) = \sqrt{2K\lambda h}$.

Proof Due to the lack of randomness in the model the optimal order point is 0, there is no need for *safety stock*. The parameters do not change, thus the order quantity Q is equal for each order cycle. We study the average costs over a single cycle, which is, by renewal theory (see Section 3.3), equal to the long-term average costs. Denote with T the time between two consecutive orders. Then $T = Q/\lambda$.

Denote the average costs for order quantity Q with $C(Q)$. It consists of order costs K/T and inventory costs. The inventory level decreases linearly from Q to 0, therefore the average

inventory level is $Q/2$, and the average inventory costs $hQ/2$. Thus

$$C(Q) = \frac{K}{T} + \frac{hQ}{2} = \frac{K\lambda}{Q} + \frac{hQ}{2}.$$

We readily see that C is convex, and that $\lim_{Q\downarrow 0} C(Q) = \lim_{Q\to\infty} C(Q) = \infty$. Therefore $\frac{d}{dQ}C(Q) = 0$ has Q^* as solution which leads to Equation (6.4).

When we order when there are $L\lambda$ items on stock, then the order arrives when the inventory reaches 0. When $L > Q^*/\lambda$, then we should consider the economic inventory, because there is still an order underway. □

The model of Theorem 6.3.1, to which we shall refer as the EOQ-model, has many interesting consequences. We discuss a number of them.

Economies of scale The inventory that is held in the EOQ model is called *cycle stock*. When $K = 0$ then ordering infinitely often would be optimal, and no stock would he held at all. Thus cycle stock exists because of the economies of scale due to ordering large quantities.

It is interesting to calculate the inventory costs per item. This is for example useful when determining the price of a product. The minimal inventory costs per item are given by $C(Q^*)/\lambda = \sqrt{2Kh/\lambda}$. This is a decreasing function of λ, thus we see economies of scale: if λ is big then we can frequently order big quantities, making order and inventory costs per item go to 0.

Robustness Let us study the robustness under changes of the parameters, first for the case $L = 0$. Assume for example that we underestimated λ by a factor 2, i.e., we took as order size $Q' = \sqrt{K\lambda/h}$ instead of $Q^* = \sqrt{2K\lambda/h}$. Then we find as relative error:

$$\frac{C(Q') - C(Q^*)}{C(Q^*)} = \frac{\frac{K\lambda}{\sqrt{K\lambda/h}} + \frac{h\sqrt{K\lambda/h}}{2} - \sqrt{2K\lambda h}}{\sqrt{2K\lambda h}} = \frac{3}{2\sqrt{2}} - 1 \approx 0.06.$$

The same result holds if we overestimate λ by a factor 2, thus if we take order size $Q' = \sqrt{4K\lambda/h}$:

$$\frac{C(Q') - C(Q^*)}{C(Q^*)} = \frac{\frac{K\lambda}{2\sqrt{K\lambda/h}} + h\sqrt{K\lambda/h} - \sqrt{2K\lambda h}}{\sqrt{2K\lambda h}} = \frac{3}{2\sqrt{2}} - 1 \approx 0.06.$$

From the form of the formula it is readily seen that the same result also holds for K and h. Thus we come to the surprising conclusion that if make an error of less than

a factor 2 in estimating any of the input parameters λ, K, or h then the relative error does not exceed 6%.

Now consider the case that more than one parameter changes. The effects might cancel out, but in the worst case the costs increase quickly: if λ and K are both twice as high as foreseen then the error is 25%.

If $L > 0$ then the result for variations in K and h remain the same. However, when λ is over or underestimated it can have dramatic consequences. Overestimating λ in this case leads not only to a wrong choice of Q, but also to orders arriving when there is still stock left. Underestimating λ leads to lost sales and/or backorders. To be more precise, when λ is underestimated by a number λ_d then there are no items to meet the demand for a total of $\lambda_d L$ items per order cycle. Thus a fraction $\lambda_d L/Q^*$ of the demand is not met. On the other hand, if λ is overestimated by a number λ_d then the order arrives when there is still $\lambda_d L$ inventory. If no measures are taken then the average costs are increased by $\lambda_d L h$.

Periodic models Variations in order lead time are also interesting to study, because of the connection with periodic models. Define $\hat{C}(T) = C(T\lambda) = C(Q)$, i.e., $\hat{C}$ are the costs as a function of the lead time T instead of the order size Q. Consider $T^* = Q^*/\lambda$. First note that in general, for $\alpha > 0$, $\hat{C}(\alpha T^*) = \frac{1}{2}(\alpha + \frac{1}{\alpha})\hat{C}(T^*)$. It follows that $\hat{C}(\frac{1}{\sqrt{2}}T^*) = \hat{C}(\sqrt{2}T^*)$ and that

$$\frac{\hat{C}(\frac{1}{\sqrt{2}}T^*) - \hat{C}(T^*)}{\hat{C}(T^*)} = \frac{\hat{C}(\sqrt{2}T^*) - \hat{C}(T^*)}{\hat{C}(T^*)} = \frac{1}{2}(\sqrt{2} + \frac{1}{\sqrt{2}}) - 1 \approx 0.06.$$

We see that choosing a lead time T within the interval $[\frac{1}{\sqrt{2}}T^*, \sqrt{2}T^*]$ can only increase costs by 6%. This can be very convenient. Assume for example that the required order lead time in a certain situation is 10 days, but that we prefer to order once a week or once every two weeks. The 6%-interval is $[7.07, 14.14]$, thus by ordering every week we increase the costs by a little over 6% and by ordering every two weeks we increase them by a little less than 6%. This way of reasoning is known in the literature as the *powers of 2* solution, because the intervals considered have the form $[T, 2T]$. Note that the result can also be formulated in terms of Q: its 6%-interval is given by $[\frac{1}{\sqrt{2}}Q^*, \sqrt{2}Q^*]$.

Backorders Backorders or lost sales are usually a consequence of stochastic demand: inventory costs would be too high if the probability of backorders or lost

sales had to be reduced to (almost) 0. However, also in the deterministic EOQ model it can be advantageous to have a small fraction of back orders. See Exercise 6.5 for an example with a fixed order size.

Remark 6.3.2 (discrete demand) So far we assumed that products are indivisible and that demand is linear. Even if products are discrete (and demand is a step function) Q^* is often a very good approximation, especially if Q^* is big. If we want to model the discrete nature of the products explicitly, then we have to decide when the order is placed: at the moment inventory becomes 0, or $1/\lambda$ time units later when the next demand is placed? The difference between the two is having, on average, half a unit of product more or less in stock compared to the continuous demand model. The optimal order size remains equal; however, it should be rounded to an integer. As long as this integer falls within the interval $[\frac{1}{\sqrt{2}}Q^*, \sqrt{2}Q^*]$, then the error is limited to 6%.

Remark 6.3.3 (production model) The main difference between queueing and inventory models is that in the former capacity plays a major role, and lot sizes in the latter. Let us extend the EOQ-model to deal not only with lot sizes but also with limited capacity, by introducing a production rate p, with $p > \lambda$. If the order size is Q, then production occurs during the first Q/p time periods. The highest inventory position is reached when production stops, with level $Q(p-\lambda)/p$. This leads to average costs

$$\tilde{C}(Q) = \frac{K\lambda}{Q} + \frac{hQ(p-\lambda)}{2p}.$$

Similar arguments as in the proof of Theorem 6.3.1 lead to the optimal order quantity $\tilde{Q}^*$:

$$\tilde{Q}^* = \sqrt{\frac{2pK\lambda}{(p-\lambda)h}}.$$

Note that $\tilde{Q}^* > Q^*$.

6.4 Multi-order stochastic-demand models

In this section we assume, in contrast with the previous section, that the demand is stochastic and also stationary. If the order lead times were 0, then there would be no reason to change the re-order policy: if the inventory becomes 0, then an order can be placed immediately. Of course the time between orders and the costs become random variables: therefore we take the average expected costs as criterion. There are no crucial differences with the deterministic-demand model: ordering the EOQ

when inventory is 0 is still optimal. It is not that simple anymore if we assume a non-zero order lead time L. In this case demand might occur while the inventory level is already 0. In this section we first assume that this demand is back ordered to the moment that the next order arrives. We consider both the cases where there is a constraint on the fraction of back orders and where the costs of back orders are part of the overall cost function. Both situation will result in ordering such that the expected stock level is non-zero when the order arrives. This stock is called the *safety stock*.

We assume that during the lead time L the demand D_L has the distribution function F_L with expectation λL. In the deterministic model of the previous section the order has to be placed when the inventory level is at λL. We call this the order level r. For the current stochastic model r is not necessarily equal to λL; often we take r higher to avoid back orders, and sometimes it can be optimal to take r smaller than λL, for example if back ordering is cheaper than keeping safety stock.

Theorem 6.4.1 *In the stochastic-demand model with continuous review and lead time L the policy that minimizes order and inventory costs under a constraint on the fraction of backorders orders or lost sales a quantity Q^* at (economic) inventory level r^* approximated by*

$$Q^* \approx \sqrt{\frac{2K\lambda}{h}} \text{ and } b(r^*) \approx \alpha Q^*$$

with α the maximal fraction of backorders or lost sales and $b(r) = \mathbb{E}(D_L - r)^+$ the number of backorders during L with initial inventory r;
The policy that minimizes order, inventory and backorder costs orders a quantity Q^* at (economic) inventory level r^* approximated by

$$Q^* \approx \sqrt{\frac{2\lambda(K + qb(r^*))}{h}} \text{ and } r^* \approx F_L^{-1}\Big(1 - \frac{2hQ^*}{2q\lambda + hQ^*}\Big);$$

The policy that minimizes order, inventory and lost sales costs orders a quantity Q^* at (economic) inventory level r^* approximated by

$$Q^* \approx \sqrt{\frac{2\lambda(K + qb(r^*))}{h}} \text{ and } r^* \approx F_L^{-1}\Big(1 - \frac{hQ^*}{q\lambda + hQ^*}\Big).$$

Proof Let us quantify the different aspects of the system with backorders, for a policy that orders Q units if the inventory level reaches r. The expected length of the time between two deliveries remains Q/λ. Therefore the fixed order costs are again $K\lambda/Q$. The expected

number of back ordered items per cycle is denoted as $b(r)$. It can be derived from F_L: $b(r) = \int_r^\infty (x-r)dF_L(x)$. Determining the average inventory in the system is more complicated. First we make a distinction between the physical inventory, which is the inventory actually at stock, and the inventory level including back orders, which can therefore be negative. A good approximation of the average physical inventory is the average of the physical inventory at the beginning and the end of the cycle. At the end of the cycle this is $r - \lambda L + b(r)$. At the beginning of the cycle this is $r - \lambda L + Q$. (We assume that this quantity is positive.) Thus we approximate the average positive inventory with $\frac{1}{2}(r - \lambda L + Q + r - \lambda L + b(r)) = r - \lambda L + (Q + b(r))/2$.

We start with the service level formulation. We first consider the system where we minimize the sum of order and holding costs $C(r, Q)$ under the service level restriction that the probability that an arbitrary unit is back ordered is not bigger than α. This gives as minimization problem: $\min_{r,Q}\{C(r,Q)|b(r)/Q \leq \alpha\}$, with $C(r,Q) = K\lambda/Q + h(r - \lambda L + (Q + b(r))/2)$. In general, this problem is hard to solve. However, if α is small, then $Q^* = \sqrt{2K\lambda/h}$, the EOQ, is a good approximation; r^* should be chosen such that $b(r^*) = \alpha Q^*$. The interpretation of the average inventory is straightforward: $Q^*/2$ is the cycle stock, as for the deterministic model, and $r^* - \lambda L$ is the safety stock that is present to avoid too much back orders. Note that safety stock can be negative; this can easily be seen if demand is deterministic and $\alpha > 0$.

Next we consider the single objective formulation with backorders. Now consider the case where the additional costs for each back ordered item are equal to q. Then the total costs can be approximated as follows:

$$C(r,Q) \approx \frac{K\lambda}{Q} + h\Big(r - \lambda L + \frac{Q}{2}\Big) + \Big(\frac{q\lambda}{Q} + \frac{h}{2}\Big)b(r).$$

Differentiating to r and Q gives as minimal values r^* and Q^*:

$$Q^* = \sqrt{\frac{2\lambda(K + qb(r^*))}{h}},\ r^* = F_L^{-1}\Big(1 - \frac{2hQ^*}{2q\lambda + hQ^*}\Big).$$

When compared to the EOQ, we see that K is replaced by $K + qb(r^*)$ in the expression of Q^*. Thus the fixed order costs K are augmented with the back order costs per cycle. Again, these optimal values are not simple to calculate. However, an iterative scheme, starting with Q^* approximated by the EOQ, converges fast to the optimal solution.

Lost sales can be modeled similarly: q are now the costs of lost sales. The main difference in the approximation is the level after the order arrival: as there are no back orders to fulfill this is on average equal to $r - \lambda L + b(r) + Q$. Therefore

$$r^* = F_L^{-1}\Big(1 - \frac{hQ^*}{q\lambda + hQ^*}\Big).$$

Another difference is the average cycle length, which becomes $(Q + b(r))\lambda$. Modeling this would complicate the solution considerably. □

Periodic review with $K > 0$ As for the deterministic models, we can consider periodic review models. The principle change is that the safety stock is not only used to bridge the order lead time, but also the order period. Thus F_L has to be replaced by F_{L+T}, where T represents the remaining time until the next order moment. This gives a new level s instead of r, such that as soon as the inventory drops below s then an order should be placed at the next order moment. It is optimal not to order a fixed quantity Q, but up to a level S. Such a policy is called an (s, S) policy.

Periodic review with $K = 0$ Periodic models with $K = 0$, backorders, and lead time L shorter than the period (assumed to be 1) are equivalent to single-period models. Consider an order moment. Then the order after this order arrives at $1 + L$ from now, and there is no reason to order now for demand occuring after $1 + L$, because it is free to order at 1. Theorem 6.3.1 can be used with F the distribution function of the demand during $1 + L$.

6.5 Multi-stage and multi-item models

The sections above give an overview of some of the most important results and concepts in single-stage single-item models.

Often items are stored at different levels. In these cases order policies depend on the inventory at all stages, not just the next stage. Understandably, optimal order policies become very complicated. A good practice is to base decisions on the total downstream stock, the so-called *echelon stock*.

Models with multiple items are also interesting. By combining orders cost reductions can be achieved. Again, optimal policies are very hard to calculate. Often there are two re-order levels for each item. When the lowest level is reached items are ordered. When an order for a certain item is placed, then all items for which there is less at stock than the higher level are ordered as well.

6.6 Further reading

An excellent source for inventory models is Zipkin [168].

The Chapters 1, 2, and 4 of O.R. Handbook 4 Graves et al. [67] deal with subjects related to the ones discussed in this chapter. The same holds for Chapter 12 of Handbook 2 Heyman & Sobel [74], which considers stochastic inventory theory.

Many books on production or logistics consider also inventory models, such as Hax & Candea [72] and Bramel & Simchi-Levi [26].

When it comes to inventory systems, most of the above references deal with periodic models. In this chapter we decided to give a central role to continuous review models, and to deal with periodic models through the powers of 2 approximation. Results on stochastic continuous review models can be found in Johnson & Montgomery [81] and in Chapter 1 of Graves et al. [67].

For multi-stage models we refer to Chapter 2 and 4 of Graves et al. [67].

6.7 Exercises

Exercise 6.1 Consider a newsboy problem with a demand that is normally distributed with expectation $\mu = 20$ and variance σ^2.
a. Find the optimal order quantity for the parameter values $p = 1$, $v = 0.1$, and $k = 0.5$ for variance 1 and 2.
b. Find the optimal order quantity for the parameter values $p = 1$, $v = 0.1$, and $k = 0.6$ for variance 1 and 2.
c. Calculate the costs for each of the four situations that we considered.
d. Give an intuitive explanation of the results.

Exercise 6.2 An agricultural firm harvests K kilograms of a certain product. The company has two ways to sell their product: to a supermarket at a price p_r per item or at a market at a price p_m. The supermarket will buy all the firm is willing to sell them, the demand at the market D is random. Leftover products are worthless.
a. Formulate your expected income as a function of the amount of product that you sell to the supermarket.
b. Give the policy that maximizes your expected income.
c. Calculate the policy for $K = 1000$, $p_r = 0.9$, $p_m = 1.0$, and D is normally distributed with expectation 1100 and standard deviation 300.
d. The management is not only interested in maximizing expected income, but is also risk-averse. What should management do in your opinion? Explain yourself using heuristic arguments.

Exercise 6.3 Consider a continuous-time multi-order deterministic-demand continuous-product inventory model with $\lambda = 5$, $K = 10$, $h = 1$ and $L = 1$.
a. Compute the optimal re-order level and re-order size.
Now demand is stochastic; it occurs according to a Poisson process with rate $\lambda = 5$.

For the rest the system is the same. Items that are not available are backordered.
b. We use the same re-order policy. Estimate the probability that backorders occur in a cycle.
c. It is the objective to avoid backorders in at least 9 out of 10 cycles. How should we choose the re-order policy to achieve this?

Exercise 6.4 It is stated for the EOQ model at page 98 that the error in costs is 25% if λ and K are both twice as high as foreseen. Show this.

Exercise 6.5 A person receives a monthly salary S on a bank account. Each month is assumed to have 30 days. From this bank account she pays each day her daily expenses d. We assume that $30d < S$. She has the option to put money on a savings account. The savings account has a daily interest rate r, there is no interest rate on the account where her salary arrives, unless the amount is negative: then she pays an interest rate of p, $p > r$. Interest is payed at the end of the month (the same day the salary arrives). The day the salary arrives she decides how much money to put on the savings account and the money is imddediately transfered. The first month we start with 0 on the savings account and S on the other account.
a. Model this problem as an inventory model: classify the model and determine the necessary parameters.
b. Calculate the amount to put on the savings account that maximizes the interest at the end of every month. You need not do this exactly, a good approximation is fine. This amount is not equal for every month!
c. Give the definitions of safety stock, cycle stock, and seasonal stock.
d. What type of stock is the money on the standard account?
She uses the money on the savings account to pay for her summer holidays.
e. What type of stock is the money on the savings account?

Exercise 6.6 Consider an inventory model with Poisson(10) demand, lead time 1, $K = 100$, $h = 1$, and maximal 5% backorders. Estimate Q^* and r^*.

Exercise 6.7 Consider an inventory model with $\lambda = 10$, $L = 2$, $D_L \sim N(20, 20)$, $h = 0.1$, $K = 10$ and $q = 1$. Implement in an appropriate tool (for example a spreadsheet) the recursion for Q^* and r^* using the expression for average excess Equation (1.2). Give the optimal values.

Exercise 6.8 A shop sells goods. When ordered at the beginning of day n, the ordered goods arrive at the beginning of day $n + 1$, and they can be sold from that day on. A unit of goods costs p, and is sold for r ($r > p$). Every night that a unit spends

in stock costs h. There are no order costs, orders are therefore placed every day.
a. Model this as an inventory model, by introducing random variables for demand, stock, and order sizes. Give the relations between the variables.
b. Express the expected sales at day $n+1$ and the expected inventory costs at the end of day $n+1$ as a function of the stock at the beginning of day n (including the arriving order) and the order placed at the beginning of day n.
c. Suppose that the demand on every day are uniformly distributed on $[0,1]$. Calculate the expected sales at day $n+1$ and the inventory costs of the stock at the end of day $n+1$, given the stock at the beginning of day n (including the order that just arrived), and the order placed at the beginning of day n.
d. Let R be the order policy that miximizes for each day n expected profit minus inventory costs at day $n+1$. Do you think that this order policy maximizes the average expected profit minus inventory costs? Motivate your answer!
The same shop also sells perishable goods. When ordered at the beginning of day n, the ordered goods arrive at the beginning of day $n+1$, and they can be sold during days $n+1$ and $n+2$. After that they are thrown away, without cost nor reward. A unit of goods costs p, and is sold for r. There are no order costs.
e. Model this as an inventory model, by introducing random variables for demand, stock, and order sizes. Give the relations between the variables.

Exercise 6.9 Consider a company with 10 outlets. Demand of a certain products occurs at each outlet according to independent Poisson processes, with average 10 each day at each outlets. Orders can be placed once a week, replenishments are immediate (they are done overnight). Only 1% of lost sales are accepted.
a. How many items should each outlet have in stock after each replenishment?
Consider a situation where all deliveries are done from a central location, which is like having one outlet with a daily expected demand of 100.
b. How many items should be in inventory in this situation?
Consider finally the more realistic situation where there is a central warehouse that delivers to the outlets every day. These replenishments occur overnight: what is ordered at the end of the day can be sold at the beginning of the next day. The central warehouse is replenished every week. Thus the outlets should only keep one day of stock, the warehouse for one week.
c. Give an approximation for the stock at both location such that the solution under a is both outperformed in stock costs and order reliability.
(Of course transportation costs can be higher: ten weekly long-distance shipments are replaced by one weekly long-distance and daily short-distance shipments.)

Exercise 6.10 A distribution system consists of 10 outlets and 1 DC. Outlets order in

lots of 10, on average once a week (a week consists of 5 working days). Demand from the outlets is approximately Poisson. Delivery is immediate. The order lead time to the DC is 1 week, and is done with an order quantity of 400.

a. What should be the safety stock at the DC to have less than 1% backorders?

The outlets change policy: they order with a lot size of one, but with the same average total demand.

b. What should, in this case, be the safety stock at the DC to have less than 1% backorders?

It is determined that the optimal lot size is 5 for orders from the outlets at the DC.

c. Propose a way to manage the inventory at the DC such that the safety stock is as low as possible.

Chapter 7

Optimization

In this chapter we discuss optimization. In the previous chapters we focused on the performance analysis of stochastic processes, and some very specific optimization problems for inventory systems. Our ultimate goal is to consider very general optimization problems based on stochastic processes. We discuss if and how we can solve these problems, starting from the standard deterministic linear optimization problems.

Every subject in this chapter is itself the subject of many dedicated books and hundreds or even thousands of articles in the scientific literature. By no means we can give a comprehensive introduction to all these subjects. Our goal is to show the relations between the different problems, give a flavor of their solution techniques, and discuss the connections with the stochastic models discussed in the previous chapters.

7.1 Framework

Many optimization problems of interest to us are of the following form:

$$\min\{g(x)|x \in S\}, \tag{7.1}$$

for $g : \mathbb{R}^n \to \mathbb{R}$ and $S \subset \mathbb{R}^n$. In this chapter we will discuss the most common special cases and their solution techniques. These techniques differ because of the properties of g and S. For example, for a given x, it might be that $g(x)$ can only be approximated using simulation, or that g is a function known in an explicit form for which even the derivative is available. Note that it can also be the case that a solution to (7.1) does not exist.

Example 7.1.1 Suppose we have to split a fixed amount s of service capacity between k services, where the performance of each service is measured through the Erlang C formula (see page 69). The objective is to maximize the overall weighted performance. This gives the following optimization problem:

$$\max\Big\{ \sum_{n=1}^{k} w_n \mathbb{P}(W_Q^n(s_n) \leq t) \;\Big|\; s_n > \frac{\lambda_n}{\mu_n},\ s_n \in \mathbb{N},\ \sum_{n=1}^{k} s_n = s \Big\},$$

with w_n the weight of service n, $W_Q^n(x)$ the waiting time of service n when the assigned capacity is x, t the acceptable waiting time, and λ_n and μ_n the parameters of service n. The condition $s_n > \lambda_n/\mu_n$ assures that every service is stable. As minimization is equivalent to maximization by multiplying the objective g with -1 the problem is in the standard form (7.1).

In this example S is finite and $g(x)$ is a non-linear function defined for all $x \in S$. Note that S can be empty when there is too little capacity.

It is quite common that S is not explicitly known, but that S can be formulated implicitly as $S = \{x|L(x) \in U\}$ with $L : \mathbb{R}^n \to \mathbb{R}^m$ a known function and $U \subset \mathbb{R}^m$ a known set. Equation (7.1) then becomes

$$\min\{g(x)|L(x) \in U\}. \tag{7.2}$$

A well-known example is *linear optimization* (best known as *linear programming*, but the naming conventions are changing). In this situation $g = c^T x$ for c some n-dimensional vector, $L(x) = Ax$ for A an $m \times n$ matrix, and $U = \{x|x \leq b\}$ with $b \in \mathbb{R}^m$. This leads to $S = \{x|Ax \leq b\}$.

Example 7.1.2 In Theorem 4.2.3 linear equations are given for finding the stationary distribution π of a Markov process. Now suppose that the transition rates λ are linear functions of one or more parameters. Then minimizing a linear function of the stationary distribution is an linear optimization problem, with the linear equations of Theorem 4.2.3 as constraints. More common examples are given in the next section.

Linear optimization is an example where S is a closed set. Example 7.1 is one of many problems with a countable or finite set of solutions S. An important class of problems that has aspects of both is *mixed-integer linear optimization* (MILO): it has a linear g, constraints of the form $Ax \leq b$, and additionally $x_i \in \mathbb{N}$ for $i \in A \subset \{1, \ldots, n\}$.

Certain problems that can be formulated as MILO problems are very hard to solve, even though evaluating $g(x)$ and $L(x)$ for a given x can be done very rapidly.

Problems where g can only be evaluated using simulation are even more challenging. Sometimes also $L(x)$ needs to be approximated by simulation. Both problems with integer variables and with continuous variables are of interest.

A different class of optimization problems should perhaps be called stochastic dynamic optimization problems, but which are known under the names *stochastic dynamic programming* and *Markov decision problems*. These are dynamic problems in which a decision has to be taken at every point in time depending on the evolution of the process up to then. As the first name already suggests dynamic programming is the main solution method.

In the subsequent sections the different types of problems are discussed one by one.

Remark 7.1.3 In many optimization problems we have multiple objectives, but we need a single function to optimize. There are two solutions to this issue: either we combine the objectives in a single objective function by taking a weighted average, or some of the objectives get a maximum allowed value by adding them as constraints.

Example 7.1.4 The EOQ model of Section 6.3 is an example of a model with two objectives of which the (weighted) sum taken: holding costs and order costs. Another example is the following. In Section 6.4 we discussed inventory models with order costs, holding costs, and backorders or lost sales. In Theorem 6.4.1 we formulated different inventory models with different objectives, for example minimizing total weighted costs and minimizing order and holding costs under a constraint on the fraction backorders.

7.2 Linear optimization

The standard linear optimization (linopt) problem is as follows: $\min\{c^T x | Ax \leq b\}$, with c an n-dimensional vector, A an m-by-n matrix, and b an m-dimensional vector. Very efficient algorithms exist to solve this type of problem. The best known is the *simplex algorithm*, invented by G.B. Dantzig during World War II. S can be seen as the intersection of m half spaces. Where n half spaces intersect there is a "corner". The simplex algorithm consists of moving from corner to corner until the minimal-cost corner is found. Because of the linearity the optimum can be found in one of the corners.

Nowadays there are even more efficient algorithms, known as "interior-point methods", which allow problems to be solved with thousands of variables and constraints. The different solution methods are implemented in optimization modules that can be called from within Excel, special-purpose modeling environments, and

most programming languages. Because of this implementing linear optimization has become the work of a few specialists, while millions of people worldwide use their software. See Chapter 10 for more details.

Example 7.2.1 The archetypical linear optimization problem is the *product mix problem*. A factory has to decide which products to produce, given resource limitations. This problem has the form $\max\{p^T x | Ax \leq b, x \geq 0\}$ where x_i is the amount of product produced of item i, p_i is the profit per unit produced, $a_j i$ is the amount of resource j required to produce 1 item of product i, and b_j is the amount available of resource j.

7.3 Convex optimization

In this section we study problems of the form $\min\{g(x)|x \in S\}$ where g is a *convex function* and S is a *convex set*. We call a function $f : \mathbb{R}^n \to \mathbb{R}$ convex if $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ for all x, y and $\alpha \in (0,1)$. A set S is convex if it follows from $x, y \in S$ that $\alpha x + (1-\alpha)y \in S$ for all $\alpha \in (0,1)$. For these *convex optimization* problems it can already be challenging to consider $\min g(x)$, so without additional constraints. Note that for linear problems this problem is unbounded unless g is constant. The way these problems are solved is by progressively, in a discrete way, moving towards the minimum. To do this, for some current solution x, we need to determine the direction in which we want to move and the stepsize. It is common to take the direction with the steepest descent; to determine this we need the gradient ∇g, which is the vector of derivatives in all dimensions. Then, to determine the step size, we need to determine the point where the convex function in the direction of the steepest descent curves back. For this we need to know the second derivative. However, in many cases we do not even know ∇g explicitly, we can only evaluate $g(x)$ in any given point x. Therefore, estimating the direction of the steepest descent and the step size in this direction are the main challenges of convex optimization.

Example 7.3.1 Assume we have a k-dimensional newsvendor problem: we have to split S items between k locations, location n having a demand with distribution function F_n. There is no possibility of relocating items. We will denote with $C_n(s)$ the total costs for location n when s items are assigned to it. Note that in the proof of Theorem 6.2.1 it was shown that C_n is a convex function. When the assignment to each location can be non-integer, then we have a convex optimization problem. Let us formulate it for $k = 2$:

$$\min\left\{C_1(s_1) + C_2(s_2)\middle| s_1, s_2 \geq 0,\ s_1 + s_2 = S\right\} = \min\left\{C_1(s) + C_2(S-s)\middle| 0 \leq s \leq S\right\}.$$

In the second formulation we reduced the dimension of the problem to 1. Note that $C_1(s) + C_2(S-s)$ is a convex function of s.

We first give an algorithm for the case that $S = \mathbb{R}^n$.

Algorithm for unconstrained convex optimization
0. Set $k = 0$, choose x_0
1. Determine or approximate $\nabla g(x_k)$
2. Determine a step size γ_{k+1}
3. $x_{k+1} = x_k - \gamma_{k+1}\nabla g(x_k)$
4. Stop if required precision is reached;
   otherwise: $k = k + 1$ and go to step 1

There are multiple ways to choose the series γ_k. Here are a few common choices:
- through a line search in the direction $\nabla g(x_k)$, which means finding numerically $\gamma_k = \arg\min_\gamma g(x_k - \gamma \nabla g(x_k))$;
- by estimating or computing the second derivative in the direction $\nabla g(x_k)$, and then using the minimizer as if the function along the gradient is linear (the so-called *Newton step*);
- by taking a series that guarantees converges to the minimum such as $\gamma_k = 1/k$.

These methods can also be applied to non-convex functions and non-convex feasible sets. Evidently, in such a situation no convergence to the global minimum is guaranteed. To augment the changes of finding a good or even the global minimum it is advised to restart the algorithm several times from different points.

We finally look at the situation where $S \neq \mathbb{R}^n$. There are two ways to introduce this in the algorithm:
- by adapting step 3 as to make sure that all $x_k \in S$;
- by changing g such that the value is higher outside of S, by adding a penalty function.

For both approaches there are several possibilities, depending on other choices made in the algorithm. A simple penalty function is $C\mathbb{I}\{x \notin S\}$ with $C \gg 0$. A disadvantage is that this function is not convex. An example of a convex penalty function is as follows. Assume $S = \{x | L_i(x) \leq b_i\}$. Then the function $C\sum_i \max(0, L_i(x) - b_i)$ is convex, 0 on S, and > 0 when $x \notin S$. Another example of a convex penalty function is $C\sum_i [\max(0, L_i(x) - b_i)]^2$, which has the additional advantage that it is differentiable.

If step 3 is adapted to avoid moving out of S then we can restrict the line search to the part that is in S, which is always connected because of the convexity of S. For the

other choices we modify $x_{k+1} = x_k - \gamma_{k+1}\nabla g(x_k)$ into $x_{k+1} = \Pi_S(x_k - \gamma_{k+1}\nabla g(x_k))$, with Π_S some projection into S.

Example 7.3.2 We continue Example 7.3.1. Take for example as starting solution $x_0 = S/2$. We use Equation (6.1) to determine $g(x_0 - h)$ and $g(x_0 + h)$, with $g(x) = C_1(x) - C_2(S - x)$. Then we take

$$x_1 = \Pi\Big(x_0 - \gamma_1 \frac{g(x_0 + h) - g(x_0 - h)}{2h}\Big)$$

with $h > 0$ small, $\gamma_k = 1/k$ for example and Π the projection on $[0, S]$, that is,

$$\Pi(x) = \begin{cases} 0, & x < 0; \\ x, & 0 \leq x \leq S; \\ S, & x > S. \end{cases}$$

We repeat this procedure until $|x_{k+1} - x_k|$ is very small.

The performance of engines for convex optimization has drastically improved in recent years, making them almost as efficient as linear optimization engines.

7.4 Mixed-integer linear optimization

Sometimes a problem can be formulated as a linear optimization problem but with additional integral of binary constraints of the form $x_i \in \mathbb{N}_0$ or $x_i \in \{0, 1\}$. When all decision variables are integer or binary then S is countable, and we might use the general method for countable state spaces discussed the next section. However, when the problem is indeed a linear optimization problem with additional integer/binary constraints then the solution method can make use of this. The idea is as follows. Suppose we solve the problem without integer constraints and integer variable x_i has a non-integer value a. Then we can make two new subproblems: one in which we add $x_i \leq \lfloor a \rfloor$, and one in which we add $x_i \leq \lceil a \rceil$. Now our original problem with integer constraints is equivalent to the best subproblem with integer constraints, and we will not get solutions anymore with $x_i \in (\lfloor a \rfloor, \lceil a \rceil)$. This we call *branching*. We can do this recursively until all solutions satisfy the integrality constraints. The one with the lowest value is the optimal solution.

Sometimes the structure of a problem is such that we find integral solutions right away. When this is not the case we might have to execute the linear optimization calculation at least $2^{k+1} - 1$ times before all variables of all subproblems are integer. It can be even more because the same variable might occur multiple times.

Example 7.4.1 Consider

$$\max\left\{3x_1+5x_2 \;\middle|\; \begin{array}{l} 2x_1+3x_2\leq 7 \\ x_1,x_2\in\mathbb{N}_0 \end{array}\right\}.$$

The unconstrained problem has as solution $(0,7/3)$. Thus we branch to two subproblem: one with constraint $x_2\geq 3$, the other with $x_2\leq 2$. The former has no solution, the latter gives

$$\max\left\{3x_1+5x_2 \;\middle|\; \begin{array}{l} 2x_1+3x_2\leq 7 \\ x_2\leq 2 \\ x_1,x_2\in\mathbb{N}_0 \end{array}\right\}.$$

Now the optimal solution is $(1/2,2)$. We add the constraints $x_1\leq 0$ and $x_1\geq 1$. The former gives the integer solution $(0,2)$ with value 10, the latter gives $(1,5/3)$. Branching again for variable x_2 finally gives the optimal solution $(2,1)$ with value 11.

The procedure as described can be numerically very demanding. To reduce computation time we should try to cut off branches of the branching tree. To be able to do this, we should realize the following. Let us suppose we have found one or more integer solutions and let us call the best the *incumbent* and suppose it has value α. Then any branch with value $\geq\alpha$ can be cut off: adding constraints will only lead to a higher value thus it will never become optimal.

Example 7.4.2 A well-known mixed-integer linear optimization problem is shift scheduling as introduced by Dantzig. Suppose there are T intervals in which employees have to be scheduled, at least s_t in interval t. Employees work according to shifts. The K different shifts are characterized by a matrix A for which $a_{tk}=1$ if shift k works during interval t, $a_{tk}=0$ otherwise. Let c_k be the costs of shift k, and x_k the decision variable indicating the number of agents in shift k. Then the problem of finding the optimal schedule can be formulated as:

$$\min\left\{\sum_{k=1}^{K}c_kx_k \;\middle|\; \begin{array}{ll} \sum_{k=1}^{K}a_{tk}x_k\geq s_t, & t=1,\ldots,T \\ x_k\in\mathbb{N}_0, & k=1,\ldots,K \end{array}\right\}. \tag{7.3}$$

A typical application of this is in call centers. The common choice is $s_t=\min_s\{\mathbb{P}(W_Q^t(s)\leq\alpha)\geq\beta\}$, with $W_Q^t(s)$ the waiting time in interval t. This means that the fraction of calls that waits less than α is β, in every interval. For $W_Q^t(s)$ often the Erlang C formula is used.

Example 7.4.3 When items are discrete then it is natural to add integrality constraints to the product-mix example 7.2.1.

Very powerful optimization engines exist for mixed-integer linopt that can handle thousands of variables and constraints. See Chapter 10 for more details.

7.5 Local search

There are many optimization problems for which $|S| < \infty$ or countable. When $|S|$ is small enumeration (trying all $x \in S$) is an option, and sometimes it is possible to formulate the problem as a mixed-integer linopt problems. Often however this is not possible, or the solution time is too long. A general method to solve problems with a countable S is *local search*. Local search is a *heuristic*: an algorithm that is not guaranteed to find the optimal solution.

The central idea is as follows. For each $x \in S$ we define its neighborhood $N(x) \subset S$. We say that a x^* is a local minimum with respect to N if $f(x) \geq f(x^*)$ for all $x \in N(x^*)$. Now a simple algorithm to find a local minimum is as follows: if the current state is x, then its whole neighborhood is searched. If a state y is found with $f(y) < f(x)$, then y becomes current. This is repeated until a local minimum is found.

Example 7.5.1 Consider the *traveling salesman problem* (TSP): a tour (a closed path) along m cities with minimal length has to be found. A common neighborhood structure, *2-opt*, is as follows. For a given tour, select all combinations of two neigboring cities, and make a new tour by reconnecting them the other way around. For example, consider a problem with 5 cities, and current tour 1-3-5-4-2-1. If we take out 1-3 and 4-2 then the following tour can be constructed: 1-4-5-3-2-1. Using this as the basis for our local search algorithm quickly leads to a good solution. However, there is no guarantee that this tour is the global optimum. Another often-used neighborhood is 3-opt. It consists of selecting any combination of 3 connections and considering all ways to reconnect them.

The TSP can also be formulated as a mixed-integer linear optimization problem. This requires a binary decision variable x_{ij} for every pair of cities. The constraints should make sure that there is a single tour that visits every city once. To avoid tours that do not contain all cities there is a constraint for each subset of S. This results in a formulation with around $2^{|S|}$ constraints, making this approach practically infeasible.

Local search converges to a local minimum, except for some rare cases where convergence to the optimal solution can be shown. Therefore it is used in those cases where the optimal solution is hard to find. Note that it is always possible to find the optimal solution, as long as $g(x)$ can be computed for all $x \in S$ and S can be enumerated. However, this can take very long. Evidently, the time it takes for an algorithm to run depends on the type of problem, but also on the size: it makes a big difference if you have a TSP to solve with 10 or 1000 cities. Let m be some parameter indicating the size of the problem, such as the number of cities in the TSP. Local search is especially used in those cases where there is no optimal algorithm that has a running time that is polynomial in m, thus in cases where only algorithms with an

exponential running time are known. The TSP is an example of such a problem. It is a member of the class of *NP-complete* problems, which is the main subject of research in *complexity theory*.

Example 7.5.2 Consider the use of example 7.4.2 in call centers, and assume we are interested in obtaining the waiting time restriction on average over the whole day instead of every interval. Let w_t indicate the overall fraction of calls that arrive interval t. The optimization problem then becomes:

$$\min \left\{ \sum_{k=1}^{K} c_k x_k \;\middle|\; \begin{array}{l} \sum_{t=1}^{T} w_t \mathbb{P}(W_Q^t(\sum_{k=1}^{K} a_{tk} x_k) \leq \alpha) \geq \beta \\ x_k \in \mathbb{N}_0,\ k = 1, \ldots, K \end{array} \right\}.$$

$\mathbb{P}(W_Q^t(s))$ is non-linear function of s, and local search is a good solution mehod. The neighborhood consists of adding, removing and changing (multiple) shifts.

Several methods exist to prevent from getting stuck in a local minimum. As an example, *simulated annealing* allows for a deterioration with a certain probability. The hope is that this allows you to go over a hill and get to a better local minimum. When the parameters are well-chosen and the algorithm can run indefinitely then it can be shown that eventually the global minimum is reached. It is interesting to note that simulated annealing introduces randomness in the way a deterministic problem is solved. This in contrast with the problems of the next sections, where problems that are stochastic in nature are studied.

7.6 Simulation optimization

In this section we assume that $g(x) = \mathbb{E}G(x)$ with G a random variable which distribution depends on x. G can have all kinds of forms: it can have a relatively simple distribution or it can be a performance measure of some discrete-event system. What all problems we consider in this section have in common is that the only way to obtain information about g is through simulation: for a given x we sample $G(x)$ a number of times and use the average as an approximation for $g(x)$.

Example 7.6.1 Next year's balance sheet of a company is made stochastic by giving a distribution for parameters such as interest, sales, etc. There are a number of different decisions to be made. The company is looking for the decision that maximizes the expected profit under constraints for market share, defaulting, etc.

Example 7.6.2 In an emergency department the demand (patients) and the supply (doctors, nurses, etc.) vary during the day. Performance indicators, such as expected waiting time, are functions of a complicated time-inhomogeneous queueing process. Simulation is used to approximate these indicators. When we want to optimize the supply, for example by changing the schedules of doctors and nurses, we have a simulation optimization problem.

Compared to deterministic methods, simulation optimization (simopt) has two disadvantages: even when the optimal solution is considered and simulated, it might not be recognized as the optimum, because of variations in the simulations. For the same reason, we are likely to underestimate the costs of the minimal value.

Example 7.6.3 Consider a situation with $|S| = 10$, $g(x)$ close together for all x, but a high variability in simulation outcomes. Then the best solution found is not necessarily the optimal one and its value is likely lower than $g(x)$.

A solution to this could be to simulate often enough all solution to obtain very accurate solutions, but usually this is too time consuming, we do not have the *simulation budget* for it. Thus, in simopt, we not only have to decide which solutions to consider but also how many effort we spend on simulating them.

Just as their deterministic counterparts simopt problems can be categorized by the form of S. First we consider a problem which is trivial in the deterministic setting, where $S = \{x_1, \ldots, x_k\}$ is an unordered set with k at maximum say a few hundred. If $g(x)$ can be computed easily then $\min\{g(x)|x \in S\}$ can be solved through enumeration. But what to do in the case of simulation? How often should we sample each solution to get a reliable answer? Should we sample each solution equally often or are their smarter methods? This problem is called *ranking and selection*. We will have a closer look at it, also because it nicely shows the main features of simulation optimization.

If we have a *simulation budget* of in total m runs then we could simulate every solution $\lfloor m/k \rfloor$ times and take the solution with the lowest mean. However, this is not very efficient. It is better to simulate every solution first $m_0 < \lfloor m/k \rfloor$ times to have a first estimation of the mean and its estimation error, by calculating the sample mean and variance. This can be done as follows. Let $g_i(x)$ be the ith realization of $G(x)$. Then the sample mean $\hat{g}_n(x)$ and variance $s_n^2(x)$ based on n replications are defined by

$$\hat{g}_n(x) = \frac{1}{n}\sum_{i=1}^{n} g_i(x), \quad s_n^2(x) = \frac{1}{n-1}\sum_{i=1}^{n}\Big(g_i(x) - \hat{g}_n(x)\Big)^2.$$

Solutions with a high sample mean $\hat{g}_{m_0}(x)$ and low sample variance $s^2_{m_0}(x)$ are unlikely to be optimal and should therefore be discarded. The remaining simulation budget can simply be split evenly among the remaining solutions; more advanced rules exist. We discuss a common selection rule.

Selection rule No selection rule can guarantee that the subset with candidate solutions I contains the optimal solution x^*. The objective is to design a rule such that $\mathbb{P}(x^* \in I) \geq 1 - \alpha$ but with I as small as possible. A common rule is as follows, after n simulations of each state:

$$I = \left\{ x \;\middle|\; \hat{g}_n(x) < \hat{g}_n(y) + \Phi^{-1}\big(\sqrt[k-1]{1-\alpha}\big) \frac{\sqrt{s_n^2(x) + s_n^2(y)}}{\sqrt{n}},\ y \neq x \right\}. \tag{7.4}$$

The reason behind the rule is, with $\hat{G}_n$ the average of n independent replications of G and S^2 its variance,

$$\mathbb{P}(x^* \in I) = \mathbb{P}\Big(\hat{G}_n(x^*) < \hat{G}_n(y) + \Phi^{-1}\big(\sqrt[k-1]{1-\alpha}\big) \frac{\sqrt{S_n^2(x^*) + S_n^2(y)}}{\sqrt{n}}, y \neq x^*\Big) \approx$$

$$\prod_{y \neq x^*} \mathbb{P}\Big(\hat{G}_n(x^*) < \hat{G}_n(y) + \Phi^{-1}\big(\sqrt[k-1]{1-\alpha}\big) \frac{\sqrt{S_n^2(x^*) + S_n^2(y)}}{\sqrt{n}}\Big) =$$

$$\prod_{y \neq x^*} \mathbb{P}\Big(\frac{\sqrt{n}\big(\hat{G}_n(x^*) - \hat{G}_n(y)\big)}{\sqrt{S_n^2(x^*) + S_n^2(y)}} < \Phi^{-1}\big(\sqrt[k-1]{1-\alpha}\big)\Big).$$

The approximation is not an equality because the eventualities are not independent. The expression to the left of the "$<$"-sign in the last equality has, thanks to the CLT, approximately a normal distribution with a negative expectation and variance 1. Therefore the probability is likely to be bigger than $\sqrt[k-1]{1-\alpha}$ and therefore $\mathbb{P}(x^* \in I)$ is likely to be bigger bigger than $1 - \alpha$. (This result can be made mathematically precise if we assume that $G(x)$ has a normal distribution and if we use a t-distribution in the definition of I.)

Common random numbers In the selection rule we assumed that $G(x)$ and $G(y)$ are independent: We estimated the variance of $\hat{G}_n(x) - \hat{G}_n(y)$ with $S_n^2(x) + S_n^2(y)$, which is because $\sigma^2(G(x) - G(y)) = \sigma^2(G(x)) + \sigma^2(G(y))$ in the case of independence. Sometimes it is possible to exploit the structure of the model to reduce the variance of $G(x) - G(y)$ by making them positively correlated. Of course, every

replication of $G(x)$ will remain independent. When we can reduce the variance it is the case that G is a random variable of some other random variable that occurs in both $G(x)$ and $G(y)$. Hence the name *common random numbers* (CRN). For example, it might be that all solutions involve the same customer arrival process. It then makes sense to use the trajectory of arrival moment and perhaps also service times for all $G_n(x)$, $x \in S$. This way usually $\sigma^2(G(x) - G(y)) < \sigma^2(G(x)) + \sigma^2(G(y))$ (although it is in theory possible that the variance increases). In the definition of I in (7.4) $s_n^2(x) + s_n^2(y)$ should be replaced by

$$\frac{1}{n-1}\sum_{i=1}^{n}\Big(g_i(x) - g_i(y) - \big(\hat{g}_n(x) - \hat{g}_n(y)\big)\Big)^2.$$

This way I becomes smaller: with less effort the same precision can be obtained, or the same simulation budget can be used to get more reliable results.

Example 7.6.4 In the balance sheet example CRN can be used by taking the same input value samples such as interest rate across the different solutions. For the emergency department the patient arrival moments, treatment steps and durations can be taken the same.

Local search If S is larger or even countable then we cannot start by simulating all $x \in S$ a number of times. In this situation there is often some structure that we can exploit. Just as in the case of deterministic local search we define a neighborhood $N(x)$ for every $x \in S$. During each iteration we choose randomly a point in the neighborhood of the current point and simulate both once. Then we move to the best of the two and we iterate again. In more detail the algorithm is as follows.

Algorithm for local search simulation optimization
0. Set $n(x) = 0$ for all $x \in S$, choose x^*
1. Select randomly $x' \in N(x^*)$, obtain simulations $g_{n(x^*)}(x^*)$ and $g_{n(x')}(x')$, calculate $\hat{g}_{n(x^*)+1}(x^*)$ and $\hat{g}_{n(x')+1}(x')$, increase $n(x^*)$ and $n(x')$
2. If $\hat{g}_{n(x^*)}(x^*) > \hat{g}_{n(x')}(x')$ then $x' = x^*$
3. Repeat from 1 until simulation budget is exhausted (or some other stopping rule)
4. $x^* = \arg\max_x\{n(x)\}$

This is one of the simplest algorithms that exists, many more elaborate algorithms are described in the literature.

Convex constraint sets For convex S we choose an algorithm which parallels the one for deterministic convex optimization on page 111. We take

$$x_{k+1} = x_k - \gamma_{k+1}\hat{\nabla} g(x_k)$$

with $\hat{\nabla} g(x)$ an approximation of the gradient in x and γ_k an appropriate series such as $\gamma_k = 1/k$. This algorithm is called *stochastic approximation*. Note that it can move away from the local minimum, because $\hat{\nabla} g$ is only an approximation of the gradient. The choice of γ_k guarantees converge to a local minimum. In practice it is better to start with a constant step size: experiments show that convergenge is generally much faster.

The challenge of this approach is to approximate ∇g. This is done as follows:

$$\left(\hat{\nabla} g(x_k)\right)_j = \frac{g_k(x + he_j) - g_k(x)}{h},$$

for some well-chosen h. Thus the gradient of the objective is obtained as a difference of two realizations. An alternative approach is

$$\left(\hat{\nabla} g(x_k)\right)_j = \frac{g_k(x + he_j) - g_k(x - he_j)}{2h}.$$

Note that this requires twice as much simulation. To avoid having to do n or $2n$ simulations at every iteration we can also use the method of *simultaneous perturbations*. In that case a vector Δ is constructed with $\mathbb{P}(\Delta_j = -1) = \mathbb{P}(\Delta_j = 1) = 1/2$. Now the approximation is as follows:

$$\left(\hat{\nabla} g(x_k)\right)_j = \frac{g_k(x + h\Delta) - g_k(x - h\Delta)}{2h\Delta_j}.$$

Stochastic constraints Up to now assumed that S is a set for which we can decide if $x \in S$ with certainty, without simulation. The disadvantage in the simopt-setting is the fact that we are never 100% sure if a solution is feasible. A common approach is to put the constraints that are simulated in the objective using a penalty function, just as we did in Section 7.3.

Example 7.6.5 We continue with Example 7.5.2, but now in a multi-dimensional setting. Suppose we have a multi-skill call center, with customers arriving in multiple queues and employees having different types of calls they can handle. Some form of routing is in place where

calls are assigned to employees with the right skills. This complicated queueing network cannot be analyzed analytically, simulation is the only method. This simopt is the only way to find the optimal combination of shifts and skills. Note that the constraints are simulated, not the objective. For more details, see Chapter 17.

It should be noted that stochastic constraints are often *soft constraints*. In many practical optimization constraints, the set of constraints can be split in two: soft constraints and hard constraints. Hard constraints really have to be satisfied, soft constraints allow for some margin. Stochastic constraints are often soft constraints. In the call center example, the shift structure (represented by the a_{tk}) is a hard constraint, the service level is a soft constraint: it is acceptable if β is not met by a small margin.

7.7 Dynamic optimization

A class of problems that is hard to fit into the framework of Equation (7.1) are *dynamic optimization* problems. Such a problem consists of a model which state evolves over time and in which decisions have to be taken, depending on the state, which influence costs and the evolution of the model. Because of this time dependence we have to evaluate the future evolution before we can solve the current decision problem. This leads to a *backward recursion* approach to solve the problem. Let us state this formally.

A stochastic dynamic optimization problem or *Markov decision chain* is a Markov chain (see Chapter 4) with the additional features of costs and actions. We assume there is a finite time horizon T. The evolution of the system is as follows: if at $0 < t < T$ the state is $x \in \mathcal{X}$, and action $a \in \mathcal{A}$ is chosen, then costs $c_t(x,a)$ are incurred, and the state at $t+1$ is y with probability $p_t(x,a,y)$. The problem to solve is: which actions to choose for every state-time combination such that the total expected costs are minimized?

The crucial step in solving these problems is to define $V_t(x)$ as the expected total minimal costs from t on when starting in x. $V_t(x)$ for $t < T$ can be computed using the following recursion:

$$V_t(x) = \min_{a \in \mathcal{A}} \Big\{ c_t(x,a) + \sum_{y \in \mathcal{X}} p_t(x,a,y) V_{t+1}(y) \Big\}.$$

For the last interval we simply take $V_T(x) = \min_a c_T(x,a)$. Now we can recursively compute the V_t, starting with $t = T$, and then going backward until we reach $t = 1$.

Example 7.7.1 (Revenue management) A company has T time units (say days) to sell a capacity of C (say seats on a flight) for as much money as possible. We assume that every

time unit only 1 booking can occur. We have two price levels p_1 and p_2, $p_1 > p_2$. If we offer the low price then our probability of selling is q_2, if we offer the high price then we sell with probability $q_1 < q_2$. At every point in time, also depending on the current free capacity $x \in \mathcal{X} = \{0, \ldots, C\}$, we should decide which price to offer. This leads to the following backward recursion formula:

$$V_t(x) = \max\Big\{p_1q_1 + q_1V_{t+1}(x-1) + (1-q_1)V_{t+1}(x), p_2q_2 + q_2V_{t+1}(x-1) + (1-q_2)V_{t+1}(x)\Big\}$$

for $t < T$ and $x > 0$, $V_T(x) = \max\{p_1q_1, p_2q_2\}$ for $x > 0$ and $V_t(0) = 0$ for all t.

This type of model is further discussed in Chapter 18.

Example 7.7.2 (Shortest path) Let $\mathcal{X}$ be the nodes in some road network, and $\mathcal{A} = \mathcal{X}$ where action y means going to y with the direct link, if one exists. The goal is to reach a certain state x^* using the shortest possible path. We take $V_T(x^*) = 0$ and $V_T(x) = \infty$ for $x \neq x^*$ and, for $t < T$,

$$V_t(x) = \min_y \Big\{l(x,y) + V_{t+1}(y)\Big\},$$

with $l(x,y) = 0$ if $x = y$, the length of the connection if there is a direct connection between x and y, and ∞ if y cannot be reached from x in one step. Then $V_t(x)$ can be interpreted as the minimal distance from x to x^* in less than $T - t$ steps. When we take $T = |\mathcal{X}| - 1$ then we are certain to find the shortest path without restrictions on the number of nodes visited: it makes no sense to visit the same node more than once (assuming that $l(x,y) \geq 0$).

This algorithm has a running time in the order of n^3, with n the number of cities: n times we should consider n cities for which we look at n possibilities every time.

Example 7.7.3 (Distribution of a Markov chain) We interested in computing in an alternative way the distribution of a Markov chain at T for some given initial state at time 0, which is π_T as defined in Section 4.1. Consider a Markov decision chain without actions, $c_t(x) = 0$ for all t and x except for $V_T(x^*) = 1$. Then the recursion

$$V_t(x) = \sum_y p(x,y)V_{t+1}(y) \tag{7.5}$$

can be used to determine $V_0(x) = \pi_T(x^*)$ for initial state x. Thus the recursion (4.1) can be used to compute the distribution in all states for a given initial state, (7.5) can be used to calculate to probability of reaching a state for every initial state.

Learning We can add another level of complexity to our dynamic optimization problems, by considering problems where the transition mechanism has to be learnt on the fly. There are different solution approaches to these problems, with can be

classified into more statistical and Bayesian methods and methods finding their origin in artificial intelligence, called *reinforcement learning*. The main trade-off in these problems is between *exploration* and *exploitation*, that is, should we try to minimize costs given our current beliefs in the state transitions or should we invest in having better information? A discussion of these methods goes beyond the scope of this text.

7.8 Further reading

All introductory text books on operations research (such as Winston [164]) contain chapters on linear and convex optimization and local search. An example of a more advanced text on linear and convex optimization is Luenberger & Ye [107].

Our discussion of simulation optimization is largely based on Nelson [119], Fu [56] and Pasupathy & Ghosh [123]. [119] is a text book on simulation including a chapter on simopt, [56] is an accessible introduction to the subject, [123] is a more research-level tutorial.

The best-known text book on Markov decision chains is Puterman [127]. A recent text, with an emphasis on applications, is Bhulai & Koole [22].

7.9 Exercises

Exercise 7.1 Implement Example 7.3.2 in Excel and try it for different parameter values. Execute it also for $C_1 = C_2$ and prove that the answer is correct.

Exercise 7.2 Reproduce all the steps of Example 7.4.1 by making a 2-dimensional diagram and by using the Excel solver.

Exercise 7.3 Formulate the TSP as a mixed-integer linopt problem.

Exercise 7.4 Give an example of an instance of the algorithm on page 118 in which $\arg\min_x \hat{g}_{n(x)}(x) \neq \arg\max_x n(x)$ when the simulation budget is exhausted.

Exercise 7.5 Suppose you have to drive from Amsterdam to Berlin. For each stretch of road the distribution of the time to travel is given.
a. Give an algorithm to find the route that minimizes the expected travel time (hint: adapt the dynamic programming algorithm).
b. Give an algorithm that maximizes the probability that the travel time is less than 5 hours.

Exercise 7.6 a. Calculate the optimal policy of Example 7.7.1 for $C = 20$, $p_1 = 300$, $p_2 = 100$, $q_1 = 0.25$, $q_2 = 0.5$, and $T = 60$.
b. Calculate the expected number of empty seats under the optimal policy.

Exercise 7.7 Consider an arbitrary Markov chain with 3 states, and use both the method of (4.1) and of (7.5) to calculate the distribution after 10 steps, for a certain initial state.

Part II

Modeling

Chapter 8

The Modeling Process

Modeling is a word used in many contexts and sciences. In this monograph we deal with *mathematical* modeling, the process of solving real-world problems using mathematical techniques. This process involves much more than just mathematics. In this chapter we try to lay a theoretical foundation of the whole problem-solving process.

Before going into the details, let us pose ourselves the question what we expect to learn from this chapter. Indeed, many people state that, while solving mathematical models is a science, modeling is an art! With that they mean to say there is no theoretical foundation to modeling, as there is to model solving, but that good modeling needs a combination of talent and experience. However, in the current chapter we try to convince you that there are some general rules to learn and pitfalls to be avoided. It remains true however that practical experience is indispensable for good modeling.

8.1 Introduction and definitions

Managers solve problems. These problems can be of many different natures, with human aspects, organizational aspects, etc. Some of the problems have quantitative aspects that go beyond simple calculations. This book is about this type of problems. Rarely a problem with quantitative aspects can be solved taking all its quantitative and non-quantitative aspects into account. Therefore a *model* of reality needs to be constructed first.

A model is a (simplified) description of a real-world process or phenomenon. This is often a written description, but it can also be a physical construction, of for example a building. The object which is to be modeled need not exist (yet). To avoid confusion

we use the neutral term *system* to designate the object of our study. When we want to stress the dynamic nature of the system we also use the term *process*. We see a process as a related group of activities with a common goal.

Example 8.1.1 A machine or a group of machines is an example of a (production) system. If the main interest is not the group of machines but the production or transformation of goods then we speak of a production process.

With the *environment* of a system we mean all other systems and processes with which the system can interact. The formal definition of a model that we use is as follows.

Definition 8.1.2 *A* model *is a description of a part of a system or process and its interaction with its environment that allows an analysis of certain aspects of that system or process.*

The extent to which system details are taken into account and the choice of the details depend on the objectives of the modeling phase. Differences in depth of the analysis will be discussed later. Here we stress that the model depends also on its (possible) use in the organization. This is reflected in the definition by the fact that a model is such that it allows an analysis of aspects that are of interest to the model builders.

Example 8.1.3 A model of a data communication line used for throughput estimations could abstract from cell losses. This model would evidently be useless for estimating losses.

The term modeling in a narrow sense is just the process of constructing the model. However, usually we designate by modeling the whole method of solving (business) problems using models.

Definition 8.1.4 Modeling *is a methodology for problem solving in which the use of models plays a crucial role.*

We realize that the word 'problem' has a somewhat negative connotation, but we use it because we think that it describes the concept best. Alternatives are 'challenge' and 'question'.

A *mathematical model* is a model in which the relations within the model are given in mathematical terms. These notes deal with mathematical modeling, but some of the ideas are also applicable to models in a more general context.

The next section describes the modeling process by breaking it up into several steps.

8.2 Steps in the modeling process

In a typical modeling project we can distinguish several phases. They are shown in the figure below. The phases correspond to arrows, the starting situation and the products of the phases are represented by ellipses.

Starting from a system and a research question or problem we begin with the *model construction phase*. The result of this phase is a model. This step is qualitative in the sense that relations between quantities are given. An important aspect of the model construction phase are decisions concerning which details to model and which to leave out.

The resulting model is solved using one of the mathematical techniques to be discussed later in Part I. However, finding the right solution technique and executing it is not all that happens in the *model solving phase*. To execute the model data has to be available. The necessary data collection and analysis is an important and time consuming part of this phase.

The solution to the model solving phase does not directly give us the solution to the system problem. Indeed, as the model is a simplified representation of the system under study, translating the model solution back to a system solution is not always easy, and sometimes not even possible. Making this possible is one of the main concerns when modeling (the other ones being the availability of relevant data and the possibility to solve the model). For a well chosen model this translation is not that difficult, most of the time of the reporting phase goes into convincing the problem owners of the correctness and feasibility of the proposed solution.

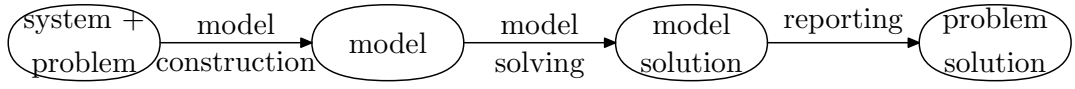


Figure 8.1: The modeling process.

We assume that the objective is part of the model, therefore the objective is not noted apart. This is for two reasons: it is common in the literature on solution techniques that the objective is part of the model, and it shows well that, while modeling, the objective should be taken into account.

From Figure 8.1 we might well get the impression that modeling is a linear process, i.e., it finishes after having dealt with the various steps consecutively. This is a wrong impression. In any stage of the modeling process there is feedback possible, the most important one is from the system solution back to the modeling phase. This is for example the case if the system solution is not implementable due to reasons that were abstracted from in the modeling phase. Another example of feedback is from

the model solving phase to the modeling phase, if it is discovered that the model is too hard to solve and thus needs simplification.

The implementation can start as soon as we have a satisfactory outcome of the modeling phase. Often it is done first on a small scale. This allows the problem owners to gain confidence in the model outcomes, and effects that were not modeled can be studied on a system level. Feedback to the different phases of the modeling process occurs again.

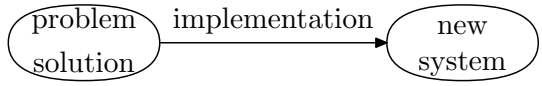


Figure 8.2: The implementation.

Note that modeling and implementation are usually done by different people. Modeling is often done by consultants specialized in modeling, the implementation is done by those who manage the systems and processes involved.

Example 8.2.1 In a large retail organization management is considering to change the distribution policy, that comes down to reallocating certain activities in the supply chain. The model contains all activities in the supply chain from the distribution center to the store. Due to the complexity of the model it was decided to use simulation as solution technique. Data collection was a major issue: certain handling times had to be measured on the spot! When the final results of the modeling process were presented to a large group of managers (only after several iterations of improvements based on feedback from a smaller group of logistics managers), they were considered to be counter-intuitive. It was decided that the proposed policy was to be tested first on a small but representative groups of products. Based on this a decision concerning all products was to be taken.

Remark 8.2.2 Improving business processes, through modeling or otherwise, involves more than solving a single problem once. Nowadays businesses improve their processes continuously. Our modeling process can be seen as steps of iterative problem-solving programs such as the *Deming cycle* or *Six Sigma*'s *DMAIC*. The Deming cycle ("Plan-Do-Check-Act", PDCA) is an iterative problem-solving method developed by W.E. Deming who was a statistician working on quality control. Six Sigma is a statistical improvement method originally developed at Motorola. DMAIC stands for Define-Measure-Analyse-Improve-Control, a variant of PDCA.

8.3 Business problems

So far we gave definitions of a model and of modeling, and we discussed the various steps of a modeling project. Here we describe some general aspects of business prob-

lems and we classify the different types of problems that we encounter. We will see for which type of problems modeling can help us find a solution. In the next section we take a closer look at the structure of a mathematical model.

To formulate problems such that they are amenable for analysis and that different solutions can be compared we need to make the process under consideration *measurable*, i.e., we need to find ways to *quantify* the aspects of the process that are important. These numbers are called *(key) performance indicators* (KPI's), and are often easily translated into a model. One might even say that KPI's themselves are models of the company's performance. KPI's should be formulated with care and their strict use can sometimes lead to unwanted situations.

Example 8.3.1 A call center wants to give friendly service with a short waiting time. Both can be quantified: the friendliness is measured by asking customers to rate it on a scale of 1 to 5, the KPI for waiting time is the percentage of callers that wait less than 20 seconds. Strictly following this latter objective means that callers who have waited more than 20 seconds should be ignored and left waiting "forever" (of course, they abandon after some time). This is in contrast with the general objective of a short waiting time. However, call centers are sometimes tempted to use this policy, certainly if the payment they receive is a function of the waiting time KPI.

Example 8.3.2 Similar issues can occur in health care. A striking example, where adhering to service level came in the place of quality health care, has been reported in a British hospital: "In an attempt to meet the target four-hour Accident & Emergency waiting time, patients were sometimes "dumped" in a ward without nursing care" [1].

We give a number of ways to classify business problems.

Programmed and unprogrammed problems The most important classification is that between programmed and unprogrammed problems (also called structured and unstructured). Problems are called programmed if "they are repetitive and routine, to the extent that a definite procedure has been worked out for handling them" (Simon [143], p. 46). Note that this procedure can be some highly suboptimal heuristical rule which has nothing to do with modeling. Problems which are not programmed are called unprogrammed. Of course there is a whole continuum between programmed and unprogrammed problems. Modeling can play a role in the solution of both programmed and unprogrammed problems, but its impact on the final solution is often bigger for programmed problems. There are several reasons for that, among which are the fact that for repetitive problems data is often available, and that programmed problems are usually easier to model.

Strategic, tactical and operational decisions A second way to classify problems is by its level of management decision. According to Anthony [8], there are three levels of decisions:

- strategic decisions, dealing with the determination of long-range goals and the means to achieve these;
- managerial or tactical decisions, concerning the realization of the long-term goals and the management of the resources;
- operational decisions, dealing with the short-time planning.

Of course, top management is concerned with strategic decisions, and low level management or the production employees themselves are concerned with the short-term planning. We see that the lower the decision level, the shorter the horizon over which the decision takes effect, and thus low level decisions are more often repetitive. Therefore problems at an operational level lend themselves better to a modeling approach, but sometimes strategic or tactical decisions can also be supported by mathematical models. "On the average, the decisions that the president and vice-president face are less programmed than those faced by the factory department head or the factory manager" (Simon [143], p. 31).

Internal and external coordination Another way to classify problems is by the fact whether they have only internal or also external aspects. Thus problems that are concerned with *external coordination* are related to the way the firm should react to changes from the outside. Examples are the behavior of competitors (new products or prices), changes to the workforce due to union and government decisions, etc. *Internal* coordination is concerned with problems that exist totally within the interior of the firm. External coordination problems are very often at the strategic level, internal problems at the tactical/operational level. Sometimes this is even used as a definition of strategic and tactical/operational.

Classification by objective Problems can also be classified by the goal which is to be achieved. This objective can range from an educated guess to the solution of a certain problem to a computer system that takes automatically and independently its decisions. Even if the goal is to find an optimal decision, it can be better to build a tool able to answer "what if" questions by which the managers involved can find the solution by experimentation. The goal should be taken into account when modeling.

It is clear that a modeling approach is more successful in situations where a detailed solution is wanted.

Example 8.3.3 To fulfill a bottleneck analysis in a production system a simple spreadsheet model may suffice. To produce an optimal production schedule one often has to solve a complicated model.

Design and control problems A final distinction is between *design* and *control* problems. Design problems are those related to setting up a system or process, control problems are those dealing with operating systems or processes. Design problems are often at the strategic or tactical level, control problems are often operational.

Example 8.3.4 In a distribution setting, choosing the location of warehouses is a design problem at the strategic level. Selecting the daily routes from the warehouse to the customers is a control problem at the operational level. Setting up the information and other systems to be able to select these routes in an efficient way involves decisions at the tactical level.

Planning and scheduling What are called control problems above can be split up in *planning* and *scheduling* problems. Planning is concerned with the long-term control issues, often at the tactical level. Scheduling deals with the operational short-term control. Sometimes the word control is also used to indicate the activity consisting of checking whether the foreseen plans or schedules are met and taking the appropriate actions when necessary.

Example 8.3.5 In Part III we make the distinction between production planning and production control. Production planning deals with building production plans taking into account various constraints. Production scheduling deals with the actions that allow the production system to keep to the production plans.

8.4 General model structure

In this section we describe the general form of models. While discussing modeling phases we already stated that the system problem, translated to the model level, is part of the model. Thus the model needs to be *solved*. Models are often identified by their solution techniques: e.g., models with a linear objective and linear constraints are often called linear programming models.

The solution to the model is the output of the model solving phase. The input is the model itself. This input can be split in two: the form of the mathematical

description and the parameter values. It is important to make this distinction: the form of the input determines the solution techniques, and if the problem is repetitive, then only the parameter values change. Having in mind problems with this repetitive nature, it is logical to think of a model as only comprising the mathematical rules specifying the relations, and to see the actual parameter values as input. The same distinction shows the two driving forces behind modeling as a way to solve business problems. The first is the availability of enough computational power on the desktop to solve complicated models. Powerful standard software packages for models such as linear programming make this computational power easily accessible. The second driving force is the availability of enough relevant data concerning the problem at hand. Indeed, companies nowadays gather enormous amounts of data, all having the potential to be turned into useful information for improving business processes. Seen as such modeling really turns data into information! Data collection is the subject of the next section.

The output can have different forms. Sometimes it is just a number, which could give the answer whether targets set by the management will be met or not. It can also be a complete policy specifying what to do in all possible situations. The last type of output is often on-line, meaning that (using some computer system) at any moment in time, given the current situation, the optimal control action can be determined.

Example 8.4.1 A model of a service center can well have as output the expected waiting time before service for arriving customers. A similar model where the number of servers can be controlled could have as output rules such as "if the number of waiting customers is higher than n, call for assistance". In call centers this is often implemented on-line, for example using a text display.

Every system or process has one or more objectives or goals and utilizes resources to achieve these goals. Thus modeling (and business problems in general) is always a balance between service (the extent to which goals are met) and costs for the use of resources, or, stated differently, between product and production costs. We quantify the quality of service with the *service level*. Often the service level is of a statistical nature, a certain fraction of the products should satisfy a strict quality constraint. This is because often no guarantees about all products or to all customers can be given. The typical objective for a model with costs and service level as separate entities is to minimize costs under a service level constraint or vice versa.

Example 8.4.2 In a call center with a Poisson arrival process queueing can always occur, no matter how many agents there are. Service level constraints are therefore of the form: a fraction x of all calls should be answered within s seconds.

From a business perspective the use of service levels also has certain disadvantages. They have to do with the fact that no matter which service level definition is chosen, it is never exactly aligned with the business objectives. Thus, when managers are evaluated on the basis of the extend to which they reach the service level objective, actual quality of service can decrease. We give two flagrant examples.

Example 8.4.3 An outsourcing call center gets a penalty for each day that 80% of the calls is not served in 20 seconds. To reach this service level the first-come first-served order discipline is abandoned: Calls who have waited more than 20 seconds are answered with lowest priority or not at all, sometimes leading to very long waiting times and low customer satisfaction. A (partial) solution is to impose FCFS or to use a different service level definition. See Chapter 17 for more details.

Example 8.4.4 In British hospitals there used to be a rule which said that 100% of all patients in the emergency departments should be discharged within 4 hours. Not adhering to this rule had severe financial consequences. This led in certain situations to a deterioration of patient care. For example, ambulances were asked to wait outside of hospitals until enough capacity to handle the patients timely was available. Nowadays the 4-hour rule is still used as a peformance indicator but the consequences of not abiding to it are less severe. This allows again care workers to focus on delivering good health care instead of focusing only on the service level.

OR professionals tend to integrate both service level and resource utilization in a single objective. The reason for this is that it becomes possible to optimize the system to this objective. However, from a practical point of view it is often better not to integrate service and costs, for several reasons. In the first place it is difficult to choose good weighing factors, necessary for the construction of the single objective. These weighing factors represent how much management is willing to spend to increase the service level with one unit. Of course this is very hard to quantify, if at all possible. In the second place, even if it would be possible to determine these weighing factors, it is often not desirable. This is because often we want to keep the service level explicit, for example for marketing reasons. Solutions for which one of the objectives cannot be improved without decreasing the other are said to lie on the *efficiency frontier*.

Example 8.4.5 The general service manager of a large company selling copy machines described the goal of the maintenance activities as giving the customer "a good service for a reasonable price". Although the value of service is hard to quantify, the manager had a good idea of what he found acceptable for which price. A queueing model helped making the trade-off between service level and labor costs by estimating the time between a maintenance request call and the arrival of a technician, as a function of the number of service personnel.

Example 8.4.6 The standard linear programming example is the determination of the optimal production mix. Suppose a plant can produce M products, each product of type m that is produced can be sold for a price p_m. There are N resources. Of resource n there is c_n available, and product m demands a_{mn} of resource n. The product mix that maximizes profit can be obtained by solving

$$\max \left\{ \sum_{m=1}^{M} p_m x_m \;\middle|\; \begin{array}{ll} \sum_{m=1}^{M} a_{mn} x_m \leq c_n, & n = 1, \ldots, N \\ x_m \geq 0, & m = 1, \ldots, M \end{array} \right\},$$

where x_m is the production level for product m.

Here the service level is translated into profit, and costs for resource utilization are translated into upper bounds for the amount of available resources.

Example 8.4.7 A bank uses call centers as its main means to communicate with its customers. For marketing reasons the waiting times of calls must be very short. Thus labor costs are minimized under the condition that the service level is higher than a certain level.

8.5 Data collection and analysis

Gathering and preparing data is an important activity in a modeling project. The mathematical aspects are partly outside the scope of this book, but can be found in many text books on data analysis and statistics. Here we discuss some practical aspects.

First we have to realize the importance of obtaining correct data. "Garbage in—garbage out" is not without reason a well known phrase. Getting good input for your model is more than doing statistics well: above all it is acquiring, measuring and estimating data correctly and then using it correctly.

Due to advances in information technology companies register more and more business transactions. This simplifies the task of gathering data enormously. Still, this does not mean that this data can be used right away. Often data is aggregated or gathered in another way that is not appropriate for immediate use in a model. Sometimes another model is needed to generate data that can be used as input for the original model.

Example 8.5.1 All sales of a retail organization were automatically stored in an information system. However, at the end of each month, these sales were aggregated to monthly sales numbers. To perform a simulation study on the impact of a new inventory policy detailed sales over a year were needed as input. These numbers were generated based on the monthly sales and the detailed sales over the last month. This model could be verified when new data became available after a month.

Example 8.5.2 In a call center log files with many statistics are available. However, often they are aggregated over 15 minute periods. To analyse the performance of a call center one often needs the call length distribution. This distribution is hard to obtain as only 15 minute averages are available.

For other issues on data analysis we refer to the literature on this subject.

In what we described so far we assumed that data is already available. This is not always the case; sometimes the necessary input has to be estimated or measured, if at all possible. It is clear that this can be an extremely time-consuming activity.

Another distinction is between internal and external data. Internal data is relative to the firm, external data needs to be acquired externally.

Example 8.5.3 To assess the profitability of an investment one often needs to estimate the interest rate for future years. This is a typical example of external data.

The enormous amounts of data that are currently being gathered by businesses have stimulated people with IT backgrounds to think what can be done with this data. This has stimulated the development of new data analysis and optimization techniques, partly in parallel to the mathematical fields of statistics, stochastic modelling and combinatorial optimization. The notions used in this context are analytics, data mining, computational intelligence and business intelligence. According to Davenport & Harris [50], optimization is the final and most sophisticated part of business intelligence. It is interesting to note that, although OR/MS is hardly mentioned, many of the examples that are described in [50] are part of "traditional" OR/MS.

8.6 Verification and validation

It goes without saying that it is of crucial importance that solution techniques are implemented correctly, and that the system behavior is represented well by the model. Checking these carefully convinces not only the modeler of the correctness of his or her model and data, it also can play an important role in convincing the problem owners of the chosen approach.

The term verification and validation are often used in a simulation context, but they can and should be used for any type of model. Verification means verifying that the implementation of the model is correct; validation means that it is checked that the model outcomes correspond with those of the system up to a certain extent. As the system need not exist already this is not always possible. It is always possible however to check parts of the model, or to predict outcomes in some other way.

Validation not always means comparing a model with a system; it can also be the case that a model is validated with another model, one which has more detail for example.

Example 8.6.1 In call centers data is not always reliable. Validation and if necessary parameter tuning (also called *calibration*) can be done on the basis of a simulation model. The resulting parameter values can then be used in some optimization model, whose outcomes can be checked with simulations.

Validation of large-scale stochastic models is often difficult. This is even more so if human behavior is involved. This is a serious objection against the modeling of processes. The possibility of a failure to validate the model should be taken into account before starting the modeling process.

8.7 Suboptimization

Modeling is always a compromise between the scope of the model and the complexity. If the model is too complex to solve satisfactorily then decreasing the model scope is an option. This has the risk that the influence on system parts that are not modeled is ignored. This influence can be important enough to change the proposed solution to the problem.

Example 8.7.1 In logistics it was common business for each participant of a supply chain to optimize its own processes. However, by seeing the supply chain as a whole, considerable improvements can be obtained. Therefore supply chain planning is currently one of the hot issues in operations management.

Another possibility for checking the influence of a small scope optimization procedure on larger scope issues is using simulation as a large scope model. Thus the small scope optimization is *validated* by a large scope simulation. Often however the influence of suboptimization on other systems is hard to quantify, and modeling cannot help us to assess the consequences.

Example 8.7.2 In a production line it was decided that large production batches were more efficient. The upstream systems however were confronted with increasing demand sizes and were forced to increase stock levels. Due to this total costs went up.

8.8 To model or not?

Modeling is a "white-box" approach: it requires that the behavior of the system under study is completely specified. Modeling is therefore a time-consuming activity, that demands much of the knowledge and skills of the modeler(s). An important question to be asked before starting a modeling process is therefore: can we solve the problem by an alternative approach that requires less time and that is (at least) equally reliable in giving the right answers? As modelers are often eager to apply their knowledge and skills, this question is not asked often enough.

A good candidate for a "quick and dirty" solution method that is often overlooked by OR/MS professionals is a statistical black-box approach. Here the inputs of the system are directly related to the output, and on the basis of this conclusions can be drawn. Of course, this works only if the system already exists and if data on the behavior of the whole system is available. This approach can be compared to simulation, where the output is also analysed in a statistical way. The advantage of simulation is of course that it allows to study non-existing systems or not yet implemented scenarios, the disadvantage is that modeling is necessary. The statistical method has gained popularity due to the increased availability of data and the development of new machine learning methods.

Example 8.8.1 A hospital department was trying to find out what the main causes where of waiting times of patients. During a month they collected data on waiting times and activities of medical personnel. At the time both a modeling study was started and a statistical analysis was done on the outcomes. The statistical analysis readily gave useful results. The modeling study gave unreliable answers that could not be improved on due to a lack of reliable data concerning the activities of the doctors.

An intermediate approach is to analyze a system first through a statistical approach and then model only the part(s) where most of the improvement can be obtained. This avoids going through a long modeling study, and often simple analytic models suffice. The main advantage is that the probability of finishing the project is much higher, in much less time: the invested time is spent much better, the average return on invested time is higher. An additional advantage is that the proposed solution is focused on the issues where most can be gained.

Example 8.8.2 A production system can be modeled in its totality, giving superior results, but only if complete data of the entire process is available. This is not often the case. Instead, one could focus on the production step that is the bottleneck and/or on the one that is responsible for most of the delay. Modeling only this node is much less work. In a subsequent project another node, perhaps the new bottleneck, can be analyzed.

Philosophically speaking, statistics can be seen as an *inductive* method: on the basis of a number of observations general conclusions are drawn. Modeling, on the other hand, is a *deductive* approach: on the basis of known behavior of components conclusions for specific models are drawn. Note however, that the behavior of the components is often obtained through statistics. Therefore, statistics also plays an important role in almost any modeling project.

Remark 8.8.3 In this section we discussed two scientific system improvement methods: modeling and statistics. In practice also methods are used which are based on a mixture of science, experience and common sense, notably *Lean Manufacturing*, *Six Sigma* and the *Theory of Constraints* (ToC). Lean finds its origins in the Toyota manufacturing plants of the 1970s, Six Sigma was conceptualized at Motorola using ideas from statistical process control. Also ToC has scientific roots, its founder Goldratt was a physicist applying scientific ideas to improvement business processes. They are discussed in more detail in Chapter 12 on manufacturing, although they are also applied in for example health care.

8.9 Model builders and problem owners

Modeling is an activity that can be applied in many fields. Similar mathematical models are suitable for production as well as administrative processes, and a solution technique such as linear programming has so many applications that it is a standard option in any major spreadsheet. Thus modeling can be seen as a *generic* discipline. Applying it successfully therefore demands people specialized in modeling.

The modelers or model builders usually have no responsibility for the systems they model. This responsibility lies with the managers that are concerned with the daily operations of the systems, the *problem owners*. The modelers are internal or external consultants. The differences between internal and external consultants are disappearing, as most internal modeling groups become responsible for their own results, and because they have to compete with external consultants.

A third group of people that might be involved in a modeling project are IT specialists. Here as well we see different constructions: programmers can come from different departments in the problem owner's or model builder's organizations. Sometimes the implementation is done by an independent IT firm, which can be the main contractor or the subcontractor. That this can both occur is understandable if we know that on one side there are companies which are specialized in modeling that do not implement, and that on the other hand modeling is sometimes a small part of a big IT project, in which the IT company involved as main contractor is not specialized.

It is clear that communication between the project partners is of crucial importance. This demands from the modeler, besides good communication skills, insight in the problem domain and the implementation aspects.

A crucial part in the relation between modeler and problem owner is convincing the last one that the solution proposed by the model is correct. Verification and validation only allow to check the modeler's work, it does not allow the problem owner to participate in the modeling. Making managers and employees participate in the modeling can greatly increase the acceptance of the final solutions.

Example 8.9.1 A logistics company restructures its European distribution network. To convince local managers of the cost-effectiveness of the proposed solution a computer system is built in which the effects of changes can easily be calculated, during a meeting. This way they can convince themselves of the correctness of the solution.

Being responsable for the implementation of the solution, the problem owner has to deal with the organizational consequences of it. Often, a simple rule with little organizational consequences is preferred over a less simple slightly better ("optimal") rule. This should be taken into account during the modeling process.

Example 8.9.2 A hospital wanted to maximize at the same time throughput in one of its care chains (from the operation room to the intensive care to the normal care) and the occupancy of the beds. A complex model-based information system was proposed. However, a simple rule (always keeping a few beds empty at the normal care unit for patients dismissed from the IC) performed almost as good and avoided communication overhead between the different departments.

When modeling projects fail this is usually not due to a lack of modeling skills of the modeler, but moreoften due to implementation problems. For this reason good project management is of crucial importance to the success of modeling projects. Several business improvement frameworks strictly describe how the relations between the different project members should be and which hurdles are to be taken. Examples are the statistical quality program *Six Sigma* and Goldratt's *Theory of Constraints* which focuses on finding bottlenecks in all kinds of processes.

The strict division between modeling and management skills in the firm makes that modeling is used mainly for large-scale projects. However, the quality of everyday decision making can be improved as well by modeling insights and techniques. Therefore managers in relevant areas should have modeling knowledge. On the other hand, one can pose the question why modelers can only be found in specialized firms and dedicated departments. The main reason is that managers do not believe in the

usefulness of modeling for problems other than low-level operational planning tasks. This is partly due to the fact that managers rarely have a background in modeling. However, the modeler is to be blamed as well: he or she does not always speak the same language as the manager, and is often more interested in sophisticated mathematical methods than in solving real-world problems. The expulsion of the modelers to specialized OR departments (sometimes even privatized), where they are financially responsible for their results, guarantees the usefulness of their work to the company.

This situation does not stimulate the every-day use of modeling in the firm: the modeler is only called for if the management has a clear-cut task to be done which involves, in the management's opinion, modeling. This task is often of an operational nature; without modeling knowledge or experience a manager will not easily rely on modeling for strategic or tactical decisions.

8.10 Skills and attitudes of model builders

The business environment in which a model builder is working forces him or her to have certain skills and attitudes in order to be successful. Certainly, the right scientific knowledge about models and their solution techniques is the starting point. Note that this knowledge need not only have some depth but also a certain broadness: the modeler should be aware of the different techniques and model types that are at his or her disposition. Next to mathematical knowledge the model builder should have relevant knowledge of information technology, as nowadays models are rarely solved by hand.

Apart from scientific knowledge the modeler should be aware of the specific issues that play a role in the area he or she is working in and of the generic way to solve certain problems in this specific area. Next to a quicker understanding of problems it gives confidence to the problem owners in the skills of the model builders. It also makes the model builder less dependent of the model owner, in terms of requiring information about the problem. The mathematician John Dennis put it this way: "Being an applied mathematician and working with people from other fields you have to be an anthropologist. You have to learn their languages." If the modeler is well aware of the situation, the roles can even be reversed: then it is the modeler who indicates problems and directly proposes solutions.

Some financial knowledge is also useful, certainly when it comes to issues at the tactical and strategic level, such as investment decisions. To understand and anticipate on the decision process it is necessary to speak the manager's financial language:

to understand what ROI means, where a computer tool shows up on the balance sheet, etc.

Communication is a key issue in the modeling process, and therefore the modeler should have the right communication skills, both orally and written. Most of the time the modeler is member of a specialized staff department or of an external firm bringing in specialized knowledge. Thus communication is needed in both directions, to understand the problem owner's problem, and to communicate the proposed solution. This situation as external specialist demands a special attitude from the modeler. He or she should give the problem and the problem owners a central place. This shift from technically oriented to problem oriented is sometimes hard to make for mathematically educated consultants. It also requires that the analysis should not be more complicated than necessary to solve the problem.

The idea of giving a central place to the problem owner, the customer, comes back when reporting. Of prime importance when writing a report, preparing a presentation, or even designing an interface, is to put yourself in the customer's place. You should ask yourself questions as to the (scientific) level of your audience, the types of problems they are interested in, and so forth. Often a report is not written for a single person, but for different people with different interests in the problem solution. Therefore a report contains often a short executive summary for those who want very quickly an idea of the content and main conclusions. The main text is written with the typical problem owner in mind, the persons with whom is mostly communicated. Usually mathematical details are avoided as much as possible. These mathematical (and also non-mathematical) details are reported on in the appendices. These sections are meant for employees directly working with the modeled systems (typically line management), and they give the modeler the possibility to clarify on technical modeling issues.

When you put yourself in the customers place, it becomes possible to ask yourself the customer's next question(s). This advantage of being "one step ahead" can be of crucial importance.

Finally, a model builder is asked to execute a modeling project for his or her specific technical capacities. However, he or she works within the environment of the firm, with its political issues and different interests of the stakeholders. It is important always to keep an eye on these issues, and to report on results in a way that is not confronting. At the same time it should be clear that honesty and scientific soundness are never to be tempered with.

8.11 Further reading

An elaborate text containing a lot of background information on problem solving and especially modeling is Turban [157]. Most books on Operations Management give information on the Deming cycle and Six Sigma. See, for example, Chapter 10 of Van Mieghem [114] on this subject. Seddon [139] discusses how abberations like in Example 8.3.2 can occur. Chapter 2 of Simon [143] develops a theory of managerial decision making, in which the distinction between programmed and unprogrammed problems plays a central role. The distinction between strategic, tactical and operational control comes from Anthony [8]. A nice discussion of the three levels, in the context of logistics and with emphasis on the type of decision, can be found in Hax & Candea [72]. A recent view on modeling, mostly dealing with simulation, is given in Pritsker [126].

The failure of OR to support strategic decisions is well described in a series of papers by Ackoff (see, e.g., [2]). A recent paper, still elaborating on the question why mathematical modeling is less successful than can be expected, is Meredith [113]. It blames the lack of validation for this failure. A possible way to successful modeling of strategic and tactical decisions is making the managers participate in the modeling process. An approach to this is *Participative business modeling*, developed by J.A.M. Vennix. See Akkermans [4] and the papers (e.g., [3]) which form this thesis. Most of the time simulation is used as solution method.

Warner [160] is an accessible text that helps understanding the (financial) way of thinking of upper management.

A Dutch text on planning and scheduling from a managerial point of view is Jorna et al. [83]. Any text book on management and organization can be helpful in putting mathematical modeling in the right business perspective.

Davenport & Harris [50] gives a business-oriented view of analytics and business intelligence.

Instructive insights on how OR/MS professionals work and perceive themselves are given in Willemain [162].

A useful text on verification and validation is Kleijnen [92]. Although it focuses on simulation, most of it applies also to other solution techniques. Gallivan [59] discusses validation, induction/deduction, and sensitivity analysis (see Section 9.7) in a health-care context, but his argument apply in general as well. Sargent [136] is a good and accessible starting point for verification and validation.

Galbraith [58] discusses the implications of installing overall planning tools in complex organizations. *Local* solutions require adding slack capacity (and are there-

fore mathematically speaking suboptimal). Senge [140] argues that a small change in a process can have a big impact, which he calls *leverage*.

A good starting point to get information on the process improvement frameworks Six Sigma and the Theory of Constraints is Wikipedia. Next to that we want to mention De Mast et al. [110] and Goldratt & Cox [66].

Many of the issues discussed here that are not related to models are part of *project management*. More information on project management, focused on process improvement projects in health care, can be found in Belson [21]. Note that project planning (Chapter 13) is part of project management.

OR works well if all stakeholders agree on the problem and the way to solve it. But what to do if people involved disagree on the model and, therefore, also on the best way to solve the problem? *Soft Systems Methodology* has been designed for dealing with such situations. Checkland & Poulter [35] give a nice concise introduction to this method.

8.12 Exercises

Exercise 8.1 Consider Exercise 8.3.1.
a. Consider the average waiting time as performance indicator. What would be the best order or orders in which to handle calls?
b. Formulate one or more different PI's that are closer to the objective of measuring short waiting times. Explain to which extend they are sensible to changes in average waiting time and variability in waiting time.

Exercise 8.2 A consultancy company has a tool to help companies place their warehouses throughout Europe. For each possible allocation it is capable to calculate the total annual costs. Classify the problem that can be solved with this tool using all classifications of Section 8.3.

Exercise 8.3 Relate verification and validation to the figure of the modeling process (Figure 8.1). Which step(s) have to be done again if the result of the verification is negative? Answer the same question for validation.

Exercise 8.4 Some people say that modeling is "art, not science". What do you think of this opinion?

Chapter 9

Model and System Properties

The number of mathematical models that can be found in the literature is huge. Choosing the right model and solution technique for a certain system problem is not always easy, even if one is familiar with the most common models. In Part I we discussed different classes of models. In this chapter we discuss certain generic aspects of models *and* systems, which can be useful in making modeling decisions and choosing the right model.

9.1 Generating versus evaluating

In Chapter 8 we treated all output of a model similarly, whether it consisted of just a number or a whole policy. Here we make a difference between *evaluating* and *generating* models and systems.

Generating models generate decisions. Typical examples are linear programming and dynamic programming. This decision can be a single number (or even "yes" or "no"), some vector or a range of decisions. These are all special cases of *policies*, which specify at each point in time and for each situation what to do. Therefore we say that generating models generate policies.

We have chosen the term generating models instead of, for example, optimizing models, for the following reason. When a policy is sought, we often search the optimal one (with respect to some objective). For certain classes of models this is computationally not feasible, and we have to stick to heuristics that approximate the optimal policy. Therefore the term generating is more appropriate.

Evaluating models evaluate decisions. They just give a number (or a set of numbers), the policy (if something as a policy can be identified at all) is thus part of the in-

put. Simulation is the prime example of an evaluating modeling technique (although simulation can also be used for optimization).

The distinction between generating and evaluating is even more clear at the system level. Is there already a proposed solution to the problem available that only needs to be evaluated, or do we want that the solution is generated by the modeling process?

Many models can be formulated as a generating model and a evaluating model. Often the generating version has a special case that is evaluating. This means that generating models are often harder to solve. It also means that evaluating models allow for more details to be modeled. In this sense there is a trade-off between the quality of the model solution and the validity of the model.

If the system problem is of an evaluating nature, then of course an evaluating model is to be used. But even for a system problem that consists of finding an optimum it can be better to build an evaluating model. This is the case if the objective or the constraints are difficult to model, or if "playing" with the model can help the problem owners to get confidence in the model. This is often the case with strategic decisions. New business concepts cannot be created through a modeling approach, but their business value can be validated by it. Programmed problem at the operational level that are of a generating nature usually are solved with a generating model.

Example 9.1.1 A large logistics company wants to change certain processes, without having a clear idea what the alternatives are. Modeling the processes as a dynamic program forces unrealistic simplifications in the model. Therefore it is decided to simulate the systems, and to test various scenarios by changing system parameters in the simulation.

Finally, it can be the case that formulating a satisfying generating model is not wanted or possible, for example because of time constraints, or because of the complexity of the model. In such cases an evaluating modeling technique such as simulation might be an alternative.

9.2 Fluctuations and uncertainty

Without changes that occur inside or outside our systems there is no need to implement any changes. As such, management can trivially be seen as adequately dealing with changes or fluctuations within systems and their environment. In this section we make the important difference between fluctuations that are predictable and

fluctuations that are not (fully) predictable. Whether certain fluctuations are unpredictable or *uncertain* depends on the information available; different managers can have different information.

Example 9.2.1 The number of calls offered to a call center fluctuates over the year, during each week, and during each day. The long-term fluctuations are to some extend predictable. The minute-to-minute fluctuations are very hard to predict. A manager who is aware of a new advertisement campaign can predict an increase in call volume; for a manager who is not informed this increase will come as a surprise.

Galbraith [58] defines uncertainty as the difference between the amount of information required to perform a task and the amount of information already possessed by the organization. It is thus the amount of information that must be acquired during task execution.

Uncertainty is unwanted. "Managers attempt to avoid uncertainty as much as possible" (Turban [157], p. 163). On the other hand, the environment in which companies operate is becoming less and less predictable. For this reason the central issue in strategic management is how to deal with uncertainty (Ansoff & McDonnell [6]). On top of that, managing under uncertainty is more difficult than in a deterministic setting, because the result of an action can change from time to time. Or, as Senge (see Chapter 17 of [140]) puts it: "'Learning by doing' only works so long as the feedback from our actions is rapid and unambiguous."

Uncertainty with respect to certain aspects of our systems is translated in our mathematical models as random variables. We can prove that avoiding randomness, i.e., unpredictable variations, is better. Assume that our performance is represented by a function p, which takes an action a and a random variable X as input. Then it is readily seen that $\max_a \mathbb{E}\, p(a, X) \leq \mathbb{E} \max_a p(a, X)$, assuming that the maximizations exist. At the r.h.s. the maximizing action depends on the realization of X, thus the value of X is known to the decision maker: the variation was predictable and the decision maker could react. On the l.h.s. only its distribution is known, and one action has to be taken for all possible situations at once, independent of the realization.

Example 9.2.2 Sales persons usually have little information about their company's production planning. Therefore, while negotiating a new order, they cannot take the "state" of the production system into account. Nowadays ERP systems can give this type of information to sales persons. Thus they can negotiate orders depending on the remaining production capacity, thereby improving capacity use and decreasing the probability that an order cannot be met.

Uncertainty has many sources. On a system level, we make a distinction between internal or external uncertainty. External randomness enters the model through the interaction with the environment, internal randomness comes from the system itself.

Example 9.2.3 When modeling a service center the stochastic arrival times of customers and their stochastic service times are external randomness. In a production system the stochastic *time-to-failure* of a machine is internal randomness.

We saw that avoiding randomness (in the sense of already knowing the realization) improves performance. For internal randomness this can often be done, although this demands an additional effort. This gives, in principle, a means to quantify the value of management information.

Example 9.2.4 In an industrial environment machine break downs can lead to very high break down costs. By automatic *condition monitoring* the degradation of machinery can be observed and preventive maintenance can be efficiently scheduled.

By definition, external randomness cannot be influenced. However, this is not totally true. On a model level, we can make some part of the environment part of the model. On a system level this translates into trying to make the environment part of your own system, for example by negotiating with suppliers.

Example 9.2.5 Suppliers in a logistics chain have to keep high stocks as to be able to deal with the unpredictable high volume orders that retailers place. Currently, using ICT (Information and Communication Technology), production companies sometimes even know the stock level of the retailer and are thus able to produce at the right time just the amount of products needed.

In the example we see well how information can reduce unpredictable elements in business processes. Many companies nowadays have tremendous amounts of data, with a great potential for cost reduction. (Despite this, certain managers are so overwhelmed by the amount of available data that they do not want to have access to it, trying to avoid "information overload".)

So far we identified system uncertainty with model randomness. However, uncertainty in a system is not always reflected by randomness in the model, it can be that during the modeling process some uncertain parameter is replaced by a constant, for example its average. This is a modeling choice, it can simplify the model enormously.

In fact, randomness in models is of such importance in OR/MS that it splits the research community in two: those that occupy themselves with stochastic models

and those that consider non-stochastic models, a branch which is often called combinatorial optimization.

Example 9.2.6 Navigation software used to use fixed times for traveling road sections, mainly based on length and maximum speed, ignoring random travel times due to the possibility of congestion. Nowadays, manufacturers are adding real-time congestion information to their systems. The next step is using statistical methods to predict congestion and to integrate this in the routing software.

We have to stress that we interpret "avoiding uncertainty" as knowing beforehand what will happen. Having less uncertainty is always better. Avoiding fluctuations is not always preferable, although it usually is. For example, less variability in service times leads to shorter waiting times in queueing systems (Theorem 5.3.2). This is even more so in the case of production networks with multiple stations and finite buffer space between stations (see Chapter 12).

Example 9.2.7 In a production environment, management often prefers a smooth demand. However, situations exist where it is preferable that orders arrive batched together. This is for example the case for machines with long switch-over times from one product to another.

Confronted with fluctuations, there are two positions possible. The first is accepting the fluctuations and dealing as good as possible with it, perhaps using the type of advanced planning methods that are discussed in this monograph. This is the typical mathematician's approach. A second approach however, extensively used in practice and discussed in the business literature, is the one where one tries to reduce as much as possible the (internal) fluctuations. Next to a reduction of fluctuations this leads to an improvement of quality, as fluctuations and quality are intimitely linked. Thus, instead of reducing the influence of fluctuation by a smart planning method, one wants to emphasize the influence of fluctuations as to force workers to reduce them, for example by reducing buffer spaces. These are central ideas in the revolutionary Toyota Production System (see the box on page 193).

9.3 Computational complexity

Models can be complex in different ways: their size can be large, with many details, the time to build an algorithm can be long, and the time to execute an algorithm can vary depending on the model and the solution technique. To appreciate a discussion of scope of size of models we have to understand the notion of computational complexity first.

Computational complexity has to do with running times. For many models there exist algorithms which are guaranteed to finish execution in a time with is bounded by some polynomial function of the problem size. These problems are said to have a polynomial complexity. This guarantees that the execution of the algorithm takes a reasonable amount of time, even for large problem instances. For other problems there is no such algorithm known, only methods with exponential running times are known. These problems are called non-polynomial. Thus computational complexity is not related to the problem size, it has to do with model *properties*.

Example 9.3.1 For a given graph, finding the shortest path between any two vertices can be done in polynomial time (of the order n^3 with n the size of the problem), for finding a minimal length tour that visits each vertex once (the traveling salesman problem) no polynomial-time algorithm is known.

Of course, for any realistic problem an algorithm that finds the optimal solution can be constructed. However, solution methods for these non-polynomial problems are often equivalent with an enumeration of all solutions. For realistic model sizes this leads to unacceptable running times and (computer) resource utilization. To overcome this computational hurdle, algorithms are constructed for many non-polynomial problems that have acceptable running times, without guaranteeing optimality. Some of these algorithms can be seen as procedures that search the solution space in an intelligent way. See the discussion in Section 10.2.

One might wonder if advances in hardware technology will make non-polynomial algorithms have acceptable running times for reasonable problem sizes. However, it is the exponentiality of these algorithms that assures that this is not reasonable to expect. The only hope is for a mathematical breakthrough (e.g., a polynomial time algorithm for the traveling salesman problem), but experts consider this very unlikely. On the other hand, getting the most out of a polynomial algorithm becomes less and less cost-efficient: instead of optimizing your code it is cheaper to let the computer run a little longer. Thus basically any polynomial-time algorithm will do fine, while problems without polynomial-time algorithms should always be solved using heuristics. (There are some exceptions to this rule, but these are merely academic issues.)

It should be noted that for many standard problems (such as the traveling salesman problem) heuristics are developed that perform very well. Thus computational complexity is only of concern to those who design and implement algorithms. A user of a decision support system (discussed in Chapter 10) or a model builder that uses existing routines for model solving can safely assume that very good algorithms exist

for most standard problems. That a heuristic does not necessarily lead to an optimal solution, is of less concern in practical applications. A reason for this is that a model is already a description of reality, thereby introducing modeling errors that are probably bigger than the difference between the optimal solution and the value found by the heuristic.

Having introduced the concept of computational complexity we can now consider other types of complexity.

9.4 Scope and size of models

It is not hard to formulate a model with many system details. Solving such an enormous model on the other hand can be very difficult. We already saw that for very complicated models only evaluating solution techniques such as simulation can be used. And, even if we manage to solve the model using a generating technique, the solution might be too complex to implement: a model solution is often as complex as the model.

Example 9.4.1 Systems to control traffic at freeways use detection loops in the road surface. Placing more loops gives more information. However, designing an algorithm that uses this information in an intelligent way is less simple, and depends necessarily on many variables.

However, sometimes, to avoid suboptimization, or because we need a very detailed policy, we need to take a highly complex model. This complexity has two dimensions: the degree at which details are modeled, and the *scope* of the model, i.e., whether or not many different aspects of a system or process are modeled. While small size models often fall within one of the standard classes, with algorithms or even software readily available, if models get bigger special solution methods need to be developed. Models can even get so complex that they cannot be solved in a single optimization step, and a multi-stage procedure is necessary. Often the outcome of later phases is used to improve the objectives of earlier phases.

Example 9.4.2 In the *vehicle routing problem* goods have to be distributed by a group of vehicles (for more details, see Section 15.3). Several algorithms first assign the goods to the vehicles, by grouping together destinations which are geographically close. What then remains to solve is a traveling salesman problem (TSP) for each vehicle. This initial solution is improved by making goods change vehicles in some smart way, and by solving the corresponding TSPs afterwards.

It is not always the case that a large-size model leads to a complicated solution technique. One of the strong points of simulation is that we can model virtually any level of detail, and nowadays also linear programs with thousands of constraints can be solved.

Another important factor while modeling is the time it takes to find the right solution technique and to implement it. For many projects this is one of the main cost factors. The availability of standard software for some well-known solution techniques such as simulation or linear programming makes implementation times short. For many other models first a solution technique has to be selected (or even developed!), which has to be implemented afterwards. Evidently this takes an enormous amount of time and scientific knowledge, both of which are not always available in companies. Indeed, most of the development of algorithms is done at universities. One can ask the question whether this is a good situation. In any case, communication between companies and universities needs to be very well for the new techniques to be useful in practice. Testing and tuning can also take a lot of time.

In general one can state that the modeling time increases as the model size increases. But there are examples of models which are very simple to formulate, but which are very hard to solve. For example, for certain easily formulated queueing networks that are no analytical solutions known for say the waiting time distribution.

A great advantage of simulation over other models (typically queueing models), even if the model size makes programming necessary, is that it is not impossible to give good estimates of its implementation time. For commercial purposes this can be a major advantage. Also LP models and certain heuristics are relatively easy to implement, but as said before, for example queueing models can be extremely hard to solve. In such cases implementation takes longer, and the implementation time has greater variability. Sometimes this investment pays off compared to simulations, for the simple reason that once the model is solved and implemented results are obtained fast and accurately. Sometimes the model is already optimizing. If the model is of an evaluating nature then it is often easier to use it as the basis of an optimizing procedure compared to simulation, because of the long running times that it can take for simulation to give reliable output. Additionally, the structure of an analytic solution teaches us much about the model; simulation is merely a black box, giving random results. (Although this randomness can be reduced by increasing the number of runs that is made; see the relevant literature on simulation.)

The above holds especially when we compare queueing and simulation for models that describe the system in much detail. For more high level analyses (for example

at a tactical or strategic level) queueing models can be very useful.

9.5 Scale and flexibility

While scope is a property of models, we define scale and flexibility as system properties. Often, decisions at the strategic or tactical level are taken that restrict the possibilities for decision making at the operational level. For example, if resources are assigned to different decision making units, then there is no more possibility to use each others resources if required. Of course, the assignment can be done in the optimal way. However, due to fluctuation in for example demand, this assignment can only be optimal in an average or expected way. Allowing for more flexible resource usage by introducing scale at the operational decision level makes it possible to adapt to fluctuations. The price to pay is the increase in the size of the model: it might be much harder to solve. Also at the organizational level the complications might be quite big, as a more centralized form of decision making is required.

Example 9.5.1 In all the application areas discussed in Part III we find economies of scale. We name a few:
- in manufacturing companies flexible machines replace dedicated production lines;
- in hospitals beds are assigned to patients at the hospital level to avoid saturation at certain wards while beds are free in other wards;
- in multi-lingual call centers bilingual agents have the flexibility to deal with calls of different languages.

An increase in scale always allows for better solutions, because it is always possible to use the "unscaled" policy. A more formal argument is as follows. We have two series of decision problems, $f_1, \ldots, f_T$ and $g_1, \ldots, g_T$, who each have the strategic and operational decision as argument. Then it is optimal to delay the strategic decision and adapt it to the current decision problem:

$$\sum_{t=1}^{T} \min_{a_t} \Big(\min_{u_t} f_t(a_t, u_t) + \min_{v_t} g_t(a_t, v_t) \Big) \leq \min_{a} \sum_{t=1}^{T} \Big(\min_{u_t} f_t(a, u_t) + \min_{v_t} g_t(a, v_t) \Big),$$

with a and a_t the strategic resource assignment decision and u_t and v_t the local operational decisions.

Often it pays off to invest in flexibility to obtain the economies of scale. Examples are bed-side equipment in hospitals or additional training to make workers in call

centers or elsewhere multi-skilled. However, it should also be clear that the advantages of flexibility show diminishing return. This implies that the amount of flexibility has its optimum: after a certain level the decrease in operational costs do not counterbalance the costs of flexibility anymore. Figure 9.1 illustrates this phenomenon. A good example is cross-training agents in a call center: not all agents should be multi-skilled, only a proportion (see Chapter 17 for more details).

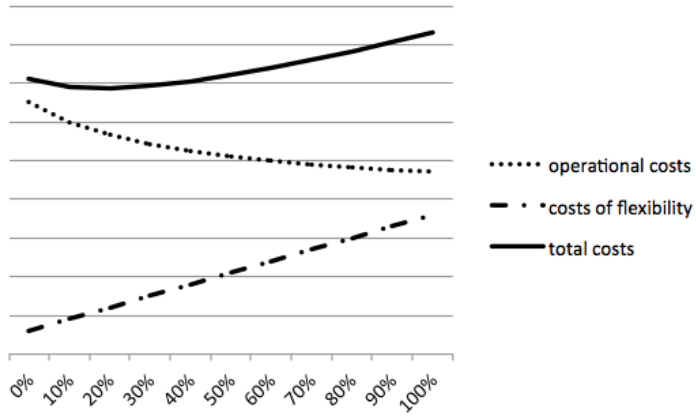


Figure 9.1: Typical form of costs as a function of the level of flexibility.

9.6 Team problems and games

Up to now we tacitly assumed that there is a single problem owner interested in solving problems using modeling. In certain problems however there is no central decision maker, but the decision is decentralized, i.e., there are several decision makers that each control a part of the system. Depending on the objectives of the decision makers these problems are called team problems or games.

Splitting the model up in parts which are each controlled by a single decision maker is not always possible, and if it is possible, then optimizing each part separately can lead to suboptimality. This can however be a viable option, as the model complexity might be reduced significantly by concentrating on parts. The distinction between team problems and games for the problem as a whole is defined as follows.

In games the different decision makers or players have a different objective. Note that we consider objectives as winning a game as different: the players want different players (often themselves) to win. In a business context games can for example be

used to model markets. In the literature one also finds the destinction between games where players can cooperate and where they cannot.

In team problems we assume that all decision makers have the same objective. If they also have the same information, then each decision maker can calculate its own globally optimal decision (and the decisions for all other decision makers). Therefore, from a modeling point of view, it is better to let a single decision maker decide for all others. For a problem to be a real team problem there must thus be different information available to each decision maker.

Example 9.6.1 A node in a computer or telecommunication network has no up-to-date information on the state of other nodes and lines. Dynamic load balancing of the link loads is therefore a team problem. Note that such a network is an interesting example of a system where uncertainty can be reduced. This can be done by sending packets to other nodes informing them of the state of the originating node. The price of this information is the extra charge of the network due to the control packets.

Note that problems are not only team problems because of the impossibility to share information. It might be too costly, or the resulting overall policy might be too complex. In this sense a team problem might be a compromise between a centralized policy that takes all information into account (which is therefore a function of all system details, and thus extremely detailed) and completely decentralized decisions (with the risk of suboptimalization). Note however that it is often easier to compute centralized policies than decentralized policies.

9.7 Robustness

An important issue in modeling is the *robustness* of the model. We call a model robust with respect to a certain input parameter if a small change in that parameter induces only a small change in the output.

Example 9.7.1 In linear programming the robustness of the solution can be quantified, using *sensitivity analysis*. The dual variables, automatically generated by the simplex algorithm, play a major role in the sensitivity analysis. Most LP packages include sensitivity analysis.

Robustness has important consequences for the data analysis. If a parameter is robust, then the accuracy of its estimation is of less importance than for a parameter that is not robust. This should be reflected in the statistical analysis of the input.

Robustness can take many forms. Often it is considered with respect to a number (say next month's interest rate), but it can also be with respect to the form of the distribution of a random variable.

Example 9.7.2 The Erlang B model, discussed in Section 5.4, is robust under changes of the service time distribution. This means that the performance of the model depends only on the average call lengths. For related models, such as the Erlang C model, this is not the case.

Knowing for which parameters the model is sensitive is one side of the coin; knowing which system parameters are likely to change is another. This requires domain knowledge. Here we see again the importance of domain knowledge for the modeler.

Example 9.7.3 In a call center call lengths change only very slowly, while the offered call volume can show drastic changes on any time scale. Therefore it is more important to monitor changes in arrival intensities than in call holding times.

Example 9.7.4 Inventory control is often a trade-off between costs and risk of lost demands. Usually the stock level is controlled by two numbers: if the stock goes below a specified *minimum*, parts are ordered up to the *maximum*. In a model with daily reordering opportunities this policy can lead to high ordering costs, especially if the minimum is close to the maximum. This is less the case for weekly reorder opportunities, as we order at least the demand over a full week. However, if the minimum is not well chosen, we risk lost sales. Which aspect is prevailing depends on the system under study.

We should be careful with basing conclusions on a model which has certain parameters that are likely to change and for which the model is not robust. If it is not possible to come up with a satisfying modeling solution, then this fact should be communicated to the problem owners.

In this section we clearly saw the interplay between data analysis and statistics on one hand and mathematical models on the other. Therefore data analysis should not be considered as an independent activity, but as an integrated part of the modeling activities. Modelers should therefore also be trained in data analysis.

9.8 Choosing the solution technique

For many models different solution techniques exist. Reasons for choosing a technique are the accuracy of the results, the run times, the time to implement the technique, and the possibility for using the technique for other problems. Which one prevails depends on the problem that is to be solved.

Practically speaking, one usually searches for the technique that gives the desired accuracy and run times with the shortest implementation time possible. This often rules out mathematical sophistication.

Example 9.8.1 Many problem are linear in nature, and can therefore be solved using linear programming. For many subclasses of linear programs special-purpose algorithms exist. Often there are no standard implementations; as long as run times remain accaptable it is better to use standard LP software. An additional advantage is that the more general technique allows for more flexibility in case the problem changes.

Special-purpose algorithms should only be used if the run times or accuracy requires it. In other cases general-purpose algorithms are preferably because they allow for more system changes and are easier transferred to other systems.

Example 9.8.2 Standard manpower scheduling problems in call centers can be formulated as integer linear programming (ILP) problems. In practice other techniques such as heuristics are used; ILP lacks the flexibility that local search and other heuristics have.

9.9 Dynamic versus static

System problems often involve time, they are often *dynamic*. Non-dynamic problems are called *static*. For example, the product mix problem modeled in example 8.4.6 is static, no time is involved. Routing and sequencing problems are good example of dynamic problems: in which order should jobs be scheduled or locations visited?

Also for solutions techniques we can make a distinction between dynamic and static methods. Stochastic processes are dynamic, while those that solve problems of the form $\min\{f(x) \mid x \in S\}$ usually are static. It is important to note that there is no 1-to-1 relation between dynamic system problems or models and solution techniques. For example, dynamic routing problems that can be formulated as a TSP are often solved using static techniques.

Note that a problem that involves repeatedly solving a similar problem without interaction between the different moment is not a dynamic model, it is merely a sequence of similar problems. Thus dynamic models involve interaction between the decisions at different points in time.

Example 9.9.1 A transportation company has to deliver goods daily. If goods need to be delivered at a fixed day and if the drivers return each day to the depot this is not a dynamic problem. If goods can be delivered at different days or if drivers can stay overnight at different places it becomes dynamic: earlier decisions influence the situation at a later point in time.

Dynamic problems usually also involve decision making over time. For a deterministic model the future behavior can be exactly predicted, and therefore there is no issue of online decision making. This is not the case if behavior over time is stochastic. In this case there are two possibilities: either the decision takes the evolution into account, or it does not. For these cases we use again the terms static and dynamic: in the former case we call the policy dynamic, in the latter case static. Note that they coincide if there is no randomness involved in the problem.

Example 9.9.2 Most routing software use deterministic estimates of travel times. Letting you guide in the car by the software or taking a printout with you is (mathematically speaking) equivalent. However, if road conditions are taken into account, and if the software in your car updates its travel time estimates, then the optimal route is dynamically adapted. It might even mean making a U-turn, for example if a traffic jam is being formed ahead. If you print your route before traveling using road condition estimates then your routing policy is static.

9.10 Further reading

An excellent source of information on OR models is the series *Handbooks in Operations Research and Management Science* (North-Holland), edited by Nemhauser & Rinnooy Kan. This is a series of books containing high level introductory texts to most areas of operations research, written by experts. Volume 1 [120] deals with combinatorial optimization, Volume 2 [74] with stochastic models, Volume 3 [41] with computing, and Volume 4 [67] with models of logistic systems. Most of the models and subjects discussed in the next chapters have a chapter in one the handbooks dedicated to them.

There are many accessible undergraduate level text introductory OR/MS text books, more appropriate to people new to the field. Without any claim concerning quality or completeness we give two of these books: Taha [149] and Winston [164].

Zimmermann [167] discusses aspects of modeling uncertainty. Next to probability theory he also discusses the possibility of other frameworks such as fuzzy sets. In this monograph we focus on probability.

The observation that the time spent on a modeling project involving simulation is rather short comes from Buzacott & Shanthikumar [32], p. 15. In Chapter 1 of this book some general observations on modeling are made.

The standard text book on complexity is Garey & Johnson [62].

A list of models, with little emphasis on the mathematical aspects, can be found in Chapter 5 of Turban [157].

The difference between generating and evaluating models is introduced in Anthonisse et al. [7].

A recent, quite interesting article on the Toyota Production System is Spear & Bowen [145].

9.11 Exercises

Exercise 9.1 A newsvendor buys newspapers for 0.70 euro and sells them for 1 euro a piece. Leftover newspaper are worthless. Historic data over a year shows that 90% of the sales are in the range $[150, 250]$.
a. Calculate the optimal order level and the expected revenue for demand having a Poisson distribution with expectation 200.
b. Is this a good assumption? What would you take as an approximation for the demand distribution?
c. Calculate the expected revenue for the order level you just found and the new demand distribution.
d. Calculate the optimal order level and the expected revenue for the new demand distribution.
e. Relate your finding to the concept of robustness.

Exercise 9.2 a. Show $\max_a \mathbb{E}p(a, X) \leq \mathbb{E}\max_a p(a, X)$ for X discrete.
The previous exercise implies that uncertainty is always unwanted. This does not hold for fluctations.
b. Give an example where fluctuations lead to lower costs, and one where fluctuations lead to higher costs.

Chapter 10

Model-based Computers Systems

Few models are solved nowadays without the help of computers. Sometimes this is limited to data analysis or the implementation and the execution, once, of a solution technique. At other times the computer is used to *support* the whole decision making process. For certain models solution software is readily available, for other models algorithms have to be implemented entirely. We discuss the different possibilities in this chapter.

10.1 Classification

Model-based computer systems basically come in three flavors, when looking from the user's perspective. The main distinction is whether the user is human or not, and whether the human user is a modeler or a problem owner.

When the user is a problem owner, often a planner within a company, then he or she probably wants a software tool with a good user interface that is specially developed for the type of task that the planner is doing, and that allows the planner to create solutions interactively with the software (otherwise there is no use in having the planner). This allows the planner to take care of aspects of the problem that are not modeled.

Such systems are called *decision support systems* (DSSs). Many definitions of DSSs can be found in the literature, going back to the early 1970s. Keywords in most of them are *computers*, *models*, *unstructured problems* and *human interaction*. We use the following definition.

Definition 10.1.1 *A* DSS *is an interactive model-based computer system that helps humans solve a certain class of business problems.*

Note the use of "a certain class of": we do not want to include in the definition systems with some modeling capacities such as standard LP solvers or spreadsheets. It becomes a DSS when it is especially adapted to a certain class of business problems. Note that with this definition already a small LP model in Excel can be a DSS.

The user of a DSS is usually a problem owner; therefore it is problem oriented. On the contrary, modelers often have tools at their disposition that are focussed on solution techniques, not on application areas. They have a tool for simulation, and one for mathematical programming, and so forth. We will call these *modeling tools*. For convenience they also have (sometimes graphical) user interfaces.

Definition 10.1.2 *A* modeling tool *is an interactive model-based computer system that helps humans solve models with a certain class of solution techniques.*

It should be noted that the user interface of a modeling tool usually gives more freedom to the user than that of a DSS. This makes sense, as a modeler usually has better knowledge of modeling issues than the user of a DSS, he or she has the knowledge to use this freedom. In fact, certain modeling tools can be used to build DSSs.

Modeling tools exist for various techniques. For Monte Carlo simulation (discussed in Section 1.9) there are several add-ins for spreadsheets. They allow you, simply said, to turn spreadsheets cells into random variables for which you can draw a value repeatedly. Powerful graphical reporting tools are supplied.

Let us now consider discrete-event simulation discussed in Section 3.2). Tens of tools exist, usually with a graphical user interface that allows the user to build the model in a intuitive way. Often there is an underlying programming language that can be accessed by the user to model features that cannot be entered using the graphical interface. One attractive feature is to follow the dynamic evolution of the system in a visual way. Indeed, simulation tools allow you first to model the system you want to simulate in a visual way, and then to follow the simulation visually. This is great for debugging and for making a nice presentation.

Although strictly speaking not (always) model-based, we would like to mention that also for forecasting quite a number of special tools exist.

Also for mathematical programming modeling tools exist, for example AIMMS and GAMS. Actually, in these tools different solution methods can be chosen, the modeling environment is to a certain extent independent of the implementation of the algorithm, the solution module. See the next section for a discusssion of mathematical programming solution modules.

Finally, there are systems that take model-based decisions without human interference. This can be the case because there is no time to consult a human decision

maker, because there are too many decision to be made for humans to handle, or because the model-based decisions are good enough and need no human improvement. Let us call these systems *automatic decision systems*.

Definition 10.1.3 *An* automatic decision system *is a model-based computer system that takes decisions without human interference.*

Example 10.1.4 A company is setting up a call center. The lay out is decided upon after a simulation study by a consultant for which he used a simulation *modeling tool*. For workforce scheduling a *decision support system* is bought from a company specialized in call centers. The call routing is done online by an *automatic decision system* that is part of the software of the telephone switch.

10.2 Optimization modules

It can occur, for all three different types of model-based systems introduced in Section 10.1, that they have the same solution generating module in it. These solution modules can often also be bought independently of the user interface, for example by a company who wants to develop a decision support system for a specific class of problems. This is in particular the case for mathematical programming modules.

Mathematical programming software is in many forms available to the modeler: as spreadsheet add-in, as stand-alone program, or as routine that can be called from within a specially written program, in a way that is transparent to the user. Indeed, most of the scheduling programs that many companies use have in the background some commercially available mathematical programming routine running. Software surveys are available on the Internet (see Section 10.6).

10.3 Platforms

For the construction of model-based software the modeler and/or software developer has a number of possibilities at his or her disposition. The choice is often a compromise between the result and the time that needs to be invested. Evidently, a modeling tool for personal use has less requirements on the user interface than a DSS with a large user group. The major types of platforms are spreadsheets, general mathematical (symbolic manipulation) tools, off-the-shelf modeling tools, and regular programming environments. Evidently the former three are mainly employed if there are few users, a choice for the latter can be motivated by the requirements on

the user interface (e.g., when there are many users). Which one of the first three is preferred depends partly on the match between platform and solution method.

Traditionally, DSSs as all software tools were tailor-made computer programs completely developed in a high-level programming language such as C. This demands an enormous development effort: not only the mathematical algorithms need to be implemented, but one also has to take care of issues such as the user interface. Indeed, it happens often that modelers building DSSs complain about the little time they spend on modeling and the large amount of time that they are programming. Of course, even if a firm builds everything itself, that does not mean that user interfaces and algorithms need to be developed from scratch for each modeling project: specialized firms and departments try to reuse software they made before. Next to that there free and commercial routines available for solving standard models such as linear programming (see Section 10.2). It should also be noted that the construction of user interfaces becomes easier and easier with the advent of programming languages such as Visual Basic and Delphi. Building dedicated DSSs is useful and cost efficient for large modeling projects with many users and a model that is often executed (with different data).

We continue with discussing the possibilities in case it is not considered necessary to build the model-based tool in a high-level programming language. This is the case when we are building a DSS or modeling tool with a small user group, often consisting of experts.

If possible off-the-shelf modeling tools are preferred. If there is no suitable tool available, then general mathematical packages such as Maple or Matlab are sometimes preferable to low level programming. It takes less time to implement, but running times are usually longer.

A relatively recent development that does not fall within one of the classes previously discussed is the use of spreadsheet, notably MS Excel. At its base a spreadsheet, a matrix of cells between which simple mathematical relations can be defined, Excel is rapidly becoming the mostly used modeling tool. This is because of a number of reasons. The first is its availability: being part of MS Office Excel is installed on almost any PC. Due to its usefulness for basic (financial) calculations many know how to use Excel's basic functions. This assures that users quickly learn to work with Excel.

The second reason for the popularity of Excel is the availability of model solvers and the flexibility to add your own algorithms. Many statistical functions and even a mathematical programming tool are standard available in Excel. Additional add-ins can be build or bought for performing all kinds of tasks. Finally there is an underly-

ing programming language (similar to Visual Basic) with which virtually everything can be done.

The third reason is the flexibility of the user interface: using standard functions one can quickly build a simple DSS for private use by the modeler. On the other hand, Excel offers many possibilities to make the interface more user friendly and fool proof: one can add buttons, graphics, etc. Altogether, Excel is an extremely versatile modeling tool with which almost any prospective DSS user is already familiar. It allows the modeler to start with just a simple model, and to add features and it user friendly while going through the modeling project.

Example 10.3.1 A firm was competing for a large maintenance project at an airport. To determine the labor costs of the project an LP model was implemented in Excel. The user interface was kept as simple as possible because the DSS was only used by the model builders. When the project was granted the same DSS was used fro scheduling personnel. Of course the user interface had to be changed to adapt to the new users.

For standard business environments there is also the possibility to acquire a dedicated DSS. For example for call center manpower planning problems there are several DSSs on the market (see Chapter 17).

10.4 Decision support systems

DSSs are widely used within companies. They often play a role in which model-based decision making is only of minor importance. In this section we take a closer look at the desired functionality of DSSs.

The definition of Turban [157, p. 85–87] of DSSs is not a definition, but more a list of features. One of them is especially relevant to our view of DSSs: *DSSs support all phases of the decision making process.* This is evident by the fact that a DSS helps solving a business problem, not just a model instance.

The other 13 points are less relevant, sometimes even absurd: E.g., demanding that end-users, only with minor assistance from specialists, build DSSs limits its capacities to only simple LP and simulation models. (This is reflected in the model list in [157, Ch. 5], where only the very basic models are discussed.)

More useful to us is the classification of Anthonisse (based on personal communication). He defines the functionality of a DSS as follows: A DSS is capable of

- selecting,
- generating,
- manipulating,

- evaluating,
- presenting,
- memorizing, and
- in and exporting

data, models, and schedules.

In Anthonisse's point of view DSSs are restricted to scheduling problems, at the operational level. This explain the word schedule in his definition. If we replace schedule by a more general term such as solution his definition could well be used for other types of problems.

Let us discuss some important aspects of DSSs on the basis of this classification. The basic functionality of a DSS is the possibility to *generate solutions* automatically. However, the user can create or changes solutions as he or she wants, by the possibility to *manipulate solutions*. Thus the user is free to use the solution techniques that are implemented in the DSS as much or as little as he or she likes. The user interface should be such that it is very easy to manipulate solutions. Only this should make the user prefer the DSS. Next to that the DSS is capable to *evaluate solutions*, for feasibility and efficiency. Using a DSS is often a repetition of generating, manipulating, and evaluating solutions.

Example 10.4.1 A DSS is developed for scheduling employees of the catering service of a large company. This catering service is characterized by employees with many different skills and short tasks at different locations in the building. Although the DSS is capable of generating solutions itself, this option is used little. Due to personal preferences and capacities that are extremely hard to implement the proposed solutions are of low quality. However, the option to evaluate solutions is very useful. The planner can use this to see if people are scheduled double, if they have to work in overtime, etc. The real gain is in the reporting phase, where a single push on a button generates reports concerning numbers of hours worked, overtime, etc., for each employee. When this work was this done by hand this took the planner two days a week.

The possibility to *manipulate models* does not mean that the user can implement other solution techniques; it means that it is possible to tunes models, for example by setting certain parameters.

Example 10.4.2 A call center employee is responsible for planning and scheduling of the agents working in a call center. For this she uses a DSS especially designed for call centers. There are certain legal restrictions that need to be satisfied all the time. There are also personal preferences (weekly night off, car pooling, etc.) that cannot be satisfied all the time. By changes parameters the relative importance of these personal preferences can be changed.

Presenting solutions is of course of prime importance, to communicate solutions to users and others.

Example 10.4.3 A DSS was built to determine the new location of the warehouses of a firm. The decisions to move warehouses had a high impact on the workforce, because many of them had to move. The DSS was used on a tour along all the warehouses to convince the employees of the necessity of the intended decision. During the meetings new solutions could be entered, evaluated and presented.

10.5 Integration and interaction with other systems

A modeling study is rarely executed without the use of other computer systems. The most important reason is the determination of the input parameters, something that usually needs to be done every time the model is executed. Data relevant for modeling is stored in different types of computer systems. Some of these systems are used for operational and administrative reasons. An important example are ERP systems. ERP stands for Enterprise Resource Planning, a class of software programs that administrate all processes within a firm. The origin of ERP systems is the computer implementation of MRP, a computational method to release jobs in a production environment (see Chapter 12 for the MRP logic). Nowadays modules of ERP systems span diverse activities such as sales, finance, etc. The big advantage of such a system is that it is company-wide. If a representative wants to sell an order, he or she can immediately verify current stock and/or production capacity and take that into account when negotiating the order parameters. After the order has been placed, purchasing can immediately react by buying lacking raw materials, etc. The information available in these ERP systems has a huge potential. Currently the major ERP vendors are developing control tools on top of the transaction data that extract relevant data from the transaction database. Using this they are building data warehouses, with mathematical modeling as one as the possible applications. This facilitates the use of OR techniques in firms with ERP systems.

Systems to support operational processes exist also in other areas. For example, for customer contact there is so-called Customer Relationship Management (CRM) software, and we see a movement in hospitals and health care in general to move from isolated systems (for insurance information, radiology, etc.) to systems that cover all aspects of the health delivery process, including the Electronic Health Record that contains all information concerning the patient.

ERP systems and its equivalents in other sectors are constructed to support the operational processes. For this reason they contain many details of for example current

customers or work at hand, but less historical data. This makes them less suitable for the extraction of management information and therefore also for modeling purposes. This explains the existence of *Management Information Systems*. They allow for the extraction and aggregation of large quantities of data to give new insights and spot trends. A term that is also often used in this context is Business Intelligence, although that also includes the extraction of new relations using statistics and data mining. See Section 8.5 for more on Business Intelligence and its relation to other terms.

10.6 Further reading

The literature on mathematical programming is enormous. Any introductory text book to Operations Research will give the basic methods. For a higher level overview we refer to Volume 1 of the series *Handbooks in Operations Research and Management Science*, Nemhauser et al. [120]. Williams [163] deals nicely with all aspects of solving problems using mathematical programming.

A lengthy text on DSSs is the already cited Turban [157].

Anthonisse et al. [7] discuss what they call *interactive planning systems*.

Jones [82] is a text on user interfaces from an OR point of view; DSSs are also discussed, and many references are given. In the same handbook [41] there is a chapter on mathematical programming systems.

For an overview of vendors and a comparison of mathematical programming tools, see the software surveys published in OR/MS Today, the INFORMS membership magazine, which can best be viewed on the internet at `lionhrtpub.com/orms/ormssurveys.html`. There are special surveys for linear and non-linear optimization. The survey of spreadsheat add-ins also includes mathematical programming tools. There is also a survey on forecasting tools.

A recent book emphasizing the importance of automatic decision systems is Taylor & Raden [151].

Interaction design is an underexposed aspect of planning systems. Getting some background in it, for example by reading Cooper [43], is absolutely worth its time.

10.7 Exercises

Exercise 10.1 A call center is open from 8 to 5. It has a full-time shift from 8 to 5 with a break from 12 to 1, and 2 part-time shifts, from 9 to 1, and from 12 to 4. Between 10 and 4 there should always be at least 5 employees available. At the beginning and

the end of the day there are few calls, requiring only 2 employees. It is the objective to minimize the total number of agent hours.
a. Formulate this as a mathematical programming problem.
b. Solve this problem using the Excel solver leaving out the integer constraints.
c. Solve the problem with the integer constraints.
d. Repeat the same questions for the problem where we replace the full-time shift by 2 part-time shifts, from 8 to 12 and from 1 to 5.

Exercise 10.2 Consider three parallel single-server queues with exponential service times. Arrivals occur to the system according to a Poisson process. Arriving customers are assigned to one of the queues independently, each using the same assignment probabilities. Formulate this as a mathematical programming problem and use the Excel solver to find the assignment probabilities which minimize the average waiting time for your choice of parameters.

Exercise 10.3 A high school is considering buying a tool for making their annual schedule at the beginning of the year (when do which lessons take place as to make the best schedule for classes and professors). Make a list of features for this system, using the classification of Anthonisse.

Exercise 10.4 The same question for the following system: an airline uses multiple fares for each connection. For each flight the number of seats available in each fare class has to be determined as to maximize expected profit. A system is developed that predicts the demand for each fare class and that can compute expected revenue.

Exercise 10.5 For the high school scheduling tool: Is it evaluating or generating? And how about the airline tool?

Exercise 10.6 Does uncertainty or randomness play a role in one of these two systems?

Exercise 10.7 Is uncertainty always caused by a lack of information? Try to find examples of uncertainty that cannot be predicted.

Exercise 10.8 In which of the two systems (the high school and the airline reservation tool) does robustness play a role?

Part III

Applications

Chapter 11

Operations Management

This third part is about some areas of economic activity in which optimization and stochastic modeling is frequently used. We start with a chapter that introduces some concepts in which these areas are different of similar. This way we characterize these areas and show their commonalities.

11.1 What is operations management?

Operations research (OR) is the science that develops mathematical methods to support decisions. It contains methods for deterministic optimization and stochastic performance analysis and optimization with connections to probability, statistics and machine learning. A number of the most important methods were discussed in Part 1.

OR can be applied to many different areas, from product development, to personnel scheduling, to personal finance. For this reason some people prefer the term *decision science* instead of OR. Others prefer *management science*, a term that is born from the belief in the middle of the last century that all forms of management decisions could at some point be quantified and solved by a mathematical algorithm. However, OR is mostly applied to design, manage and plan the production and service delivery processes of organizations. The discipline dealing with these issues is often called *operations management* (OM). Operations management is a generic science: both hospitals and car manufacturers have limited processing capacity, and need models to analyze decisions concerning the capacity. Models used in different areas are similar, but not always the same. Note that OM also uses methods that do not have a mathematical foundation like the OR methods have, but are motivated by experience and common sense. An example is *lean manufacturing*, to be discussed in

Chapter 12. *Industrial engineering* and *logistics* are often-used terms that have a large overlap with OM.

Example 11.1.1 Using optimization methods to perform shift planning for employees, determining the order in which jobs are processed or the determination of a route for parcel delivery are typical examples of OR applied to OM. Examples of OR applied to the product itself are Google's search algorithm, which uses Markov chains to determine the popularity of sites, and optimization used to determine the best places to radiate in cancer therapy.

11.2 Services

Economic activity can be split up in several sectors: production of primary goods, manufacturing, and service. Sometimes a fourth sector is identified, which includes intellectual activities such as research and education. For example, an oil company is part of the primary goods sector, car manufacturers are part of the manufacturing sector, and universities are part of the fourth sector. The hardest to characterize is the service sector.

The products in the service sector are characterized by the fact that the customer is part of the process itself: production and consumption occur at the same time and place, there is no tangible product produced in the absence of the consumer as in manufacturing. Health care institutions deliver services to their patients, consulting companies deliver services, and so forth. In a manufacturing plants goods are produced, perhaps put on stock, and then shipped to the customer, without him or her needing to be present during the production process or even be aware of it. Thus consumption happens after production, not at the same time. Although manufacturing still plays a crucial role in today's economy, the service sector currently represents the majority of economic activity in Western countries.

In manufacturing production always happens before consumption, and also at a different place. It always involves tangible products, thus product have to be distributed to customers. Because production takes place at a separate location, it can be done at a faraway location, that is especially equipped for the manufacturing. The location can also be in another country, for example where labor is available and relatively cheap. Especially China is currently very big in manufacturing. Note that also some services can be *off-shored*, notably custoimer service by telephone or email. Many organizations relocated their call centers to the Philippines or India.

The word service is also used in the context of the *service economy*, which refers to the fact that tangible products, produced in the manufacturing sector, are treated

more and more as services. Instead of getting all the same product from the same location, we see also in the manufacturing sector that the customer needs takes a central place: the product is made or assembled exactly according to his or her wishes (*customization*), it includes delivery at the location, extensive after sales, etc. This means that also companies in the manufacturing sector deliver services. These are sometimes called *product services*, to differentiate from the *service products* delivered by the service sector.

Example 11.2.1 Ricoh is a big Japanese manufacturer of electronic equipment. They used to sell their copiers to customers. Nowadays the copiers remain their own property; instead, they charge per copy, which is the actual need of the customer. Thus instead of manufacturing copiers they now focus on delivering copying services.

Example 11.2.2 A lighting company such as Philips (nowadays split off under the name Signify) used to sell light bulbs and still does. However, customers are not interested in buying bulbs, they want light in their homes and office. Therefore you see this type of companies moving to a model where they sell light to, for the moment, mainly companies. Philips maintains the lights, the customers pays for the light not for the bulbs.

This completely changes the requirements for the bulbs: the manufacturer has an interest to make them energy-efficient, because they pay for the energy, and to make them last as long as possible, as they have to replace them when broken. This is in sharp contrast with their former interests: to sell many light bulb you should assure that they do not last too long. Note that around 100 years ago there was even an international cartel giving fines to participants whose light bulbs lasted more than 1000 hours, the *Phoebus cartel* (See Wikipedia for more information).

Example 11.2.3 Dutch students rarely buy bikes nowadays, they hire them for a fixed fee per month which includes the repair of for example punctures. They buy mobility. *Swapfiets*, owned by the conglomerate PON, a major bike manufacturer, is market leader. Also in other forms of mobility we see a shift towards sharing models where you pay for the transportation instead of for the means. Examples are Felyx scooters in The Netherlands an Belgium and Sharenow with electric cars in many European city centers, where you pay a fee per minute and you can leave the car or scooter right at your destination (as long as it is within the serviced zone).

Note that these new business models forces these companies also to change their operations. Instead of only manufacturing bikes, copiers or light bulbs, they have to set up a field service department that does service at the customer location.

11.3 Orders and reservations

Services are characterized by the fact that production and consumption occur at the same time, in manufacturing production occurs before consumption. A further distinction can be made by looking at the moment of ordering (in the case of tangible items) or reservation (in the case of services). Manufacturing can take place before or after ordering. In the first situation the demand is estimated and items are produced before demand occurs, before the order is placed. We call this *make-to-stock* (MTS). Under MTS consumption can start quickly after the order, with possibly some time for delivery in between. Often ordering and receiving coincide, for example when you visit a shop or supermarket.

MTS does not allow customization. *Make-to-order* (MTO) does: the production is initiated by the customer order. Expensive industrial equipment such as the wafer steppers that ASML makes are often produced in a MTO fashion. Note however that certain components might already available, meaning that parts are produced in a MTS manner. This is close to *assemble-to-order* (ATO), a misture between MTS and MTO. ATO means that in an MTS fashion parts are produced and stocked. After that, in an MTO fashion, parts are assembled to finished products. The *Customer order decoupling point* (CODP) is defined as the point in the supply chain where production is started that is initiated by the customer order. Terms that are synonyms to MTS and MTO are *push* and *pull*. The CODP now becomes the *push-pull boundary*. Note that the terms push and pull strategy are also used outside of operations management to denote comparable concepts.

Make-to-order allows the product to be made according to customer specifications. Thus from the customer order decoupling point the production process has the nature of a service. In the early days of industrialization we saw almost no customization. To illustrate this, Henry Ford is believed to have said about the T-Ford: "You can paint it any color, so long as it's black". Nowadays customization is quite common, necessitated amongst others by the global competition that manufacturing companies encounter. It is a big challenge to combine this possibility of customization with the efficiency of mass production (*mass customization*).

Both in services and manufacturing it is preferable that the time between orders and delivery, or reservation and production, is as short as possible. That is where resources, production capacity, comes into play. Demand and supply should match. That why we discuss next resources and how to match demand and supply.

11.4 Resources

Production is made possible by the presence of a number of resources. These resources consist of raw materials and parts on one hand, and of production capacity on the other. The production capacity is determined by the availability of machines and the availability of people. Often demand fluctuates over time. Different methods exist to match demand and supply. In this section we discuss these different methods and show how different choices are made in different application areas. It is important to realize that the machine capacity is usually little flexible. In car manufacturing the capacity of expensive equipment such as paint shops is fixed. The same holds for the number of rooms in a hotel. In aviation the number of aircraft is usually fixed, some flexible is possible by for example executing maintenance off-season. Human resources are often more flexible. Depending on work contracts people can be forced to take holidays off-season, work on irregular hours, etc. Therefore organizations such as call centers, where the bottleneck in the production capacity is formed by the workforce, have a more flexible production capacity than for example airlines or manufacturing plants. This has important consequences for the way demand and supply are matched.

There are three possibilities: production before, at the same time or after the demand. Only MTS (and ATO until the CODP) has the possibility of producing before the demand. To be able to produce at the same time that demand occurs, you should always have the right capacity. This often means a combination between overcapacity and flexible capacity. Examples are urgent care such as ambulances, call centers, and also taxis and food delivery. Especially in call centers the workforce is very flexible. In aviation and hospitality the capacity is much less flexible, because their most expensive assets are the aircraft and the hotels. Instead, they manipulate the demand such, ideally, the demand meets exactly the capacity for each flight or night.

In MTO (and ATO after the CODP), elective health care and also *field service* (such as repairing kitchen equipement), production takes place after the demand. This means that customers have to wait before their product or service. The waiting time depends on the amount of overcapacity and the flexibility of the capacity. Typically, manufacturing facilities have little flexibility. Elective health care have more flexibility, but its workforce is less flexible than say call center agents, and health care also depends on expensive equipment such as MRI scanners. Thus in all these cases there is a ”stock” of customers waiting to be treated. Of course, customers often anticipate on that, and order products before they need it. Sometimes however this is not possible, it is hard to predict when your dishwasher fails, and then you might have to wait a while before a technician has time to visit you. Similarly for healthcare, you

can schedule in advance a yearly check-up, but you cannot foresee when your knee starts hurting. MTS faces the same situation as MTO, but reversed: there is a risk of high stock levels of finished items. A summary of this section is given in Table 11.1.

application area	**flexibility of capacity**	**order of activities**
MTS	$-$	production demand (distribution) consumption
MTO	$-$	demand production (distribution) consumption
call centers, taxis, food delivery	$+$	demand production consumption
urgent health care	$\pm$	demand production consumption
elective health care, field service	$\pm$	demand production consumption
aviation, hospitality	$\pm$	demand (smoothed by pricing) production consumption

Table 11.1: Overview of the steps per application area.

11.5 Planning levels

To obtain the match between demand and supply, to avoid long waiting times, to avoid lack of stock or too much stock, planning is necessary. Planning is done at multiple time levels. At the lowest level we have *task scheduling*. Capacity is given, and we have to decide on which machines, by whom, and in which order tasks are done. Next there is capacity planning, consisting of different steps. This assures the right numbers of people (and machines) are available at the required times. The number of steps can be different, but at least we should have a step where employee shifts are determined and potentially what they do. At a longer time scale we have to making decisions about our employee pool: do we need to hire new people? It depends on the application area whether these steps are executed in a structured proactive way or in a less structured reactive way.

Example 11.5.1 Hospital departments often work in a reactive way: they hire new nurses or doctors the moment one resigns, and they add appointment blocks when waiting times get too long. Nowadays however we see more and more hospitals using what they call *integral capacity management*: they forecast demand, and plan capacity according to these predictions.

Example 11.5.2 Call centers requires advanced planning methods, because demand fluctuates strongly and these fluctuations should be followed by the capacity almost up to the minute. Most call centers have a long-term capacity planning process which decides about hire and fire and training. There is a shift planning process, often supported by some dedicated call center shift scheduling software tool.

Example 11.5.3 In MTS manufacturing the imbalance between demand and supply is smoothed by stock. This allows for constant production rates, during low-demand periods stock levels increase, they decrease again when demand is high. This makes planning relatively simple: as staffing levels should be constant, there are few types of shifts, and the pool size is best to be kept constant. Some flexibility is required when it comes to skills, to make sure that all jobs in the production process are covered, also in the case of unforeseen absence of some the employees.

11.6 Inventory

From the last example of the previous section it is clear that stock or inventory plays a crucial role in MTS systems. But also in MTO systems stock will prove to be crucial as 'lubricant' in the production process.

In Chapter 12 we study different aspects related to manufacturing: short-term scheduling issues in the first sections, especially the role of variability in relation to inventory and capacity. Information also plays a crucial role, especially when considering the coordination between different parties in supply chain management. The internet is the platform at which these exchanges take place. E-commerce is the language that uses the internet to communicate.

Physical parts and products can be made before the order occurs, up to the CODP. This means that there will be inventory of these parts. Also during the production of these parts and in the later part of the process in which the customer-specific order is produced it can be helpful to stock items. There are different reasons possible for doing this. We classify the types of inventory and discuss their reasons of existence.

Cycle stock Cycle stock has to do with economies of scale: the fact that activities done in larger quantities have lower costs per item, or, equivalently, that marginal

costs are decreasing. The economies of scale can have different sources. A few common examples are:
- in production processes in which machines have to be set up this usually has to be done once in the beginning of a production run, independent of the length of the run. This increasing the batch size leads to lower overall costs and/or time per item;
- many companies charge a fixed overhead per order, next to costs linear in the number of parts ordered;
- transportation costs usually have a considerable fixed component, and a much smaller variable component.

Safety stock Safety stock is a reaction to variability in demand that cannot be predicted. To avoid backorders or lost sales one tries to have, often on top of the cycle stock, some inventory that is there just in case the demand is unexpectedly high. Usually backorders or lost sales cannot be avoided entirely, and some service level (e.g., "not more than 5% backorders") has to be defined. It can also be the case that costs can be assigned to backorders or lost sales. In that case there a trade-off must be made between different types of costs.

Seasonal stock Building up seasonal stock is a way to react to long-term predictable variations in demand. Typical examples are clothing that is sold in a certain season, and Christmas gifts that are produced in an MTS fashion during the whole year and sold during only a short period. To use the available production capacity in an efficient way it is often cost-efficient to produce the whole year round and therefore to build up seasonal stock.

Example 11.6.1 Skiing equipment is manufactured during the whole year, and sold almost exclusively during the winter. For this reason seasonal stock is built up during the whole year. IT is interesting to compare this to hospitality, a service with little flexibility: in the high season hotel rooms are usually much more expensive than during off-season weeks.

Next to the reasons mentioned above for keeping inventory there are several good reasons why we should not keep (too much) inventory. The main ones are listed below.

Investment and handling costs One of the downsides of inventory is that it requires investments. Thus every Euro invested in stock should increase the profit or decrease costs in such a way that it is worth investing it. Next to the investment costs

there are costs for having and maintaining inventory. Costs in this category are the costs of warehouses, etc. Both types of costs are sometimes hard to calculate. How should we calculate the value of a finished item on stock? It is clear that we should count the costs that were made in purchasing raw materials. But should we already count the added value? In financial reporting it is common to do this. However, to avoid expensive stock this can lead to the decision of not producing and keeping raw materials while there is ample production capacity available. This is a form of sub-optimization that needs to be avoided by either giving the right local objectives or to plan globally.

Obsolescence and depreciation Sometimes parts have a limited lifespan and should be shipped before some date. But even parts that do not deteriorate over time can decrease in value over time because of technology improvements, fashion changes, and so forth. This is called depreciation. Note that certain items at stock risk to be stolen. This needs also be taken into account. The measures to avoid theft are part of the investment and handling costs.

Physical limitations In any system the amount of inventory places is limited. If the inventory exceeds this level than a possible consequence in for example a manufacturing plant is a production stop on one or more machines.

We see that there are advantages and disadvantages to keeping stock. The last decades there has been an enormous focus on reducing inventory. It is often motivated by the successes of the *Just-in-Time* (JIT) method, which is part of the *Toyota Production System* (TPS). As an example, cycle and safety of parts of a supplier are kept in stock. This seems logical: we need cycle stock to keep delivery costs low, and safety costs to account for irregularities in the supply. However, in a JIT setting parts are delivered in small quantities from a nearby (production) facility, thereby reducing delivery costs and irregularities. Although TPS has a considerable focus on reducing inventory, it is a misconception that it promotes the removal of all or almost all of it ([145, sidebar on p. 104]).

Up to now we discussed advantages and disadvantages of keeping inventory. However, sometimes we cannot avoid having inventory. This is the case with **in-process inventory**. It occurs in an MTO environment when the demand is temporarily higher than the production capacity, or between production steps by (temporary) fluctuations in production speeds. It can be advantegeous to have some in-process inventory, it avoid 'starvation' of the production step behind it. The TPS also tries to

reduce in-process inventories (and starvation) by reducing fluctuations in machine production times.

A final disadvantage of inventory, in the context of MTO production, is the following. Due to Little's Law the average response times are proportional to the average work in process, that is, the total in-process inventory. Thus a high work-in-process is a symptom of a system with a long response time. This is not only valid for many production processes, but also for most administrative processes.

In conclusion, we have the following types of inventory: cycle stock, safety stock, seasonal stock and in-process inventory. It can be advantageous to have some of each type of inventory, but inventory also has certain disadvantages: investment and handling costs, obsolescence and depreciation, physical limitations, and longer response times.

11.7 Business processes

Up to now we did not go into detail about the content of the production phase. Activities within a company can be divided into those that are part of the *primary process* and those that are not. The primary process is the collection of activities that together make the product(s) of a company.

Within manufacturing companies the primary process can often be identified easily. It is part of the *supply chain*, which is the system, often spanning multiple companies, that allows production from raw materials up to the final product including its distribution and after-sales activities. Nowadays even return flows exist where the depreciated product is dismantled and parts are used again. Then the chain becomes a cycle.

A supply chain is a chain of different activities, often executed by multiple companies. Management has to deal with these different activities, but also with the coordination between these activities. This is what we will call *supply chain management*. Main activities of the supply chain manager have to do with ordering, production and distribution, and their smooth interaction. Many people would also use the term *logistics management*. Originally a military term, the definition of *logistics* is not always clear: it can range from almost all operational activities to only distribution issues.

An important role is played by the *customers*. With a customers we mean the buyer of a product. A customer need not always be a person, it can also be another company. Neither need it be the end user: the customer can use the company's product to add value to it and make a new product. We will make a difference between orders and sales. Sales always occur at the end of supply chain (not taking after-sales

service and depreciation into account). Orders can occur anywhere within the chain.

In the above we focused on the production of a tangible article, characteristic for the manufacturing sector. Of course, a product can be a service: a decision (will I get the loan?), medical treatment, a hotel room, a seat in an airplane, etc. For some of these services general logistics principles apply, for some of them they do not. And not even for all physical products supply chain management is useful; think of one-time products such as the construction of a large building. For these type of one-time activities project planning is useful, logistics principles and supply chain management are less relevant. Project planning is also useful in regular production companies for non-primary processes such as the development of a new product. Chapter 13 is devoted to this subject. Supply chain principles apply to systems where *similar* items are produced in the sense that (almost) the same production steps are followed by the products.

The product offered by for example airline and railway companies and hotels is also different in nature. They actually offer *capacity*: whether this capacity is used or not does hardly make any difference to the activities of the company. It does make a difference to the income of the company, which therefore tries to maximize its income given its available capacity. This activity is called *yield* or *revenue management*. It is the subject of Chapter 18.

Now let us focus on some arbitrary product, physical or not. An important point to make is that a product consists of more than the actual item that one buys; depending on its nature, it has also some other aspects such as time to delivery, its delivery location, after-sales service, etc. The time to delivery plays an important role in the design of the production process, such as deciding between MTO or MTS. Some of these activities are geographic in nature, such as delivery and after-sales repair on the customer location, field service. We dedicated Chapter 15 to planning problems with a geographic aspect.

During ordering and after-sales many contacts between customer and company can occur. Nowadays these contacts are often bundled in a single *customer contact center* (also known as *call center*, by their main activity). Waiting and scheduling issues in call centers are the subject of Chapter 17.

11.8 Further reading

To have a good understanding of how firms function and the way they are organized it can be quite helpful to read at least once a general book on management or to follow a course on it. Nickels et al. [121] is such a book that is easy to read.

Many books have been written on the Toyota Production System. Wikipedia is a good starting point to get a quick idea what it is about.

11.9 Exercises

Exercise 11.1 a. Explain the difference between "service product" and "product service".
b. Give an example of both in the context of planning software.

Chapter 12

Manufacturing

This chapter deals with manufactering, the way production facilities are organized. An important distinction that we make in this chapter is between *flow lines* and *job shops*. The lay-out, the objective and the modeling questions for flow lines and job jobs are very different. We start with modeling flow lines, which can best be characterized as production facilities for large quantities of similar products. Car manufacturing is the prime example. Next we discuss job shops, which are facilities suitable for more heterogeneous products.

Mathematical modeling plays a relatively minor role in the operations management of manufacturing. Improvement frameworks, especially the Toyota Production System (better known as *lean manufacturing* or simply *lean*), have a much greater impact. Variability and capacity considerations play a crucial role in these improvement frameworks. The goal of this chapter is not to give a general introduction to the operations management of manufacturing. Instead, we want to gain a profound insight into the underlying dynamics of manufacturing, which is impossible without the insights from stochastic modeling. General books on manufacturing are listed in the *Further reading* section at the end of the chapter.

12.1 Flow line models

A flow line is essentially a production facility consisting of a number of machines and jobs going from one machine to the next in a fixed order. There are three aspects that need to be specified: the way in which new jobs are initiated at the first machine, the size of the buffers between the different machines, and the distributions of service time durations.

When outside orders initiate the production process (MTO, see page 178) then it is not unreasonable to assume arrivals according to a Poisson process. However, flow lines are most often used for MTS production systems. In this situation production can be initiated in different ways, as we will see in Section 12.3, where we study the coordination of different actors in a production system. One interesting possibility is constant times between the order initiations, thus a process with constant interarrival times. Another frequently occuring situation, with much more interarrival variability, is batch arrivals at random points in time.

Concerning the buffers between the machines (the work-in-process buffers) there are two very different situations that occur in practice. In for example administrative processes there are no limitations to the amount of work in process, leading to infinite-size buffers between the machines or processing steps. In the case of the production of physical items the space is often limited, especially when producing large times such as cars, leading to finite buffers.

Finally we have to consider variability in the service times. In the next section we will consider managing this variability; in this section we will consider it to be an externally given variable and we consider its consequences on the performance.

The most important performance measure of flow lines is the response time distribution or properties of that such as the expected response time. The response time is defined as the total time between order arrival and the end of the final processing step. Less important but also relevant is the buffer usage, or the queue length distribution. The expected waiting time at a machine is related to the expected queue length through Little's Law (see Section 3.8). However, most of the time we are interested in the tail probabilities: what is the probability that the buffer usage exceeds a certain number?

For certain simple flow line models we can use analytical results from Chapter 5. In more complicated situations we have to rely on approximations or simulations. We consider flow lines in various situations by varying the three aspects just discussed. We always assume single servers. Extensions to multiple servers are of practical interest, but lead to little additional insights. We will start with infinite work-in-process (WIP) buffers, exponential service time distributions and Poisson arrivals.

We are going to study the 3-machine example of Figure 12.1. As common in practice the service rates of the machines are different. Using Theorems 5.6.1 and 5.3.1 we can compute the expected waiting time at every queue:

$$\mathbb{E}W_Q(1) = \frac{\rho_1}{\mu_1(1-\rho_1)} = \frac{\lambda/\mu_1}{\mu_1(1-\lambda/\mu_1)} \approx 1.8,\ \mathbb{E}W_Q(2) \approx 4.2,\ \mathbb{E}W_Q(3) \approx 1.3.$$

In total this is 7.3. For the sojourn time we have to add $\sum_i 1/\mu_i \approx 2.2$ units of service

time. Thus in this particular situation orders spend more than 3 times as much time waiting than in service.

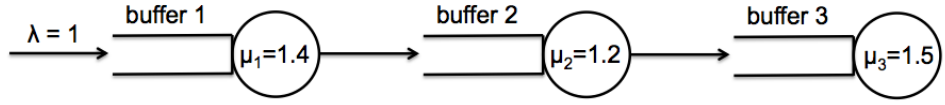


Figure 12.1: An example of a flow line with 3 machines.

For a single queue, when $\rho = 0.5$, then $\mathbb{E}W_Q = 1/\mu$. When $\rho > 0.5$, then the epected waiting time is longer than the expected service time. Note that this is always the case in a well-dimensioned flow line, but whether or not queueing is longer than service depends also on the service time distribution and other features of the system. Before looking into that, we consider the maximum *throughput*.

The throughput of a server or a system is the number of customers that are served on average per unit of time. (We already used the throughput in the context of queueing networks, see page 85.) When the tandem system is stable, if $\lambda < \mu_i$ for all i, then the throughput is equal to λ. When $\lambda \geq \min_i \mu_i$, then the throughput is equal to the service rate of the slowest server. One or more queues before the slowest server are unstable.

The slowest server is called the *bottleneck* of the system: it determines the maximum throughput. Table 12.1 shows the output of calculations for the system of Figure 12.1 for different values of λ.

λ	$\mathbb{E}W_Q(1)$	$\mathbb{E}W_Q(2)$	$\mathbb{E}W_Q(3)$	$\mathbb{E}W$	throughput
0.7	0.71	1.17	0.58	4.68	0.7
0.9	1.29	2.5	1	7	0.9
1.1	2.62	9.17	1.83	15.83	1.1
1.3	9.29	∞	2.67	∞	1.2
1.5	∞	∞	2.67	∞	1.2

Table 12.1: Waiting and sojourn times for a flow line with varying arrival rate.

Of course we see that the relative amount of waiting time increases as λ increases, but we also see that the relative amount of waiting at the bottleneck, compared to the other queues, increases as well. Because of the importance of bottlenecks we can often restrict the analysis to bottlenecks only. Also note that the unstable cases have little practical value: no system with a sojourn time $\mathbb{E}W$ that increases continuously will be left undisturbed.

The first generalization of the system is to general arrivals and non-exponential service times. We consider again the system of Figure 12.1, but with either exponential or deterministic inter-arrival and service times. We analyze a number of different scenarios, keeping the expected interarrival and service times equal. Scenarios are indicated by letter combinations such as DMDM. Each letter has two possibilities, M or D, meaning exponential (Markovian) or deterministic. The first letter indicates the interarrival distribution, the other ones the queues. The results can be found in Table 12.2. The results for scenario MMMM and DDDD are calculated, for the other scenarios the results are obtained by simulation.

scenario	$\mathbb{E}W_Q(1)$	$\mathbb{E}W_Q(2)$	$\mathbb{E}W_Q(3)$	$\mathbb{E}W$
MMMM	1.79	4.17	1.33	9.50
DDDD	0	0	0	2.21
DMMM	0.68 (0.89)	2.47 (3.53)	1.11 (1.22)	6.48 (7.86)
DMDM	0.68 (0.89)	0.81 (1.44)	0.61 (0.92)	4.32 (5.48)
MDDD	0.89 (0.89)	1.18 (0.64)	0 (0.11)	4.29 (3.86)
MLLL	1.33 (1.33)	2.48 (2.10)	0.62 (0.72)	6.64 (6.36)
DLLL	0.30 (0.44)	0.95 (1.46)	0.46 (0.61)	3.93 (4.72)

Table 12.2: Waiting and sojourn times for a flow line. Approximation between brackets.

From the table it can be seen (by comparing scenarion MMMM with DMMM) that deterministic interarrival times reduce the waiting time for all subsequent stages, although the size of the effect reduces further down the line. Also in other scenarios we see the reducing effect of replacing M by D on the waiting times. This is also the opportunity to test the approximation of Remarks 5.3.5 and 5.6.4. In Table 12.2 the approximation is given in brackets for the non-trivial simulated cases. The accuracy is not that good, showing the importance of simulation for performance evaluation.

To illustrate the tandem system and the way variability influences the performance we used exponential and deterministic service durations. Note that in practice durations are often better approximated by a lognormal distribution. It also allows us to fit both first and second moment to real data. Therefore we analyzed the model as well for lognormal distributions, having the same $\mathbb{E}S_i$ as the other cases but with $\sigma(S_i) = 0.5$ for all i. The lognormal distribution is indicated with L in Table 12.2.

The final generalization that we discuss in this chapter is the introduction of finite buffers. Now it can occur that a machine has to stop working because the buffer in front of the next machine, in which it places its finished products, is full. Thus,

there might be two reasons for a station not to work: *starvation*, when there are no items in the upstream queue, and *blocking*, when the downstream queue is full. Thus the performance of a machine depends not only on the upstream queues, but on the whole system. This makes an exact analysis, even for exponential service times, impossible; simulation is the appropriate method for performance analysis. It also makes that the maximum throughput is not solely determined by the slowest server, only in very simple cases there is a simple expression for the maximum throughput. The best way to determine the maximum throughput is to simulate a system in which the first machine has an infinite supply of items to process.

We still have to define how blocking occurs. In manufacturing systems the most common situation is that a finished part stays on the machine until there is a place available in the downstream buffer. This is equivalent to assuming that the buffer has one additional place but that production can only start when there is a place in the buffer. The latter representation is useful when implementing this system in for example a simulation.

To understand the consequences of buffers on the maximum throughput, we consider 2 extreme cases for a system with 2 servers in tandem: ∞ buffer space in between, and without any buffer space. For the ∞ buffer case the maximum throughput is $1/\max\{\mathbb{E}S_1, \mathbb{E}S_2\}$. For the 0-buffer case it can be seen that a new part enters production when S_1 and S_2 has finished, thus the throughput is $1/\mathbb{E}\max\{S_1, S_2\}$. It can be seen that always $1/\max\{\mathbb{E}S_1, \mathbb{E}S_2\} \geq 1/\mathbb{E}\max\{S_1, S_2\}$ (see Exercise 1.1). S_1 and S_2 can even be chosen such that $\mathbb{E}\max\{S_1, S_2\} \approx \mathbb{E}S_1 + \mathbb{E}S_2$, possibly leading to a throughput that is twice as high in the ∞-buffer case (see Exercise 12.2).

Now we report on simulations with varying buffer sizes. Next to Poisson and deterministic arrvals we consider scenarios where the first buffer is saturated, which results in the fact that server 1 is always processing unless it is blocked by a full downstream buffer. We indicate this situation with ∞. We assume that all service duration are lognormal with $\mathbb{E}S_i$ as in Figure 12.2 and $\sigma(S_i) = 0.5$. We put the slowest server at the end to show the effect of The scenarios are now characterized by the arrivals process and the buffer sizes. If the first buffer is finite then rejection can occur. We will not study this situation, thus we assume the first buffer to have infinite size. In the case of a saturated first queue its buffer size is also irrelevant. Therefore we only indicate the buffer sizes of the 2nd and 3rd queue. The results can be found in Table 12.3. From the results we clearly see how the buffers in front of the bottleneck fill up. We also see how the throughput decreases as the buffer sizes decrease.

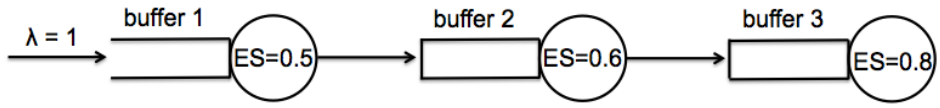


Figure 12.2: An example of a flow line with 3 machines and finite buffers.

scenario	$\mathbb{E}W_Q(1)$	$\mathbb{E}W_Q(2)$	$\mathbb{E}W_Q(3)$	$\mathbb{E}W$	throughput
$M/\infty/\infty$	0.50	0.76	2.17	5.33	1
$M/10/10$	0.51	0.85	2.06	5.32	1
$M/5/5$	0.72	1.01	1.64	5.34	1
$M/0/0$	7245.27	0	0	7248.01	1
$D/5/5$	0.15	0.35	0.97	3.38	1.03
$\infty/\infty/\infty$	-	35264.4	47586.1	82852.4	0.80
$\infty/5/5$	-	3.58	3.31	9	0.81

Table 12.3: Waiting and sojourn times and throughput for a flow line with finite buffers.

12.2 Managing variability in flow lines

In the previous section we saw that variability of service and interarrival times increases sojourn times and reduces the maximum production rate (the throughput) in flow lines with finite buffers. Adding buffer capacity and processing capacity can counter this, but at increased costs. Instead of managing the negative effects of variability, we could take a very different approach. By reducing the variability we improve the performance and reduce the necessity of buffer space and overcapacity, thereby reducing costs. The better the process, the least buffer space is necessary.

The reducting of variability is an essential part of the *Toyota Production System* (see the box on Lean Manufacturing). As an example, consider the way a seat is mounted in a car using its four bolts. Both the average time to complete the job as well as its variability is minimized by measuring exactly in which order the bolts can best be placed, where to place the tools to minimize walking time, and so forth. This procedure leads to a product of a constant level of quality (all bolts in place with the right torque) and a low constant processing time.

One of the methods of Lean consists of reducing on purpose the number of buffer places, thereby making irregularities in the process more visible. Buffer management is implemented using colored cards, so-called kanbans. Items can only be put into stock at a machine when a kanban is available. When no kanban is available then the upstream machine that has finished working on an item is blocked: it can

Lean Manufacturing

Lean Manufacturing, also known as the *Toyota Production System* (TPS), is a set of management and process improvement tools that helped Toyota become one of the leading car manufacturers today. Japanese of origin, it is to a certain extent based on the ideas of the American industrial statistician Deming. One of the central ideas is the reduction of inventory and variability. *Lean* is nowadays not only used in car manufacturing, but also in other production environments and even in services such as health care.

not put the finished item in stock at the next machine. Thus instead of building up in-process inventory upstream production is stopped. The blocking effect can, after some time, also effect higher upstream machines. This way an irregularity can propagate through the whole production chain. A kanban is released when an item is handed over to the next machine or its buffer space, where it receives a kanban belonging to that machine.

Conveyor belts

Certain production processes have no buffer spaces at all. This is typically the case when the items move automatically down the line, for example when a conveyor belt is used. Cars are usually manufactured using conveyor belts. Tasks are split in units of the time to move to the next station. This is called the *takt time*. The takt time might be changed according the required production speed, meaning that tasks have to be split differently. The takt time is the total production time divided by the demand.

In principle, no task is allowed to extend beyond the takt time. In practice, workers can call for assistance when this is likely to happen. In the rare event that one of the tasks cannot be executed then any worker has the possibility to stop the full production line. Evidently, the reasons for such an interruption have to be looked into and to be removed to improve the production process.

A flow line such as a car assembly line often uses parts as part of the production steps. In the case of cars this can range from bolts to complete engines. It is essential for a regular flow that the right parts are available at the right time. Often these parts are supplied by other companies. A regular inventory policy can be used here: when the stock falls below a certain level an order is placed of a certain quantity with a prenegociated lead time. Evidently, the fill rate should be very high: if one item is out of stock then the whole production process stops eventually. The consequence is high inventory levels, both at the manufacturer and the supplier.

A solution to this problem, again in the context of Lean Manufacturing, is the delivery of small quantities of items *just in time* (JIT) for them to be required in the

production process. A prerequisite of this is a very predictible production process in which the supplier is informed well in advance of the required parts. To avoid further variability (due for example to traffic jams) suppliers can be required to keep inventory close to the manufacturing site or even to produce onsite. This creates very close relations between manufacturer and supplier, or even a strong dependence of the supplier on the manufacturer. This type of coordination is also typical for supply chain management, which we will discuss next.

Six Sigma

Reduction of variability in time often goes hand in hand with a reducing of varability in product properties. Suppose that upper and lower specification limits (USL and LSL) are given. Then the goal is that each product falls within these boundary. Six Sigma sets a statistical target for this specification and, just as lean, it proposes a complete infrastructure for process improvement including a continuing process improvement method and specialist certification using martial arts terminology (e.g., one can be a "Six Sigma Green Belt"). The name comes from the fact that the target is that the average product outcome should be at the specification and that the normally distributed deviation should be such that the USL and LSL are 6 times its standard deviation. The probability of a part out of specification is therefore $2\Phi(-6) \approx 2 \times 10^{-9}$.

12.3 Supply chain models

In the previous sections we considered the production of products or parts within a single organization. The use of queueing models was motivated by the fact that production capacity and variations in production play a crucial role in these flow lines. Manufacturing can also be considered at a more aggregate level, as a chain of companies who all deliver their semi-finished goods to the next actor in the *supply chain*. The coordination between actors in the supply chain occurs through orders and deliveries, and is therefore inventory-oriented.

Example 12.3.1 A computer manufacturer produces monitors. The main part, the screen, is produced by a different company. The computer manufacturer uses an inventory policy to avoid stock-outs of screens. This means that when the inventory of screens drops below a certain level then a new batch is ordered, with a pre-nogeciated lead time.

From now on we consider the interaction within the supply chain, we do not look at the processes within the nodes. Thus the capacity constraints that played a

The Master Production Schedule

Central to many supply chains is the MPS, the *Master Production Schedule*. The MPS determines at every moment how many end-items should be produced, usually at a weekly level for the next coming months up to a year. The MPS is based on current stock, sales forecasts and/or outstanding (back) orders. On the basis of the production plan one can calculate backwards at which times the different productions steps should be initiated.

role in determining the production times are now replaced by lead times. This also makes it possible to allow for other activities in the supply chain next to production: assembly, transportation, or a combination of these. They all change one or more characteristics of the item or (future) product. Therefore we call them *transformations*. Inbetween transformations items have to be stored, therefore it is assumed that there are stock points between activities. Thus a supply chain can be seen as an alternating sequence of activities (represented by lead times) and items on stock.

The linear representation of a supply chain is a little too simple. When looking at a single end-product at the end of the supply chain, then we see in general not a chain but an *in-tree*; i.e., we see a directed tree where each node has only one outgoing arc and possibly multiple incoming arcs. This is also called *converging*. When considering more than one end-product we see for example that they are distributed to different outlets: a *diverging* topology. Often we see both. When making the distinction between production and distribution we often see a converging topology up to the final assembly and after that divergence. This is the case when there is a single assembly point.

To profit from economies of scale transportation should be combined. This calls for combining deliveries and optimizing routes for these combined shipments. To make use of these scale advantages related to transportation most companies have *distribution centers* (DC). Additional advantages of scale can be obtained by outsourcing the distribution to a specialized company that does the distribution for multiple companies. In these DCs finished goods are shipped (sometimes after being assembled) together with products destined for a single location or a group of geographically close locations.

DCs carry cycle stock and safety stock: lead times and lot sizes tend to be longer upstream than downstream. The high lot sizes upstream and the low lot sizes downstream necessitate cycle stock. The long lead times upstream force companies to hold big stocks in DCs to avoid running out of products. As long as the DC has a high delivery reliability stock in sales outlets can be kept relatively low. The safety stock at

the DC can be used more flexible (it is not yet decided to which outlet it will go, and therefore there are scale advantages), and holding inventory at a DC is less expensive than holding it at the outlet, because the DC is built and optimized for keeping stock (while the outlet is usually optimized for selling).

As a result the space in the outlets can be used for more different items and expensive inventory locations at outlets become obsolete. Deliveries to outlets should be frequent.

The coordinated supply chain In the supply chain as we considered it up to now every order is negotiated independently, and companies can change supplier whenever they want. This leads to orders of high quantities, i.e., big lot sizes: discounts can be negotiated and overhead is kept low. Also decisions about lot sizes are taken for each actor in the supply chain independently. This leads to suboptimization: what minimizes costs at a single station does not necessarily minimize costs for the whole chain. Thus when every actor in the supply chain minimizes its own costs there will be more inventory then is optimal for the whole supply chain. There will also be much safety stock: downstream demand will also occur in big batches, and there is no information exchange about future order moments.

The solution to these problems is two-fold: deliver frequently with short lead times small lot sizes and send inventory information to upstream actors such that they can anticipate on future demand. Current ICT technology allows companies to exchange information on their stock positions. Thus orders need not be unexpected anymore: they become predictable when one has access to other's stock levels and re-order policies. Of course this demands an extensive cooperation between the companies, including agreements on delivery service levels.

Using all information that comes available in an optimal way is extremely complicated: optimal production and order policies depend on all stock points. This is computationally infeasable and also impossible to implement. The solution to this the use of *echelon inventory*, which is all stock from a certain point on. Thus, when controlling inventory in a DC that delivers outlets, the inventory position of the outlets is added to that of the DC. This is the echelon inventory.

Responsabilities and cooperation between companies remains an issue. Therefore some companies have taken full responsability of their downstream inventory, even if outlets are not owned: 'vendor-managed inventory'. Thus in this situation the decisions concerning orders are not taken by the outlets themselves, but by their suppliers.

The idea of better managing inventory by using additional information can be

taken a step further, namely to removing stock points. This is also made possible by decreased lot sizes that made more frequent deliveries necessary, and the improved cooperation that made deliveries more reliable. An example of this is cross-docking. This lets the supplier or producer prepare the shipments directly for the outlets, depending on their inventory levels. The whole shipment is delivered at the DC, from where is it immediately sent to the outlets. Thus the DC loses its inventory function, but keeps its transportation function.

Example 12.3.2 A retail organization uses cross-docking mainly for its fast movers. Slow movers are kept on stock at the DC. For products on stock there is a lead time of 1 day, for the cross-docking products 2 days (given that the product is on stock at the DC or at the supplier).

By removing stock points along the whole supply chain we see that the old image of every company within the supply chain having its own CODP disappears, and what remains is a completely coordinated chain with a single customer-order decoupling point. Due to reduced lead times this is pushed higher upstream, thus assembly to customer specifications is better possible.

12.4 Job shop models

Different models are relevant for job shops. Roughly speaking, we have stochastic performance models and deterministic planning models. Ideally, we would prefer stochastic planning models, but they are difficult to analyze and less studied in the scientific literature. The stochastic performance models can be thought of as modeling MTO systems, where randomly order arrive which are taken immediately into production. The planning models are more appropriate for situations where production is initiated following a MPS.

Stochastic performance models First we consider performance models with random arrivals. We are interested in waiting times, inventory, and so forth. Thus queueing models are appropriate. The defining property of a job shop is that different types of jobs require different operations on possibly different machines, with different routing. This requires queueing models with different types of customers. No realistic queueing models exist for which the stationary distribution is known. Therefore we have two options: either we simulate the whole model, or we analyze a part of the system, for example the bottleneck. There are two important aspects that have to be taken into account. The first one is that there are often reasons not to serve

the customers according to FCFS, but to give priorities. The model of Section 5.5 is suitable for such an analysis: it gives expected waiting times for a multi-class single-server queue with priorities. The second reason is that when switching from one type of customer to the next there are often switch-over times. Most scientific papers on queueing models with switch-over times are highly technical. We will not discuss them here, but we will discuss switch-over times in more detail in the context of deterministic planning models.

Requirements planning When the due dates of the orders that need to be produced are known then it is possible to plan production. The simplest form of "planning" does not take capacity constraints into account: it just counts backwards in time to decide which subassambly should be produced at what time and in which quantity. This is called *material requirements planning* (MRP), because part of the outcome is how many raw materials are needed and at what time.

MRP logic

Assume that there is ample processing capacity, and that processing times are deterministic. To avoid numerous set-ups process steps are executed in fixed-size batches, the lot sizes. Lot sizes can change from process step to process step.

Under these assumptions an optimal production schedule is easily constructed. The first step is calculating back from the due date the gross requirements by *explosion* of the BOM one level. Of certain parts there is already inventory; taking account for this is *netting*, what results are net requirements. From the net requirements, using the known lead times, we calculate at which moment certain items should be produced. This is called *offsetting*. We cannot order items with any batch size, taking account of batch sizes is called *lot sizing*. Net requirements, together with offsetting and lot sizing, leads to the planned orders.

This procedure is repeated for each level in the BOM, until all orders for all parts and fabrication steps are planned. This process is called material requirements planning. A detailed description can be found in every book on production logistics. Here we just give a simple example: suppose it takes 2 hours to produce a lot of size 4 of a certain product, and 2 parts of a certain type are needed when production starts. There is an order for 10 products, 3 are still on stock. We need to produce 7 items, thus 2 lots of 4. To do so, we need 16 parts as soon as production starts, 2 hours before the due date.

An objection against MRP is that processing times are not deterministic. Random processing times can only get a place within MRP by taking processing times long enough such that with a high probability realizations are shorter. This makes production times longer, leading to increased costs, long idle times of machines and a

decrease in flexibility. The just-in-time methodology can be seen as an answer to this phenomenon.

Another important objections against MRP is that it does not take processing capacity into account. Thus, strictly obeying the MRP logic will lead to coordination problems within the production process: certain process steps are delayed because of capacity constraints (or other effects, such as inventory constraints or production time variablility, as just discussed). This led to the introduction of*capacity requirements planning* (CRP). For each resource, the needed capacity is calculated on the basis of the MRP production plan. This way the needed capacity is calculated and bottlenecks are identified. Solving the resource shortages is more complicated, and requires real planning (for this reason sometimes called *advanced planning*).

MRP and ERP systems

MRP was one of the drivers behind the introduction of large computer systems in production companies. Among the first extensions was demand management, with forecasts and the MPS. By the addition of new modules that connected production to even more business services such as finance it evolved into what became known as *Enterprise Resource Planning* (ERP) systems. They are currently among the biggest software suppliers worldwide, with the German SAP as market leader. ERP systems should be seen as transaction systems, that record all transactions within the company, without much intelligence and/or optimization.

Lot sizes MRP does take switch-over or set-up times into account. To avoid that the machines are switching often production is done in lots of the same product. This increases productivity, but it reduces the flexibility of the machine involved, explaining the desire to build *flexible manufacturing systems* (FMSs), consisting of machines that have no or very short set-up times. In MRP lot sizes depend on the machine and the type of product, but are otherwise constant. Under short-term production scheduling lot sizes will depend on the complete current situation.

Example 12.4.1 A firm produces rolls of plastic in different colors and with different prints on it. The main machines are so-called *calenders* that produce the actual plastics. After that printing and cutting operations can be necessary. Changing from one product to another involves always a phase in which the raw materials of both the old and the new products are mixed. Because of this set-up time it is preferred that production runs are long. The time it takes depends on both the old and the new color. For this reason switch-over time is a more appropriate term than set-up time.

For zero switch-over times, the short-term schedule has no influence on the long-term throughput or waiting times, as long as the server is utilized fully. Simply said: changing the order of two jobs has no consequence on the jobs scheduled afterwards. This is not the case anymore when there are non-zero switch-over times. Changing the order of jobs might also mean a change in switch-over times, and then the whole schedule might change. In the case of non-zero switch-over times, productivity and long-run response times are optimized when the time spent switching-over is minimized. For this reason it is preferably to work with relatively big lots, especially at the bottleneck.

Delay at bottleneck nodes in job shops is the main source of late deliveries. For this reason there is much pressure on the planner to change priorities regularly and to schedule often emergency orders. These are often small batches, and improve in-time deliveries in the short run, but lead to a decrease of productivity, an increased backlog and less in-time deliveries in the long run. For this reason bottleneck nodes should be scheduled such that productivity is maximized. Upstream nodes should be scheduled such that the bottleneck can produce optimally. At downstream non-bottleneck nodes the delivery dates should determine the schedule.

Bottlenecks

The Theory of Constraints is a framework to improve business processes starting from their bottlenecks. It is developed by E. Goldratt and popularized in a number of books (see Page 203). Applied to manufacturing processes, it focuses on preparing the work in process upstream from the bottleneck in such a way that the bottleneck is fully utilized. Downstream the focus is on satisfying customers.

High lot sizes has another disadvantage: high inventory. When lot sizes are determined the complete production process should be taken into account. The following example shows why.

Example 12.4.2 We illustrate the influence of lot sizing on job shop models with the following example. Consider two machines, both needed for the production of a certain item. Machine 1 is used for a range of N types of products, changing product causes set-up costs (instead of times). After processing at machine 1 the items that we consider are processed by the dedicated machine 2, that needs no set-up time. The service time at machine 1 is $1/N$, at machine 2 equal to 1. Customers arrive in all classes at the first machine according to a Poisson process with rate $\lambda < 1$. Thus the loads of both machines are equal. There are holding costs at both stations. (We can assume that for the other $N-1$ product classes there is a similar dedicated machine after machine 1; we concentrate on one of them.)

Interruptions

In most production systems a job cannot be interrupted once it has started. In some systems however one can put aside a job, start working on another job, and finish the first job at some later moment. Examples are certain administrative processes. Contrary to the intuition this reduces the average waiting in certain situations. Whether or not this is the case depends on the distribution of the production times: e.g., in the case of an DHR service time distribution (see Section 1.5) the average waiting time is minimized by working on the job that has received the least amount of service.

In yet other systems a machine can split its processing capacity and work on multiple jobs simultaneously, while keeping the same total processing rate. (This puts it aside from multi-server systems, where the total processing rate depends on the number of jobs present.) A similar effect is reached when jobs are assigned short time slots in a cyclical manner, as it is done in certain multi-tasking computer systems. This model is therefore useful for information processing systems. The policy that consists of sharing the processing capacity in a equal way between all jobs present is called *processor sharing* (PS). From results in Section 5.3 it follows that PS reduces the average waiting time if and only if $c^2(S) > 1$, where $c^2(S)$ is the squared coefficient of variation of S.

First consider machine 1 in isolation. Then high lot sizes are preferable, to avoid high set-up costs. The next class should be the one which has the most items waiting to be processed. (It can even be optimal that the machine idles while there are still products.) Let the lot size be denoted by Q. Under moderate loads and high N we have for the costs $c^1(Q)$ of a single type of item at machine 1:

$$c^1(Q) \approx \frac{K\lambda}{Q} + \frac{h(Q-1)}{2},$$

with, equivalent to the EOQ model, K the set-up costs, and h the holding cost rate. It is an approximation because it might occur that 2 subsequent batches are of the same type, requiring no set-up costs. The probability of this event is small if N is big, making it a good approximation. The optimal value is the EOQ.

Now consider the second machine. The arrival process is in *batches* of size Q (a batch means that the items are grouped). Because of the moderate load we assume that the previous batch has disappeared when a new one arrives. The holding costs per batch are $hQ(Q-1)/2$, per time unit thus $h\lambda(Q-1)/2$. Thus the total costs over both machines $c^T(Q)$ can be approximated by

$$c^T(Q) \approx \frac{K\lambda}{Q} + \frac{h(1+\lambda)(Q-1)}{2}.$$

We see that the holding costs h are multiplied by a factor $(1+\lambda)$; thus the lot size should be smaller than is optimal for a single machine.

Batch processing

Certain machines can handle multiple items at the same time without any or with little additional effort. This seems advantageous but can have quite negative effects on the whole production process: for most items processing is delayed until the batch is complete and after production high inventory is present. For the entire chain it is often better to avoid batch processing. As an example, consider a hospital pathology department which has a batch step in the preparation of tissue for analysis. This delays the process and gives lab employees a pile of work when the batch step ends. The hospital invested in new machine that allowed for much smaller batch sizes. Now the process runs smoother and faster.

Aggregate production planning Let us return to MRP and discuss real planning taking capacity into account. Indeed, while discussing the basic MRP we saw that a major drawback is the inability to control resource utilization. Top-level ways to manage resource utilization are known under the name *aggregate production planning*. It is aggregate in the sense that not all details are modeled. For example, switch-over times are not modeled. We assume that the production for each item demands utilization of certain resources, without temporal constraints.

We present an optimization model for aggregate prodcution planning. Let there be N different products, M resources (machines, labor), and T time periods to plan. Costs consist of a linear function of production costs and inventory. Typical costs related to inventory were discussed in Section 11.6, but it will be clear that early production can mean big investments.

We use the following notation:
- x_{nt} is the amount of products n produced in interval t (the decision variable);
- d_{nt} is the demand for product n at the end of interval t;
- s_{nt} is the amount of inventory of product n at t;
- h_n are the costs of holding one unit of product n one unit of time in inventory (the *holding costs*);
- r_{mt} is the amount of resource m available in interval t;
- u_{nm} is the amount of resource m needed for the production of one unit of n.

Now the production and inventory costs are minimized by the following linear program:

$$\text{minimize } \sum_{t=1}^{T} \sum_{n=1}^{N} h_n s_{nt}$$

subject to

$$s_{nt} + x_{nt} - d_{nt} = s_{nt+1},\ n = 1, \ldots, N,\ t = 1, \ldots, T;$$

$$\sum_{n=1}^{N} u_{nm} x_{nt} \leq r_{mt},\ m = 1, \ldots, M,\ t = 1, \ldots, T;$$

$$x_{nt} \geq 0,\ n = 1, \ldots, N,\ t = 1, \ldots, T;$$

$$s_{nt} \geq 0,\ n = 1, \ldots, N,\ t = 2, \ldots, T+1,\ s_{n1} \text{ given.}$$

Many more features can be added, including overtime (thereby weakening the strong resource capacity constraints), time dependent costs, etc. Note that this is a linear program, and thus the answers need not be integer. If necessary integer constraints can be added, at the cost of considerably longer computation times (see Chapter 7).

Production scheduling Production scheduling has as goal to schedule daily production in such a way that the short-term objectives are met as good as possible. These objectives are mostly related to due dates of customer orders.

Optimal production scheduling leads, mathematically speaking, to a policy where for every moment it is decided for every machine which order should be handled next. This means that decisions can vary with the number of orders, their due dates, machine calendars (e.g., indicating when maintenance is planned), etc. The resulting scheduling problem can be formulated as a mixed-integer linear optimization problem (see Section 7.4). However, to enforce precedence constraints, i.e., that jobs are executed in the right order, many 0-1 variables are needed. This makes a mixed-integer linopt approach not feasible. Local search methods (Section 7.5) or other heuristics can be used here. They usually start with planning the bottleneck.

12.5 Further reading

An excellent book that discusses in much detail the issues discussed in this chapter and much more is Hopp & Spearmann [75].

Dallery & Gershwin [47] gives an overview of mathematical models for flow lines. The most general queueing network model with different types of model for which the stationary distribution is known is the so-called BCMP-model after the initials of the authors, see [19]. The conditions (FCFS, equal service rates) are such that it is not of practical usefulness for manufacturing.

Also Silver & Peterson [141] treats in Part V many issues related to production scheduling, including MRP and JIT. We also recommend Hax & Candea [72].

The "business novel" Goldratt & Cox [66] presents an accessible introduction to Goldratt's Theory of Constraints, that is centered around the treatment of bottlenecks.

For production scheduling O.R. Handbook 4 [67] contains some interesting chapters. Chapter 5 presents queueing network models, Chapter 9 deals with (mainly deterministic) machine scheduling problems, Chapter 11 deals with MRP, and Chapter 12 considers JIT. Also Gershwin [64] considers queueing network models for production systems.

Zangwill [165] gives another view on several aspects of JIT.

Wikipedia is a great source for information on methodologies such as Just In Time or Six Sigma.

Chapter 7 of O.R. Handbook 4 [67] gives an overview of production planning. Pinedo [124] discusses machine scheduling problems.

At http://www.mhhe.com/omc/tours-frames.htm publisher McGraw-Hill maintains a list with virtual company tours.

12.6 Exercises

Exercise 12.1 Change the parameters of Figure 12.1 such that queue 1 and 3 are unstable and queue 2 is stable.

Exercise 12.2 Construct r.v. S_1^k and S_2^k with $\mathbb{E}S_i^k = \beta_i$ for given β_i such that

$$\lim_{k\to\infty} \mathbb{E}\max\{S_1^k, S_2^k\} = \beta_1 + \beta_2.$$

Exercise 12.3 Consider a flow line with Poisson arrivals, 2 servers, possibly finite buffers, and exponential service times. Construct an Excel sheet in which this system is simulated for an arbitrary length of time.

Exercise 12.4 A flow line consists of two machines with infinite in-process inventory space. Arrivals occur according to a Poisson process. Service times are assumed to be exponential with rates 2 and 3, respectively.
a. What is the maximum production rate $\bar{\lambda}$ of this system?
b. Make a plot of the expected waiting times at both machines for the arrival rate ranging from 0 to $\bar{\lambda}$.

Exercise 12.5 A flow line consists of two machines with no in-process inventory space in between. The order arrival process is Poisson.

a. What is the maximal production rate in the case of exponential service times?
b. What is the maximal production rate in the case of deterministic service times?
c. What is the "worst case situation" for given first moments of the service time distribution?
(Hint: Consider the flow line as a single station with a more complicated service time distribution.)

Exercise 12.6 Consider a flow line with Poisson arrivals, 3 machines with exponential processing times and infinite buffers.
a. Compute the expected time in process.
b. Compare this to simulation using the online tool.
c. Do the same for an arbitrary choice of non-exponential distributions using an appropriate approximation.

Exercise 12.7 Consider a flow line with Poisson arrivals, 4 machines with exponential processing times and finite buffers at machine 2, 3 and 4.
a. Compute the expected time in process for all buffers ∞-sized.
b. Suppose the total buffer capacity is 10. Find the distribution of buffer spaces that minimizes the expected sojourn time using the online simulation tool.

Exercise 12.8 The CT scanner in the emergency department of a hospital is used by urgent and semi-urgent patients. Urgent patients have non-preemptive priority over semi-urgent ones. Data analysis has shown the following numbers:

	arrivals (per hour)	average duration (min)	standard deviation (min)
urgent	1	15	8
semi-urgent	3	10	5

Compute the average waiting time for urgent and semi-urgent patients.

Exercise 12.9 Consider a machine that processes two types of parts. The parts have deterministic processing times, with duration 1 and 2 hours for type 1 and type 2, respectively. Parts to be processed arrive according to a Poisson process, with an average of 0.1 and 0.2 per hour, respectively.
a. Calculate the long-run average queue length and waiting time if the processing order is FIFO.
b. Calculate the long-run average queue lengths and waiting times for each type separate and combined under both non-preemptive priority rules.
We add holding costs. For type 1 they are equal to c per hour and part, for type 2

they are equal to $3c$.
c. Calculate the long-run average holding costs if the processing order is FIFO.
d. Calculate the long-run average holding costs under both non-preemptive priority rules.

Exercise 12.10 Consider two production lines, each consisting of two consecutive production steps. The production lines share the same resource for the second production step (but not the first). Production planning is on a MTO basis, and orders arrive according to a Poisson process. Assume that service times are exponential. The order arrival and service rates are given in the following table:

	Order arrival rate	Stage 1	Stage 2
Type 1	1	2	3
Type 2	1.5	2	α

Let the processing order at all stages be FIFO.
a. Calculate the expected total waiting and response time for both product types, for $\alpha = 3$.
b. Calculate the expected total waiting and response time for both product types, for $\alpha = 2$.
c. The same question as a., but now if type 1 has priority over type 2.

Exercise 12.11 Consider a production line with 3 machines and 2 types of jobs. A job of type i visits with probability 0.5 machines i and 3, and with probability 0.5 only machine i. Machine 1 (2) is thus visited by all jobs of type 1 (2), machine 3 is visited by half of all jobs. The arrival processes are Poisson, all service time are i.i.d. exponentially distributed. The average service times of type 1 (2) jobs on both machines they can visit is 1 (2). The arrival rate of type 1 (2) jobs is 0.6 (0.3). The service order at machine 3 is FCFS.
a. Calculate the load of each machine.
b. Calculate the expected waiting times at machines 1 and 2.
c. Describe the arrival process at machine 3. Describe also the service time of an arbitrary job at machine 3.
d. Calculate the expected waiting time at machine 3.
e. Calculate the expected total time that an arbitrary job spends in the system.

Exercise 12.12 Each job on a machine consists of two different operations that are executed consecutively. Each operation has an independent exponentially distributed processing time (with averages β_1 and β_2). Assume that input and output buffers

can accomodate any number of parts. Job orders arrive according to a Poisson process with rate λ.
a. For which parameter values is the waiting time finite?
b. Give an expression for $\mathbb{E}(X+Y)^2$ for general and independent X and Y.
c. Calculate the waiting time for $\lambda = 1$, $\beta_1 = 1/2$, and $\beta_2 = 1/3$.
Now assume that it is possible to change the machine such that the two operations can be executed in parallel, i.e., they start at the same time.
d. Show that the service time is of the form $X + ZU + (1-Z)V$, with X, U, V, and Z independent and $Z \in \{0,1\}$.
e. Give an expression for $\mathbb{E}(X + ZU + (1-Z)V)^2$.
f. Calculate again the waiting time for $\lambda = 1$, $\beta_1 = 1/2$, and $\beta_2 = 1/3$.

Exercise 12.13 A production system consists of 2 production steps. Both take an exponentially distributed amount of time with parameter μ. Production times are independent. Orders arrive according to a Poisson(λ) process. Two different configurations for the system are considered.
a. In the first configuration the two production steps are executed consecutively. There is a large buffer space in front of each production step. Calculate the maximal production rate and the system time as a function of λ.
b. In the second configuration both production steps are executed at the same time. Production on a new order can only start if both steps are finished. There is a large buffer space in front of the combined production step. Calculate the maximal production rate and the system time as a function of λ.
c. Which system has the lowest system time for λ small? Explain this.

Exercise 12.14 What is the direct influence of machine failure to MRP? And what is the long-term consequence of regular failures? Answer the same question for lean. Remember that MRP assumes infinite capacity and deterministic lead times.

Exercise 12.15 Give some advantages and disadvantages of big lot sizes.

Exercise 12.16 The objective of aggregate production planning is to find a production plan that minimizes total weighted inventory costs for given order due dates, capacities, and inventory costs. If there is no feasible plan (i.e., it cannot be avoided that some orders are late) then there is no feasible solution to the correponding linear problem.
a. Construct an example, as simple as possible, for which this is the case.
b. Extend the linear-programming formulation to the situation where penalties are paid for every time unit that a job is late. Make sure that the model remains linear.

Exercise 12.17 Generalize the aggregate production planning model as to account for additional capacity of the resources. Note that utilizing this additional capacity on top of the already available resources costs extra money. Make sure that the model remains linear.

Exercise 12.18 A job shops consists of two machines where each job has to be processed first on machine 1 and then on machine 2, without switch-over times. Job j cannot start before o_j and has to be finished by d_j. Processing times of job j are $s_m(j)$ on machine m. The objective is to minimize the overall *tardiness*, which is defined by $\sum_j (f_j - d_j)^+$ with f_j the time that job j finishes at machine 2.
a. Formulate a mixed-integer linear optimization problem that minimizes the tardiness.
b. Formulate a local search algorithm that can be extended to more general job shops.

Chapter 13

Project Planning

Projects occur in many different settings and environments. In this chapter we discuss project *planning*. We start with planning under deterministic activity durations. After that we discuss in detail the more realistic situation of random durations.

13.1 Introduction

The main conceptual difference between job shops (see Chapter 12) and project planning is that the latter deals with one project at a time, while the challenge of job shops is to make a plan for all products simultaneously, taking the capacity of the resources into account. Presented as such, project planning seems to be simpler than the planning of job shops. However, for job shops we focused on a simple linear structure. For projects we assume that the depencendies between the tasks of the projects are more general.

We define projects as follows.

Definition 13.1.1 *A* project *is a set of nonroutine activities and their interrelations meant to reach a specific goal.*

It is safe to assume that also the goal of the project is nonroutine: if the goal were routine then it is probably better to achieve this using routine activities. It is however possible to obtain a nonroutine goal through routine activities. In a production setting this is typically the case in a job shop. Projects have in common with production processes that they consist of several process steps or activities.

The equivalent of the MRP concept Bill of Materials (see the box on Page 198) in projects is the *Work Breakdown Structure* (WBS). The WBS specifies the project activities and their relations. It is often depicted using a graph model, in which the nodes represent the activities, and the edge the relations between the activities. There are a start and an end node indicating the begin and the end of the project.

In normal top-of-the-shelf (MTS) products price and quality are the main attributes of a product. In projects (just as in MTO logistics) *time* also plays a very important role. In this chapter we concentrate on the time aspects of project management. The most important moment of a project is its finish time. In the next section we learn how to calculate the earliest finish time (EF) of a project, based on deterministic activity durations and resource availability.

13.2 Deterministic durations

The EF of a project is determined by its starting time (ST) and the starting and finish times of the activities. To be able to schedule these we have to know activity durations and resource availabilities. For the moment we assume deterministic activity durations. Consider a project consisting of N activities. Let d_i indicate the duration of activity i. Certain activities can have predecessors, indicating that an activity has to be finished before a sccessor can start. These predecessor relations can be seen as arcs in the graph with activities as nodes. We assume there are no cycles; otherwise it would be impossible to start the project. We also assume that the activities are numbered such that activity i has only lower numbered predecessors. This implies that node 1 is the start node, and node N the finish node. It can be that d_1 or $d_N = 0$, to assure a single start and finish node. See Figure 13.1 for an example with 7 activities.

Now we calculate the earliest finish time of the project. We do so by calculating the earliest finish time of each node (indicated with EF_i for activity i). Assuming that $ST = 0$, and thus $EF_1 = d_1$, we can calculate the EF_i for $i = 2, \ldots, N$ with the formula

$$EF_i = d_i + \max\{EF_j | j < i, i \text{ has predecessor } j\}.$$

Of course, $EF = EF_N$.

Thus the EF_i give the earliest times at which the activities can be ready. Then $ES_i = EF_i - d_i$ represents the earliest time at which activity i can start. It is also interesting to determine the latest times at which activities should start, while keeping to the EF. Define LS_i as the latest time that activity i can start. Of course $LS_N = EF - d_N$. Now calculate LS_i, for $i = N-1, \ldots, 1$, with

$$LS_i = \min\{LS_j | j > i, j \text{ has predecessor } i\} - d_i.$$

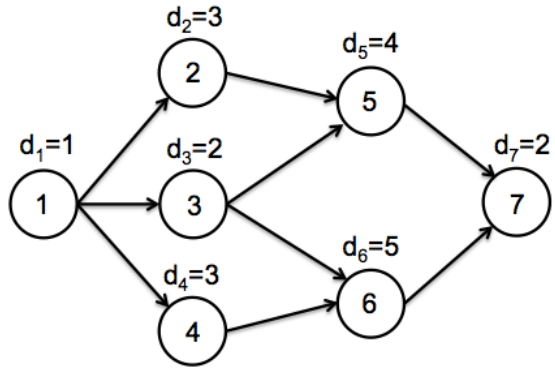


Figure 13.1: An example of the WBS of a project.

Now we can define for each activity its *slack*, defined as $S_i = ES_i - LS_i$. Activities with $S_i = 0$ are called *critical*. A set of k activities $\{i_1, \ldots, i_k\}$ is called a *critical path* if all activities are critical and if $EF_{i_j} = ES_{i_{j+1}}$. For all critical activities the start times are known. For the non-critical they still have to be determined, the possibilities for activity i being all moment in the interval $[ES_i, LS_i]$. This completes the description of the so-called *Critical path method* (CPM).

Example 13.2.1 Consider the example of Figure 13.1 with $ST = 0$. By executing the algorithm we find $EF = 11$. The critical activities are $\{1, 4, 6, 7\}$, together forming the critical path.

A schedule can be represented in a time-activity chart, where there is a horizontal line for the duration of each activity. Such a chart is called a *Gantt chart*. See Figure 13.2 for the Gantt chart of Example 13.2.1. Gantt charts were the first means to graphically display project planning. In software packages we often see Gantt charts with additional features such the critical path in red, arrows indicating the precedence relations, and an additional color for the slack of the non-critical activities.

13.3 Random activity durations

As long as activity durations are deterministic the project will come to an end at exactly EF, assuming that critical activities are started when they can and non-critical before the slack is gone. However, this is a very unrealistic assumption. While discussing production systems in Chapter 12 we already saw that routine activities can have random durations, and this holds even more for the non-routine activities that

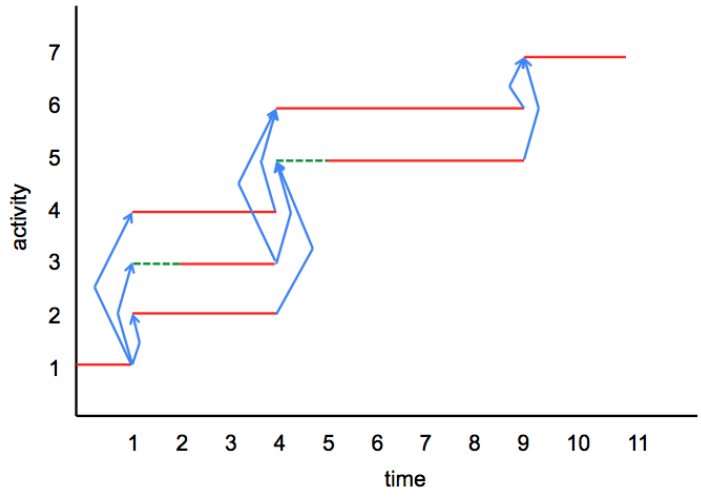


Figure 13.2: An example of a Gantt chart.

projects consist of. We discuss the influence of random activity durations on critical and non-critical activities.

The project duration for deterministic durations is the length of a critical path, which is simply the sum of the durations of the activities on the critical path. However, when the durations are random, then the critical path and the activities on it will depend on the realization. Taking this into account makes the calculation of the project duration difficult. A simplifying assumption is the following. Replace the random durations by their expectations and derive the critical path. Now consider activities on this critical path and their random durations. When the variability of activity durations is limited then one can hope that the duration of this critical path gives a good approximatin for the original random project length. This is the idea behind PERT (*Project evaluation and review technique*). It is an old technique from the 1950s: nowadays, thanks to modern computers and software, it is easier to take all activities into account, for example using simulation.

PERT assumes that the sum of activity durations from the critical path is normally distributed, which makes sense in the situation of a high number of activities on the critical path, as sums of independent random variables are approximately normal (by the Central Limit Theorem, see Section 1.7). In Table 13.1 we give the outcomes for the project of Figure 13.1, with D the total duration. All activity durations are normally distributed with expectations as in the figure and standard deviation 1.

The third column with outcomes is that of simulating the entire project. We see that in 44% of the cases one or more activities become critical that are not on the

	deterministic	PERT (approximation)	simulation (earliest start)	simulation (latest start)
$\mathbb{E}D$	11	11	11.5	12.1
$\sigma(D)$	0	2	1.9	1.9
$\mathbb{P}(D > 12.5)$	0%	23%	29%	41%
other CP	0%	0%	44%	77%

Table 13.1: Outcomes for the project of Figure 13.1.

deterministic critical path. As a consequence, $\mathbb{E}D$ is 4.5% higher than what PERT predicts.

A crucial assumption has been made here: that all activities are started as soon as all preceding activities are finished. Under stochastic durations this is an important assumption: any delay of the start of an activity will increase $\mathbb{E}D$ except for trivial deterministic cases. In practice however, activities are nor started the moment they can. People have the tendency to start the lastest time possible. If we do this for our project then we get the last column, with considerably worse outcomes.

It should be noted that the calculations of Figure 13.1 were repeated for activities with lognormal durations, leading to comparable results.

13.4 Project planning in practice

Project planning as described sofar is purely theory. How does it work in practice? Practice shows that few projects are finished on time, even though project planning methods and software are widely used.

Some of the reasons are as follows. Employees have the tendency to delay working on an activity until it is really necessary. Assume they start work at the lastest finish time minus the median of the duration. Then they are late in half of the cases! It is often impossible to compensate for such a delay. As a result, project plans are usually not based on the medians of the durations, but some safety time is added, for example by taking the 80th percentiles. Unfortunately, this merely motivates workers to start even later, and it is still hard to compensate for delay incurred towards the end of the project.

According to Goldratt (see Section 13.5), these two effects (amongst others) make that projects are hardly ever finished on time and that they take much longer than really necessary. His proposal is to remove these safety "buffers" (the difference be-

tween the 80st percentile and the median) and replace them by a single buffer at the end of the project. To avoid that non-critical activities become critical he also adds a safety buffer everywhere a non-critical activity precedes a critical one. This also solves the problem when to start non-critical activities. In this way there is no more safety time added than necessary and by surveying how much of the buffers is used planners can decide whether or not the project is on schedule. This method is very attractive because it works with deterministic estimates and finish time while it takes account of the randomness in the activity durations.

13.5 Further reading

An easy to read book containing the project planning methods CPM, PERT, and some additional topics, is Awani [15]. Another good read is Klastorin [90]. Goldratt's Theory of Constraints applied to project management (on which the last section is largely based) is described in [65]. Belson [21] is a book chapter describing project management for health care process improvement projects.

13.6 Exercises

Exercise 13.1 A project has the following activities:

Activity	Preceding activities	Duration
A	-	2
B	A	3
C	A	2
D	C	1
E	B,D,G	2
F	-	3
G	C,F	2

Assume for the moment that there are enough resources.
a. Compute the earliest finish time of the project and all earliest and latest starting times of the activities. (Hint: renumber first the activities.)
b. Give the definitions of slack, critical activity, and critical path.
c. Compute in the example project the slack of each activity. What is the critical path? Suppose that activities B and C use the same resource. Therefore they cannot be scheduled at the same time.

d. What is now the earliest finish time of the project?
e. Prove that the solution to d. gives indeed the earliest finish time possible.

Exercise 13.2 Consider the project from Exercise 13.1, but assume that the activities have lognormal distributions with all variance 1 and expectations as given in the table.
a. Approximate the 10, 25, 50, 75 and 90% percentile using the PERT technique.
b. Do the same thing using simulation, for example in Excel.

Exercise 13.3 Reproduce the numbers of Table 13.1, for example by a simulation in Excel.

Exercise 13.4 Consider the project from Exercise 13.1 with durations as in Exercise 13.2. Calculate by simulation the 80th percentile of the project duration when "latest start" is used for all durations based on the 80th percentile of every activity. Compare this to the 90th percentile of the project duration when for each activity "earliest start" is used.

Chapter 14

Reliability and Maintenance

In this chapter we study reliability and maintenance. Reliability is the study of systems that can fail or stop functioning. Maintenance deals with replacing or repairing systems or components of systems prone to failure with the aim of improving their availability.

Reliability and maintenance become more and more important in our society. The systems around us become increasingly complex; at the same time individuals and companies depend on an increasing number of complex systems for their daily functioning. Examples are the electricity network and online banking.

First we study the general theory of reliability, then we turn to maintenance. Before that we give some definitions and we make some general remarks.

14.1 Introduction

This subject consists of two main parts: *reliability* and *maintenance*. Reliability is about systems that can fail, maintenance about systems that can fail but also be repaired. Sometimes we will use the term *availability*. It deals with the functioning of a component or system at a specific point in time (also called *pointwise* availability). Reliability is more general, and deals also with the time-dependent behavior of components and systems.

For certain systems such as production lines maintenance plays a crucial role in meeting production plans and satisfying due dates of customer orders. In other systems reliability is a goal by itself. Indeed, an airplane should function during its whole flight. As such, maintenance does not seem to play a role. However, seen over a longer period, maintenance of components is crucial to assure that the airplane is

reliable during all of its flights.

In reliability theory we often assume that a system consists of *components*. Whether or not a system is functioning depends on the components. It is not always the case that all components should function for the system to function; this dependence can be more general, by including *redundancy*. Components of systems can themselves consist of components, thus a system can be studied at different abstraction levels. Another term for component is *subsystem*.

We start by considering the reliability of a single component or, equivalently, a system as a whole. An important role is played by the hazard rate and its form over time. Then we continue with systems, and we see how the reliability of a system is a function of the reliability of its components. After that we move to maintenance. Again, first for a single component. Next to *corrective repair*, when the system has failed, we consider *preventive repair*, before the system has failed. This is often much cheaper and/or takes less time. Repair can be *time-based*, *usage-based*, and *condition-based*. Often we do not know the level of wear of a component, thus we can only base our preventive maintenance on the time since its last replacement or the time it has been used.

Example 14.1.1 The oil filter of a car is typically replaced every year, which is an example of time-based maintenance. The oil itself is replaced after a fixed number of kilometers, which is usage-based maintenance. Tires are replaced when the tread depth is below a certain level, which is condition-based maintenance. When a car won't start or breaks down while driving you need corrective maintenance. This is usually more expensive, perhaps you have to be towed away and pay more for the emergency repair. Next to that, you're likely to have to change plans or miss some appointments. We have not yet mentioned a final type of maintenance: combining maintenance in a single appointment is called *opportunity-based maintenance*.

The last subject we treat is maintenance of systems, mostly based on Markov models. These models depend on many aspects, such as the form of redundancy, the number of repairmen, whether is based on wear or time, whether spare parts are in *warm* or *cold stand-by*, the status of other components, etc.

Sometimes it is hard to define what the goal is of the study of reliability and maintenance policies. In many systems reliability strongly influences the overall objective, but it is hard to set a specific objective for the reliability. Incorporating reliability in the overall objective is desirable, because optimizing maintenance by itself is a form of suboptimization. Integrating maintenance and production planning, based on the current state of the production system, is called *condition-based manufacturing*, and is

likely to become more popular with the advent of more advanced planning systems and increased possibilities of condition monitoring.

Example 14.1.2 A steel plant has some detoriorating parts which decrease the quality of the produced steel. The maintenance plan are the production plan are optimized jointly such that orders not requiring the highest quality are produced when the plant is in a slightly deteriorated state.

14.2 Reliability of a single component

In this section we consider single components, without maintenance. The usual way to represent the life time of a component is by its hazard rate (see Section 1.5). We define certain properties of positive random variables related to hazard rates.

In what follows we use increasing and decreasing in the non-strict sense (it would be mathematically correct to use non-decreasing and non-increasing instead).

Definition 14.2.1 *We call a positive r.v.* IHR *(*DHR*) if its hazard rate is increasing (decreasing).*

Of course IHR (DHR) stands for Increasing (Decreasing) Hazard Rate. This definition holds only for distributions with a density; a more general definition can be formulated that would include more distributions. For example, under this more general definition a constant would also be an IHR distribution.

To understand what IHR or DHR implies, we define first $X(s) = [X - s|X \geq s]$, the remaining life time giving the component has age s. Note that $\lambda_{X(s)}(t) = \lambda_X(s + t)$. Then, using Equation (1.12), we obtain the following:

Theorem 14.2.2 *If X is IHR (DHR), then $\mathbb{P}(X(s) > t)$ and $\mathbb{E}X(s)$ are decreasing (increasing) in s.*

Note that the inverse need not be true; counterexamples exist.

Life times, and thus the hazard rates are influenced by the following types of physical events:
- **burn-in**. This is the phenomenon that a new component might fail early, for example due to construction errors;
- **wear-out**. The phenomenon that a component deteriorates in time;
- **external events**. The fact that components can fail due to reasons not related to the component, but due to other components or the environment.

Let us consider different classes of components. If a component is not subject to burn-in or wear-out, but only to random failures independent of age, and to external events, independent of time, then it makes sense to choose a constant hazard rate. This implies that the life time is exponentially distributed. If there is only burn-in (and possibly failures independent of age and external events), then the life time distribution is an DHR distribution. Of course, most manufacturers of components try to avoid burn-in effects, by testing products before they are shipped (avoiding initial failures), or by introducing burn-in periods, before selling the products.

In case of only wear-out, the life time distribution comes from an IHR distribution. Of course, wear-out is hard to avoid, and many components therefore have an IHR distribution.

If a component is both subject to burn-in and wear-out, then its hazard rate will likely have a "bath-tub" shape. In the *burn-in phase* failures are mostly due to burn-in. After that we have a phase where failures are rare, called the *change phase*. Finally wear-out starts playing a major role as we enter the *wear-out phase*.

Remark 14.2.3 A popular distribution in reliability is the so-called Weibull distribution, named after a Swedish engineer. It is a generalization of the exponential distribution, and it has the advantage that, depending on the parameters, it can model both IHR and DHR distributions.

In many cases it is not sufficient just to predict the life time distribution of a component: we want to have a better approximation of the remaining life time as time progresses. This means gathering additional information on the component. There are in principle two (partly overlapping) ways to do this. The first consists of dividing the component in subcomponents and gathering information whether their life times are expired. The second is *condition monitoring*, where the level of wear-out of a component is measured. By obtaining in this way additional information the life time distribution is not changed (unless maintenance is performed), but while information is gathered, the moment of failure can be better estimated. This method is gaining more and popularity as sensors get cheaper and more easier connected to the internet. Also the methodology used to predict failures is changing, from Markov models modeling explicitly the state of deterioration to black box machine-learning models.

Up to now we tacitly assumed that the time to failure does not depend on the way the component is used, but only on the time. This is called *time-based failures*. Next to time-dependent failures we have *operation* or *usage-based failures*. These failures depend on the way the component is used. Note that if we define the time as the time that the system is used, then time and operations-dependent failures might well coincide.

Example 14.2.4 In a production line certain machines are always on, whether they are used or not. Therefore their failures are time-dependent. Certain tools only wear out when they are used for producing parts. Often we see that failures are neither fully time-dependent nor operations-dependent, but a mixture of both.

A special case of operations-dependent failures is *cold stand-by*. A redundant component is in cold stand-by when there is no wear-out as long it is not used. When a redundant component wears out just as a component that is used then we call this *warm stand-by*.

14.3 Availability of systems

In this section we consider the relation between the pointwise availability of the system and its components at a specific point in time t, without taking the dynamics into account. Throughout this section, let y denote the state of the system: $y = 0$ means that the system is down, $y = 1$ means that the system is up. Similarly, let y_i denote the state of the ith component. If the state of a component is random, then it is denoted with Y_i. In this case the state of the system also becomes a random variable, denoted with Y.

Let $\phi : \{0,1\}^n \longrightarrow \{0,1\}$ be the function that indicates, based on the state of the components, whether the system is functioning or not. We call ϕ the *structure function*.

If the availabilities of the components are random, then also $Y = \phi(Y_1, \ldots, Y_n)$ is a $0/1$-valued random variable. It is our objective to calculate the system availability $\mathbb{P}(Y = 1) = \mathbb{E}Y = \mathbb{E}\phi(Y_1, \ldots, Y_n)$. This can be very cumbersome, certainly if there are dependencies between the components. For this reason we assume that the components are modeled in such a way that all the Y_i are independent.

Example 14.3.1 The main office of a company is connected with its computing center through two separate connections hired from different companies. However, these connections share the same physical cable. Software problems occur independently, but hardware problems (e.g., an excavator breaking the cable) are dependent. To make all components independent the connections could me modeled by three components: one modeling the physical connection, two modeling the software connections. This occured in reality, the connections shared the same deep-sea cable without the company knowing that. When a fishing boat broke the cable the company's fleet of trucks was grounded for a couple of days.

Calculating $\mathbb{E}Y$ remains difficult, even if the $\mathbb{E}Y_i$ are independent. The function ϕ is a complicated non-linear function and computing its expectation is cumbersome. A method exists that simplifies this task somewhat, which we will explain next.

The vector $(Y_1, \ldots, Y_n)$ is completely characterized by the probabilities $p_i = \mathbb{P}(Y_i = 1)$, because all the Y_i are independent. We define the function $\Phi : [0,1]^n \longrightarrow [0,1]$ as follows:

$$\Phi(p_1, \ldots, p_n) = \mathbb{E}\phi(Y_1, \ldots, Y_n) = \mathbb{P}(Y = 1).$$

Note that $\Phi(p_1, \ldots, p_n) \neq \phi(p_1, \ldots, p_n)$, ϕ is not even defined for $p_i \in (0,1)$!

Assuming that the Y_i are independent and that ϕ is known, then Φ can be calculated by using the definition of expectation and enumerating all possibilities:

$$\Phi(p_1, \ldots, p_n) = \sum_{(y_1,\ldots,y_n)\in\{0,1\}^n} \prod_{j:y_j=0} (1 - p_j) \prod_{j:y_j=1} p_j \, \phi(y_1, \ldots, y_n).$$

Note that the summation has 2^n terms. Thus this way of calculating is only an option for small systems. Later on in this section we study a method for finding Φ for general systems, but first we look at the system function for some special configurations.

Series systems Systems where reliability plays a minor role are often designed such that every component is crucial for the system to function. This is the simplest possible structure for a system, often called a series structure. Its system function is given by $\phi(y_1, \ldots, y_n) = \min\{y_1, \ldots, y_n\} = \prod_{i=1}^n y_i$. The last representation can be used to calculate Φ:

$$\Phi(p_1, \ldots, p_n) = \mathbb{E}\phi(Y_1, \ldots, Y_n) = \mathbb{E}\prod_{i=1}^n Y_i = \prod_{i=1}^n \mathbb{E}Y_i = \prod_{i=1}^n p_i.$$

Clearly a series system will often lead to a low availability, as every component needs to function.

Example 14.3.2 A series system consists of 100 independent components, each having a 99.9% availability. The availability of the system is thus $0.999^{100} \approx 1 - 100 \cdot 0.001 = 0.9$. Complex systems can have much more than 100 components. To increase the availability of the system the availability of the components should be increased or some form of redundancy should be introduced.

Parallel systems An often used method to improve the availability of a system is adding redundancy. This can for example be done by installing the same component several times in parallel. Let us consider a system with several similar components in parallel. Suppose we have n components, and only one needs to function to assure that the whole system functions. For this system $\phi(y_1, \ldots, y_n) = \max\{y_1, \ldots, y_n\}$. For now and later use we derive the following lemma.

Lemma 14.3.3 For arbitrary $y_i \in \{0, 1\}$, $\max\{y_1, \ldots, y_n\} = 1 - \prod_{i=1}^{n}(1 - y_i)$.

Proof $1 - \max\{y_1, \ldots, y_n\} = \min\{1 - y_1, \ldots, 1 - y_n\} = \prod_{i=1}^{n}(1 - y_i)$, using the fact that $1 - y_i \in \{0, 1\}$. □

Thus we also have $\phi(y_1, \ldots, y_n) = 1 - \prod_{i=1}^{n}(1 - y_i)$. Taking expectations gives $\mathbb{E}Y = \Phi(p_1, \ldots, p_n) = 1 - \prod_{i=1}^{n}(1 - p_i)$.

Example 14.3.4 A parallel system consists of 2 independent components, each having a 99.9% availability. The availability of the system is $1 - (1 - 0.999)^2 = 0.999999$. Now if we assume that in Example 14.3.2 every component consists of two parallel subcomponents, then the availability becomes $0.999999^{100} \approx 1 - 100 \cdot 0.000001 = 0.9999$. An alternative would be to take two parallel systems, then the availability becomes 0.99. Thus we can better place components in parallel then systems in parallel.

A generalization of a parallel system is the so-called k-out-of-n system, which means that out of n components at least k should function. The system function ϕ is given by $\phi(y_1, \ldots, y_n) = \mathbb{I}\{\sum_{i=1}^{n} y_i \geq k\}$.

Example 14.3.5 The k-out-of-n system with $k = 2$ and $n = 3$ has structure function

$$\phi(y_1, y_2, y_3) = \mathbb{I}\{y_1 + y_2 + y_3 \geq 2\},$$

and

$$\Phi(p_1, p_2, p_3) = p_1p_2(1 - p_3) + p_1(1 - p_2)p_3 + (1 - p_1)p_2p_3 + p_1p_2p_3 = p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3, \tag{14.1}$$

which follows from enumeration.

For general k and n there is no expression known for $\Phi(p_1, \ldots, p_n)$. An exception is when $p_i = p$ for all i. In this case it is easy to see that

$$\Phi(p, \ldots, p) = \sum_{i=k}^{n} \binom{n}{i} p^i (1 - p)^{n-i}.$$

General systems Now we consider general unreliable systems, and we will derive a method to find Φ. However, in our analysis we cannot allow for full generality. We will make the following assumptions on the structure functions:
- $\phi : \{0,1\}^n \longrightarrow \{0,1\}$ with n the number of components;
- $\phi(0,\ldots,0) = 0$ and $\phi(1,\ldots,1) = 1$;
- ϕ is increasing, i.e., $\phi(y) \leq \phi(y')$ if $y_i \leq y'_i$ for all i.

For example, the last assumption implies that components cannot make the system go down by being up; every component should have a positive influence on the system availability. Systems satisfying these properties are called *monotone*.

Let us also introduce the following notation. For each $S \subset \{1,\ldots,n\}$ let e_S be the n-dimensional vector with entries $(e_S)_i = 1$ if $i \in S$, 0 otherwise. (Thus $e_{\{i\}}$ is the usual ith unit vector.)

Definition 14.3.6 *A* minimal path set *is a set of components $S \subset \{1,\ldots,n\}$ such that:*
- $\phi(e_S) = 1$*;*
- $\phi(e_{S'}) = 0$ *for all $S' \subset S$ with $S' \neq S$.*

The following result holds.

Theorem 14.3.7 *For a specific structure function ϕ, let $S_1,\ldots,S_m$ be the collection of minimal path sets. Then*

$$\phi(y_1,\ldots,y_n) = \max_{i=1,\ldots,m} \prod_{j\in S_i} y_j. \tag{14.2}$$

Proof For a vector y let A be the set of functioning components, i.e., $e_A = (y_1,\ldots,y_n)$.

Assume that the r.h.s. of Equation (14.2) equals 1. Then there is a minimal path set (m.p.s.) S with $A \supset S$ and $\prod_{j\in S_i} y_j = \phi(e_S) = 1$. Because ϕ is increasing we find $\phi(A) = 1$.

Now assume $\phi(A) = 1$. One by one we try to make the components non-functioning, in such a way that the system remains up. Call the resulting set S. It is readily seen that S is a m.p.s., and for this reason the r.h.s. of Equation (14.2) equals 1.

Thus the l.h.s. of Equation (14.2) is 1 iff the r.h.s. is 1. Equality holds because 0 and 1 are the only possible values. □

Thus every system can be seen as consisting of several series systems in parallel. Note however that the same component may occur in several different series of components! Therefore we cannot directly use Lemma 14.3.3.

Example 14.3.8 The 2-out-of-3 system of Example 14.3.5 has as minimal path sets $\{y_1,y_2\}$, $\{y_1,y_3\}$, and $\{y_2,y_3\}$, and thus as structure function

$$\phi(y_1,y_2,y_3) = \max\{y_1y_2, y_1y_3, y_2y_3\}.$$

We see that every component appears in two series. Thus, although $\phi(y_1, y_2, y_3) = 1 - (1 - y_1y_2)(1 - y_1y_3)(1 - y_2y_3)$ by Lemma 14.3.3, $\Phi(p_1, p_2, p_3) \neq 1 - (1 - p_1p_2)(1 - p_1p_3)(1 - p_2p_3)$, which can be concluded directly by comparing this form with Equation (14.1).

We can compute $\mathbb{E}\phi(Y_1, \ldots, Y_n)$ in the following way. By Lemma 14.3.3

$$\phi(y_1, \ldots, y_n) = 1 - \prod_{i=1}^{m} \Big(1 - \prod_{j \in S_i} y_j\Big).$$

The product can be worked out (e.g., using a symbolic manipulation package such as Maple), which results in a sum of series. It should be noted that $y_i^2 = y_i$ as $y_i \in \{0, 1\}$, which simplifies these expressions enormously. Now all y_i can assumed to be random variables, and taking the expectation is nothing else than replacing all Y_i by p_i. This way we obtained a general method for computing Φ.

Example 14.3.9 Applied to the 2-out-of-3 system this method gives

$$\phi(y_1, y_2, y_3) = 1 - (1 - y_1y_2)(1 - y_1y_3)(1 - y_2y_3) =$$

$$y_1y_2 + y_1y_3 + y_2y_3 - y_1^2y_2y_3 - y_1y_2^2y_3 - y_1y_2y_3^2 + y_1^2y_2^2y_3^2 = y_1y_2 + y_1y_3 + y_2y_3 - 2y_1y_2y_3,$$

using the fact that $y_i^2 = y_i$. Thus

$$\mathbb{E}\phi(Y_1, Y_2, Y_3) = p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3,$$

the same as what we obtained in Example 14.3.5.

Although better than enumeration this approach is still cumbersome. Why not simulate the system? A complicating factor is that failure is often a *rare event*: it happens rarely and therefore it is hard to find a reliable estimate. As an example, suppose we first simulated $k = 10000$ times a system with 5000 failures, not a rare event. Then the confidence interval for the availability is

$$\Big[\frac{1}{2} - \frac{\Phi^{-1}(1-\alpha)S_k}{\sqrt{n}}, \frac{1}{2} + \frac{\Phi^{-1}(1-\alpha)S_k}{\sqrt{n}}\Big] \approx [0.49, 0.51]$$

for $\alpha = 0.025$ and with $S_k \approx \sqrt{0.5 \times 0.5} = 0.5$, using the expression of the variance of the alternative distribution on page 12.

However, in the case of a rare event, with for example 10 failures, then $S_k \approx \sqrt{0.999 \times 0.001} \approx 0.03$ and therefore the confidence interval becomes $[0.0004, 0.0016]$. Thus for rare events the confidence intervals are smaller, but relative to the value they are larger. The situation gets worse when the events occur less frequently, for example 1 out of 10^8 times. This a typical value for the reliability of aircraft, nuclear plants or sea dikes.

14.4 Reliability of systems

Up to now we just considered the availability of a system at some (unspecified) point in time. In this section we will also take into account the life time distributions of the components. Several questions are of interest, such as the expected life time of a system, or the translation of component properties such as IHR to the system level. Unfortunately few results can be obtained in this area. We start this section with a result for series systems.

Series systems Let X denote the life time of a series system, and X_i the life times of its independent components, for $1 \leq i \leq n$. Denote with λ the hazard rate of the system and λ_i the hazard rates of its components. Again, let Y (Y_i) be the state of the system (component i) at some time t. Then $\mathbb{P}(X \geq t) = \mathbb{P}(Y = 1) = \prod_{i=1}^n \mathbb{P}(Y_i = 1) = \prod_{i=1}^n \mathbb{P}(X_i \geq t)$. This gives a simple way to compute the system life time distribution based on the component life times.

An interesting question is if X inherits certain properties of X_i. We have the following properties.

Theorem 14.4.1 *For series systems*

$$\lambda(t) = \sum_{i=1}^n \lambda_i(t),$$

and consequently X is IHR (DHR) if all X_i are IHR (DHR).

Proof From $\overline{F}(t) = \mathbb{P}(X > t) = \prod_{i=1}^n \mathbb{P}(X_i > t) = \prod_{i=1}^n \overline{F}_i(t)$ it follows that

$$f(t) = -\frac{d}{dt}\mathbb{P}(X > t) = \prod_{i=1}^n \overline{F}_i(t) \sum_{i=1}^n \frac{f_i(t)}{\overline{F}_i(t)} = \overline{F}(t) \sum_{i=1}^n \lambda_i(t),$$

the middle equation by the chain rule. Therefore $\lambda(t) = f(t)/\overline{F}(t) = \sum_{i=1}^n \lambda_i(t)$.

If λ_i is increasing or decreasing for all i, then so is $\sum_{i=1}^n \lambda_i(t)$. □

Parallel systems We continue our analysis with parallel systems. Here we find $\mathbb{P}(X > t) = \mathbb{P}(Y = 1) = 1 - \prod_{i=1}^n (1 - \mathbb{P}(Y_i = 1)) = 1 - \prod_{i=1}^n \mathbb{P}(X_i \leq t)$. However, a simple characterization of the hazard rate does not exist for parallel systems. Neither do IHR or DHR components imply the same for the system.

Example 14.4.2 Let $n = 2$, and X_i exponentially distributed with parameter i. Then $\mathbb{P}(X \geq t) = 1 - (1 - \exp(-t))(1 - \exp(-2t)) = \exp(-t) + \exp(-2t) - \exp(-3t)$. Its hazard rate is easily calculated. It is first increasing and then decreasing, as can be seen from Figure 14.1. Thus $\lambda(t)$ is neither IHR nor DHR.

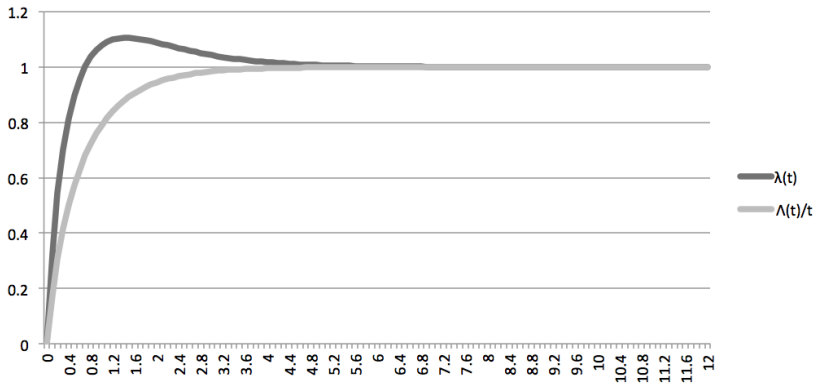


Figure 14.1: $\lambda(t)$ and $\Lambda(t)/t$ for the model of Example 14.4.2.

General systems In the remainder of this section we focus on systems that show some form of wear-out such as IHR components. In Example 14.4.2 we saw that the IHR property is not preserved at the system level. Is there a related property that is preserved? Such a poperty exists: when $\Lambda_i(t)/t$ is increasing for all i then $\Lambda(t)/t$ is also increasing, for all monotone systems. (Remember that Λ is the *hazard function*; see Section 1.5.) A distribution having the property that $\Lambda(t)/t$ is increasing is called "increasing hazard rate average" (IHRA). Note that $\lambda(t)$ increasing implies that $\Lambda(t)/t$, thus all IHR distributions are also IHRA. According to the result just mentioned, the life time of the system of Example 14.4.2 should be IHRA. This is indeed what we see in Figure 14.1.

For a proof of the closedness property we refer to the references mentioned in Section 14.7.

14.5 Maintenance of a single component

Let us consider a single component that is not only prone to failure, but that can be repaired as well. Up times and repairs alternate, and all up times and repair times have the same distribution. The time U that the component is up has distribution F_U,

the time R it takes to repair the component has distribution F_R. Then, from renewal theory (see Section 3.3, Example 3.3.2), we know that the long-run fraction of time that the component is up is given by $\mathbb{E}U/(\mathbb{E}U + \mathbb{E}R)$. By the results of Section 3.5 we also know this is also equal to $\lim_{t\to\infty} \mathbb{P}(\text{system is up at } t)$, under a non-lattice condition.

Evidently we repair or replace the component after failure, but In many systems it is possible to do some form of maintenance before failure. This is what we call *preventive maintenance* (PM). It takes often less time or is cheaper. Let us first assume that preventive maintenance makes the component "as new", thus its life time starts all over. Now we have a decision problem: under which conditions and when should we conduct PM?

In the time-based situation without condition monitoring the only reasonable maintenance policy is: conduct preventive maintenance when the life time of the component reaches τ units. If it fails before τ, conduct regular *corrective maintenance* (CM). Let the time P to execute preventive maintenance have distribution F_P. We assume $\mathbb{E}P < \mathbb{E}R$, otherwise it would not be useful to execute PM. It is also required that the component shows some form of wear-out, as the next example shows.

Example 14.5.1 Let U be exponentially distributed with parameter λ. If we execute PM at τ, then

$$\mathbb{E}U = \int_0^\tau t\lambda e^{-\lambda t}dt + \tau e^{-\lambda\tau} = \frac{1}{\lambda}(1 - e^{-\lambda\tau}).$$

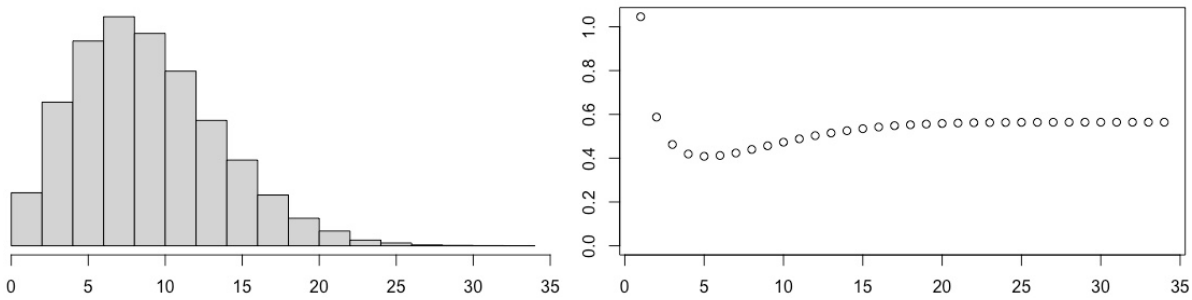
Thus the long-run fraction of time that the system is up is equal to

$$\frac{\mathbb{E}U}{\mathbb{E}U + F_U(T)\mathbb{E}R + \overline{F}_U(T)\mathbb{E}P} = \frac{\frac{1}{\lambda}(1 - e^{-\lambda\tau})}{\frac{1}{\lambda}(1 - e^{-\lambda\tau}) + (1 - e^{-\lambda\tau})\mathbb{E}R + e^{-\lambda\tau}\mathbb{E}P}.$$

Taking the derivative to τ shows take this fraction is increasing in τ. Thus it is optimal never to undertake PM, giving a long-run average availability of $(1 + \lambda\mathbb{E}R)^{-1}$.

Sometimes it is hard to determine the optimal τ analytically. In that situation backward recursion can be used with the life time as state, see Section 7.7. Another simpler possibility is simulation.

Example 14.5.2 We sampled from a Weibull(2,10) distribution, and assumed that repairs take no time, but CM costs 5 and PM costs 1. The histogram of the lifetimes and the costs for each integer-valued time-based PM policy are shown below.



Earlier in this chapter we discussed condition monitoring. By using additional information on the condition or state of the component the PM decision can be improved. Again, using Markovian decision models the optimal PM policy can be computed. The typical PM decision now becomes: if the condition of the component reaches condition x, then PM should be executed.

14.6 Maintenance of systems

For the maintenance of systems with multiple components there are two basic situations to consider: those in which the maintenance of one component does not influence the maintenance of another component, and those where this is the case. Examples of reasons why maintenance on one component influences other components are:
- there are not enough repairmen to repair all failed components at the same time;
- it is cost-efficient to repair all failed components at the same time.

First we consider the situation where the maintenance of different component is independent. Then all components are independent. Let a_i denote the long-run availability of component i, i.e., $a_i = \lim_{t\to\infty} \mathbb{P}(\text{Component } i \text{ is up at } t)$. E.g., if there is only corrective maintenance, and U_i and R_i are the up and repair times of component i, then $a_i = \mathbb{E}U_i/(\mathbb{E}U_i + \mathbb{E}R_i)$. Let a be the system availability. Then, using results of the previous section, we find $a = \Phi(a_1, \ldots, a_n)$.

More often we find that the components become dependent because of the maintenance policy. In general these situations are hard to analyze, and numerical computation and simulation are often the only available methods.

As an example of a system that can be analyzed analytically we consider the situation of a k-out-of-n system with s repairmen ($s < n$), with as special cases $k = 1$ (parallel system) and $k = n$ (series system). We assume equally distributed exponential up (repair) times U (R), with parameter λ (μ). If we identify the components with

customers, and the repairmen with servers, then we see that this maintenance model is equivalent to a queueing model with a finite source of n customers, s servers, and queueing. Using results from Theorem 5.4.7 we can compute the long-run availability of the system. For the general k-out-of-n systems it is given by $a = \sum_{j=0}^{n-k} \pi(j)$, with $\pi(j)$ as in (5.9)-(5.11).

14.7 Further reading

The standard text on reliability is Barlow & Proschan [18], including a proof of the result at the end of Section 14.4, which was first derived in Birnbaum et al. [25]. An introduction to the mathematics of reliability is Chapter 9 of Ross [130]. An advanced mathematical text is Aven & Jensen [12]. An accessible text that covers roughly the subjects of this chapter but in more detail are the lecture notes by Arts [9].

A journal entirely devoted to reliability is *IEEE Transactions on Reliability*.

There is a extensive literature on rare event simulation. A good starting point is Heidelberger [73].

14.8 Exercises

Exercise 14.1 In Example 14.3.4 it is stated that "we can better place components in parallel then systems in parallel". Show this.

Exercise 14.2 Consider a system with 2 components A and B in series. There are two spare machines C and D. Machine C can replace A or B, but when C replaces B, then also D is necessary. (Thus D is only used to "help" C replace B.)
a. Find all minimal path sets.
b. Determine ϕ and Φ.

Exercise 14.3 We consider the availability of a system with 2 components, named A and B, in series. There are two spare components C and D. Component B can be replaced by component D. Component A can be replaced by component C, but only if D replaces B as well. Machines fail independently, whether they are used or not.
a. What are the minimal path sets of this system?
b. Give the functions ϕ and Φ (giving expressions for the availability in a deterministic and in a random environment).
c. If all components have an exponential time to failure with rate 1, and they are all up, what is the expected time to failure of the system?

Exercise 14.4 To increase the availability of a computer system a second was placed in parallel. Both have an availability of 98%. However, it was found that 1% of the unavailability was due to a problem with a common power supply.
a. Model this system using independent components.
b. Formulate ϕ and Φ.

Exercise 14.5 Calculate confidence intervals as on Page 225 for $n = 10^8$ and 5×10^7, 10^4, 10^2 and 1 successes.

Exercise 14.6 Consider a system with 2 identical components, with uniform life time distributions on [0,1].
a. Calculate their hazard rates. Is it IHR on [0,1)?
b. Calculate the hazard rate of the system if the components are placed in series. Is it IHR on [0,1)?
c. Calculate the hazard rate of the system if the components are placed in parallel. Is it IHR on [0,1)?

Exercise 14.7 a. Reproduce Figure 14.1.
b. Construct a distribution which is IFRA but not IFR.

Exercise 14.8 Show that if X is IHR, then X is also IHRA.

Exercise 14.9 Make a plot of Λ of the system of Example 14.4.2 and convince yourself that the system life time is IHRA.

Exercise 14.10 A component can be in 10 different states. It deteriorates in discrete time: after every time unit it stays in the same state with probability 0.5, with the same probability it moves to the next state. State 1 is the new state, it fails when it reaches state 10. Preventive maintenance takes 2 time units, corrective maintenance takes 5 time units.
a. Determine the PM policy that maximizes the long-run average up time when the state is observed.
b. Do the same when the state is not observed.

Exercise 14.11 Reproduce the graphs of Example 14.5.2.

Exercise 14.12 Consider a system with a lifetime that is gamma distributed with 2 phases and mean 10. When the system fails we replace it by a new one. We also have the possibility to do preventive maintenance which is 5 times cheaper than corrective

maintenance.
a. What is the optimal time to do preventive maintenance? Use some appropriate (approximative) technique. Give also the average costs.
b. Suppose we know the phase of the gamma distribution. What is then the optimal policy? And the average costs?

Exercise 14.13 Consider a k-out-of-n system with n identical machines. The time to failure of each machine is exponentially distributed with mean α.
a. Give a formula for the expectation of the time to failure of the system.
b. Give the failure rate of this system for $k = 2$ and $n = 3$.
We add a single repairman to this system, the repair time is exponential with mean 1.
c. Model this system as a birth-death process.
d. For arbitrary n, give an expression for the long-run fraction of time that the system is up.

Exercise 14.14 Consider a system consisting of n identical machines. The time to failure of a machine is exponential with mean 10. The system is up if at least one machine is up. If more than one machine is up, then these spare machines are in cold standby.
a. For $n = 2$, give the probability that the system is up at time 20.
b. What is the failure rate of this system?
We add a single repairman to this system, the repair time is exponential with mean 1.
c. For arbitrary n, give an expression for the long-run fraction of time that the system is up.
d. How many machines are needed to make this fraction at least 0.9999?

Exercise 14.15 Consider a system consisting of n parts that each fails, independently of the other parts, after a time that is uniformly $[0, 1]$ distributed, i.e., the density of the time to failure of each component is 1 in $[0, 1]$, 0 otherwise.
a. Give the definition of the failure rate and the system function.
b. Compute the failure rate of the life time of a single component.
c. Let the system consist of n parts in series. Compute the failure rate of the life time of the system.
d. Let the system consist of n parts in parallel. Compute the failure rate of the life time of the system.

Exercise 14.16 a. Consider a 2-out-of-3 system with a single repairman, exponential times to failure, exponential repair times, and warm stand-by. When the system is

down the working component can still fail. Give a formula for the long-run probability that the system is up.
b. Answer the same question but now with cold stand-by, and when the system is down the working component cannot fail.

Chapter 15

Distribution and Field Service

Many services delivered by companies involve going physically to the customer location. In this chapter we consider operations management problems in which the geographic aspect plays an important role, such as food and parcel delivery, emergency services, and field service.

Roughly speaking, the planning methods can be split in two groups: those that deal with *positioning* of for example ambulances and taxis, and those that deal with *routing* of parcel delivery vehicles, field service technicians, home care nurses, etc.

We start with a taxonomy and then we discuss methods for both types of problems. We also discuss capacity planning problems and the *car stock problem*, which is relevant for field service.

15.1 Taxonomy

The delivery of many services requiring going physically to the customer location, for example when something tangible is delivered or picked up such as food or a parcel. This can be the final step in a MTO or ATO process, but delivery is considered a service. It can also be that a service is delivered at the customer location, such as first aid in the case of a traffic incident, daily help with washing and clothing of elderly people, or the repair of kitchen equipment.

There are two type of models for planning distribution. In the first workers wait during a shift until they are assigned a service request. After having fulfilled this service request they wait for the next request, either at some base station or at some convenient location. Examples of this type of service are food delivery, taxis and ambulances, but also the maintenance of essential industrial equipment. The scheduling

problem is mostly concerned with positioning the service providers at places where they are close to locations where new demand might occur. This is especially important for ambulance service because of the tight service level requirements: a typical requirement is that for 98% of all requests the ambulance arrives within 15 minutes at the location. Capacity planning (see Section 11.5) is necessary to ensure that the right number of service providers is available at each moment of the day.

In the second model the service providers execute *tours*. A tour is the journey undertaken by a single service provider in which he or she visits one or more customer locations and then returns to the base location. This base location can be some central location or it can be different for different drivers or technicians, for example their house. Often a tour takes a work day, and there are multiple parallel tours every day of the week. The scheduling challenge is to divide the service requests on a day between the different providers and then find the best route for each of them. Problems of this type are known as *vehicle routing problems*. There are quite a number of variants, for example with time windows or with capacity constraints on the vehicles. The problem of finding a single route without additional constraints is the *traveling salesman problem*, which already is known to be hard to solve. Often a service request is scheduled for the next day (as in parcel delivery in urban areas) or on regular basis, determined by the nature of the service (as in home health care). Some problems however have an additional planning step: determining the day of service. In this situation the planner tries to fill routes by making appointments, often at locations that are close together to avoid extensive traveling times. This is usually the case with technicians who visit homes or other locations to repair and install equipment.

Also for the the second type of problem related to vehicle routing capacity planning is crucial. Too many service providers leads to unnecessary costs; too few leads to work in overtime and/or long waiting times before the visit of a technician. In the next sections we go into more detail about the models.

15.2 Location covering problems

In this section we study location covering problems. The most simplest one is as follows. Suppose we have a set of locations I and a set of possible base stations J. Let J_i be the set of base stations that cover location i, for example those base station from which i can be reached within 15-minutes of driving. The question is: where should we locate p vehicles such that the highest number of locations can be covered, i.e., reached within the target time? This *maximum covering location problem* (MCLP) can be solved with the following integer linear optimization (ILO) problem with binary

decision variables x_j and y_i:

$$\max \sum_{i \in I} y_i$$

subject to:

$$\sum_{j \in J_i} x_j \geq y_i \; \forall i \in I;$$

$$\sum_{j \in J} x_j \leq p;$$

$$x_j, y_i \in \{0, 1\} \; \forall i \in I, j \in J.$$

The decision variables have the following interpretation: $x_j = 1$ iff a vehicle is stationed at base station j, and $y_i = 1$ iff location i is covered.

For an illustration see Figure 15.1. There are 12 locations (the circles) and 5 base stations (the triangles). Suppose there are 3 service providers. Where should they be located?

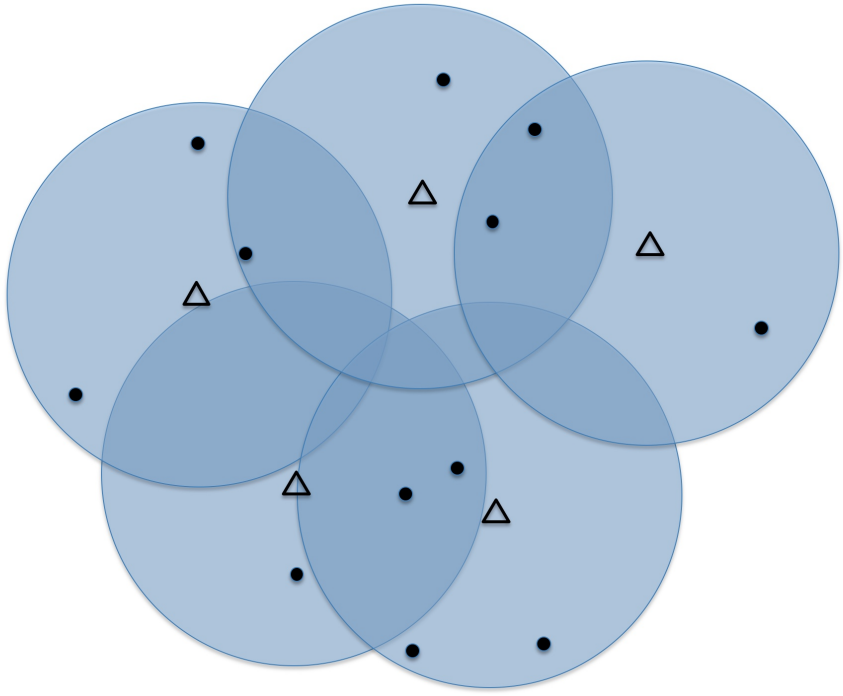


Figure 15.1: An illustration of the MCLP.

Extensions of this model are mostly developed in two directions. The first is that you could require multiple vehicles per location, or, similarly, count the expected

availability of the vehicles that cover a certain location. A second extension is adding the possibility of repositioning vehicles when one is occupied and no longer able to cover its locations.

The models discussed in this section are especially suited to emergency services such as ambulances: they have a relatively low occupancy and go most of the time to the first emergency that occurs (ignoring the fact that ambulances are also used for less urgent services and for scheduled transportation, for example from hospital to hospital). For problems with longer *acceptable waiting times* (AWT) there might be a choice between which customer to visit first and by whom, simply *earliest due data first* (EDDF) is not always the most efficient solution. This leads us to a variant of the *vehicle routing problem* as discussed in the next section which needs to be solved multiple times during the day, every time when a service request arrives or is finished. The overall objective is to maximize the fraction of customers served within the AWT. This is called the *service level* (SL). To obtain a satisfactory service level some overcapacity should exist to overcome peak demand. To avoid technicians idling this overcapacity can be used to perform preventive maintenance on functioning machines. Note that an algorithm that minimizes the fraction of late customers might fully ignore how much these customers are late. This "perverse" effect might be avoided by taking an objective other than the service level, for example, $\max(0, w - a)$ with w the waiting time of a customer and a its AWT. For a more elaborate discussion on service level definitions see Chapter 17 on call centers.

Example 15.2.1 A manufacturer of copiers has a fleet of service technicians driving around. AWTs depend on the contracts customers have, but are typically between 4 and 16 business hours. As soon as a technician finishes service on a machine he or she connects to the company's IT system to update information on the machine they just serviced and to receive their next service location. In the back an algorithm runs that repeatedly updates the schedule based on the most current information.

15.3 Vehicle routing problems

In this section we study vehicle routing problems. These are relevant when the service can wait until the next tour, usually the next day or later, but as discussed at the end of the last section, it can also be used for problems such as the one of Example 15.2.1.

The basic situation is that of making p parallel tours to which a known number of requests have to be assigned. All travel times are known and deterministic, as well

as the service times. Many different objectives make sense, for example minimizing the time until all tours are finished. There are three types of solution methods for this problem:
- exact solutions based on integer linear optimization;
- heuristics, i.e., dedicated approximation methods;
- metaheuristics, i.e., approximation methods that can also be used for other types of problems.

Realistic applications often have a size that makes solving it to optimality using ILO infeasible or too time-consuming: as the *traveling salesman problem* (TSP) is a special case the VRP is also NP-complete (see Section 7.5). We first discuss the ILP solution, then a heuristic and finally a metaheuristic.

To formulate the integer linear program we will consider a set of locations I and a set of vehicles V. The objective is to distribute all locations over the routes of the vehicles such that: (1) each location is visited exactly once by exactly one vehicle; (2) the route of each vehicle starts and ends at the base location; (3) the total tour length is minimized. To this end we will additionally consider the set $I_0 = \{0\} \cup I$ where 0 refers to the base location. Next, consider the decision variable $x_{vij} \in \{0, 1\}$ such that $x_{vij} = 1$ if vehicle $v \in V$ travels from location $i \in I_0$ to location $j \in I_0$, $i \neq j$, and $x_{vij} = 0$ otherwise. If vehicle v travels from location i to location j it will incur a travel cost $c_{ij} \geq 0$. Furthermore, let $t_{vi} \geq 0$ be a decision variable that represents the arrival time of vehicle v at location i. The vehicle routing problem can now be solved with the following integer linear program:

$$\min \sum_{v \in V} \sum_{i,j \in I_0 : i \neq j} c_{ij} x_{vij}$$

subject to:

$$\sum_{i \in I} x_{vi0} = \sum_{j \in L} x_{v0j} = 1, \forall v \in V;$$

$$\sum_{j \in I_0 \setminus \{i\}} x_{vij} = \sum_{j \in I_0 \setminus \{i\}} x_{vji}, \ \forall v \in V, i \in L;$$

$$\sum_{v \in V} \sum_{j \in I_0 \setminus \{i\}} x_{vij} = \sum_{v \in V} \sum_{j \in I_0 \setminus \{i\}} x_{vji} = 1, \ \forall i \in I;$$

$$t_{vi} + c_{ij} \leq t_{vj} + M(1 - x_{vij}), \ \forall v \in V, i \in I_0, j \in L \setminus \{i\},$$

with $M > 0$ sufficiently large.

The first constraint imposes that all vehicles start and end at the base location. The second constraint imposes that if vehicle v arrives at location i then it also departs from location i (and the other way around). The third constraint imposes that

all locations are visited exactly once by exactly one vehicle. The final constraint eliminates any subtours (i.e., no disconnected tours).

It is not difficult to see that the number of constraints is equal to $|V|(|I|^2 + |I| + 1) + 2|L|$, and hence grows quadratically in the number of locations. Since the ILP problem is an NP-hard problem, exact solution methods for the vehicle routing problem are only viable for small-sized instances. That is, the vehicle routing problem doesn't scale well to the size of the problem. From this point of view it is preferable to use heuristic methods.

The heuristic for the VRP that we discuss in turn requires an efficient heuristic to solve the TSP. See Example 7.5.1 for such a method. We first make an initial solution, by grouping the requests on the basis of the angle they make to the base location, splitting the requests evenly over $|V|$ subsets. Using the TSP heuristics we turn these into the first $|V|$ routes. See the left image of Figure 15.2 for an illustration. An advantage of this method is that it creates convex areas that are assigned to tours, often leading to short tours. We improve from this initial solution in the following way: we change sequentially all locations from route, and exchange routes of any 2 locations (2-opt, see Section 7.5). Each time we optimize again using the TSP heuristic. We repeat this process until no more improvements are possible. Applied to the example it leads to the 2 routes on the bottom route exchanging a location, see again Figure 15.2. This method can be used for different objectives, for example to minimizing the *makespan*, the moment all vehicle are back at the base location. In the case of different base locations several heuristics exist that all try to balance the number of service requests assigned to every route and to assign requests to the service provider which base location is closest by.

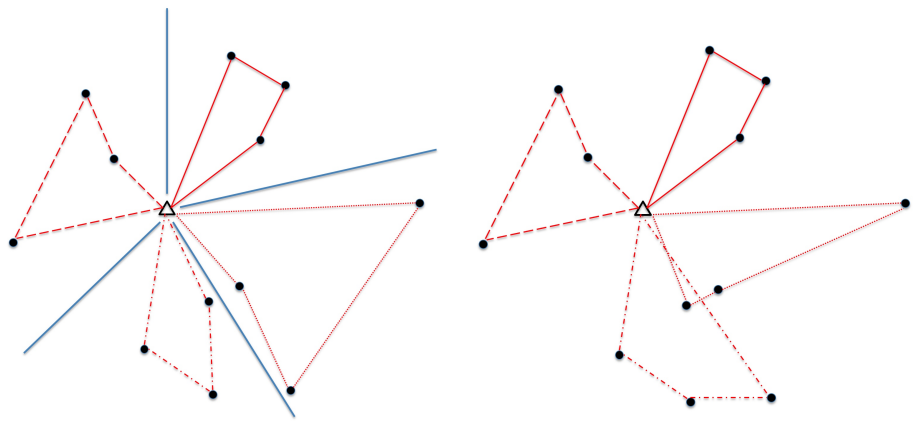


Figure 15.2: An illustration of the VRP heuristic.

One of the metaheuristics that can be used to solve the vehicle routing problem is the genetic algorithm. This algorithm is a population-based algorithm and is inspired by the process of natural selection. The general outline of a genetic algorithm is as follows: One starts with a population of individuals, where each individual is characterised by a string of symbols. The string is often referred to as the chromosome, and the symbols are often referred to as the genes. There follows an iterative procedure where two (random) individuals in the population are selected for recombination, i.e., part of the genes between chromosomes of individuals are swapped. The recombination procedure is followed by a selection procedure where "good" individuals are selected based on some fitness function, i.e., a function that evaluates the individuals on pre-defined performance measures. If the individuals improve after recombination the changes are retained, otherwise the changes are reverted. The selected individuals are used to replace the individuals from the original population, hence the population evolves from generation to generation. To prevent that the population converges to a local optimum a mutation procedure is applied at each generation to a random selection of individuals in the population, where the best individuals are often protected. The mutation procedure consists of replacing a (random) gene by another (random) gene within the chromosome of a single individual. Finally, the process of recombination, selection and mutation from generation to generation continues until some stopping criterium is reached.

The application of the genetic algorithm to the vehicle routing problem is as follows. First number the locations from 1 to $|I|$, and represent an individual in the population of solutions with a permutation of the location numbers. In this permutation we also insert $|V| - 1$ 0s in a non-adjacent manner on any point other than the endpoints. For example, the string $(4, 8, 0, 9, 6, 7, 1, 0, 2, 3, 5)$ denotes the solution of a vehicle routing problem with 3 vehicles and 9 locations, and corresponds to the tours as depicted in Figure 15.3.

A 0 demarks the start (and/or end) of a tour and may be interpreted as the base location, and the subpermutation between 0s represent the tour of one vehicle. This way each individual indeed consists of a solution to the vehicle routing problem. In order to improve the population we use a *remove-and-reinsert mutation* and a *partially-mapped crossover*. The mutation consists of removing and replacing a location within a solution of the population in a random fashion, see Figure 15.4. The crossover is applied to two solutions (called the parents) of the population and yields two new solution (the children) by swapping all locations (including 0s) of the parent solutions between two randomly selected points of the solution string, see Figure 15.5.

The genetic algorithm is then carried out as described above, with a fitness func-

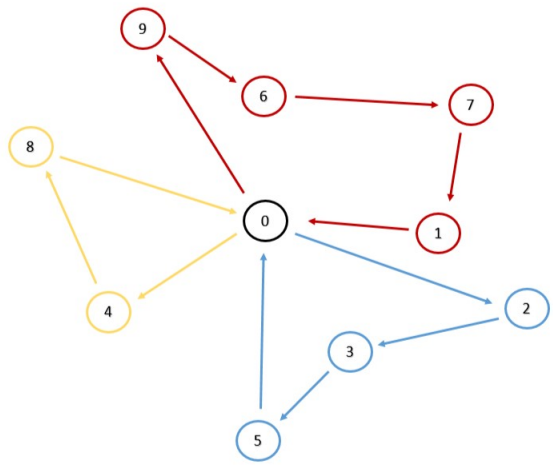


Figure 15.3: Visual representation of the solution $(4,8,0,9,6,7,1,0,2,3,5)$.

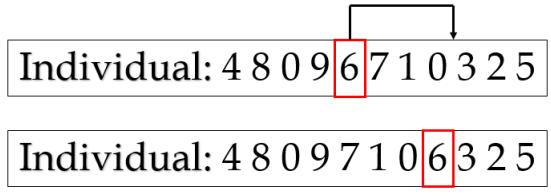


Figure 15.4: Example of mutation operator: a location is selected at random in the individuald and is re-inserted at a random point.

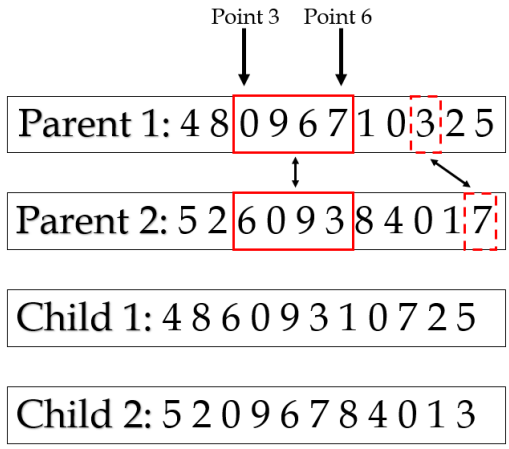


Figure 15.5: Example of crossover operator: two random points are selected, and the locations between the random points within two individuals (parents) are swapped.

tion appropriate to the vehicle routing problem (e.g., the total tour length).

The standard VRP actually has a somewhat different formulation than the one used so far: it assumes all vehicles (except one) have a finite capacity, and serving a location requires part of that capacity, as in parcel delivery. The objective is to minimize the total travel times. The heuristics presented so far can be easily modified to deal with this situation. Other main variants of the VRP are: with time windows, with backhauls, and with pickup and delivery.

In practice service and travel times are not deterministic but random. To study this stochastic version of vehicle routing we first have to reformulate the objective. Assume we try to minimize the expected time at which all tours are finished. It is important to note that both the value and the optimal solution might be different when we replace the stochastic durations by their expectations. Let $X_1, \ldots, X_V$ be the lengths of the V tours. From

$$\mathbb{E} \max\{X_1, \ldots, X_V\} \geq \max\{\mathbb{E}X_1, \ldots, \mathbb{E}X_V\} \tag{15.1}$$

(see Exercise 1.1) it follows that the objective value will be higher than was expected based on the expected lengths.

Another complicating factor of stochasticity is the fact that once certain travel and service times become known the initial solution might not be optimal anymore. This calls for rescheduling, if that is business-wise possible. (E.g., for a delivery service this won't be possible: all items to be delivered have to be put in the truck before departure.) Often these systems work with deterministic estimates of travel times

which are changed based on real-time information. Navigation systems for consumer use typically work this way.

Finding (nearly) optimal routes taking the variability into account is a challenging task on which little scientific work has been done. Simulation optimization or heuristics based on normally distributed route lengths seem to be appropriate.

Introducing randomness is a logical and practically useful extension: not only travel times are prone to partly unpredictable variations, also the places to visit, the demand, might change over time. Typical examples of the latter are field service and home care, where new customer service requests or emergency patients might change the routes during the day. Less work has been done on these stochastic routing problems. We will first classify the different problems and then go into more detail for each of them.

15.4 Scheduling appointments

Up to now we considered systems where the customer location was visited as soon as possible (in location problems) or during the next tour (the standard way of working for the VRP). However, there are advantages of scheduling the service another moment:
- fluctuations in demand can be smoothed;
- tours can be shorter by clustering demand;
- visits can be scheduled at convenient moments for customers.

This leads to systems where the planner decides on a time window with the customer (e.g., next Wednesday between 8 a.m. and noon). The time window gives planners the flexibility to reschedule jobs when new ones arrive and it allows for randomness in the execution.

The assignment of jobs to tours and the order within tours is based on the VRP discussed in the previous section. However, when working with appointments an additional scheduling problem emerges: which time window to propose to a customer at the moment of the service request? Heuristics exists to solve this problem, based on the idea that customers in a tour should be clustered by location as much as possible. This amounts to algorithms that propose time windows in which close-by locations are already present. In what follows we present a possible heuristic.

We assume that all travel and service times are deterministic and known. We consider a given number of routes to plan: typically the planning horizon (say, 5 days) times the number of technicians. Initially, all routes are empty, and for each technician the base location is known. Now, for every customer request that arrives,

two things have to be done: the new customer has to be assigned to a route, and a time window for the appointment has to be determined.

To find the best route, the new location is added to all routes. The route for which the additional travel distance is lowest is chosen. To avoid that some routes fill up quickly and others remain empty, which is certainly a possibility when the base locations are the same, we can add a penalty for routes having little idle time left. Next, the time window has to be chosen. Evidently, this time window is between the time windows of the previous and next locations in routes, and it is chosen proportionally to the travel distances to these locations.

15.5 Capacity planning

Up to now we discussed operational scheduling decisions: how to assign tasks to the available capacity. Important long-term decisions concern the design of the system and the amount of capacity required. A typical design question is whether or not to split the service region in separate independent entities. Smaller independent regions can have managerial advantages; from a planning perspective a bigger scale is always better. Both models do not necessarily conflict: management can be done at a smaller scale than planning.

Let us consider capacity decisions. As in most service systems, capacity should exceed demand to make sure that under demand and capacity fluctuations the required service levels can still be met. A special feature of the systems considered in this chapter is the fact that travel times should be counted as well. Tight requirements on service levels have a double effect on the required overcapacity: it should be higher because of the demand fluctuations, but also because travel times are longer, because there is less room for route optimization.

These issues play also a role in specialization. In service organizations with highly specialized maintenance we see a trend towards specialization. The price to pay for this is less scale advantages in the sense that one group of technicians cannot take over another group's peak service requests. It also leads to longer travel times. This calls for *multi-skilled* technicians. We will deal in more detail with these types of issues in Chapter 17 on call centers.

Let us focus on those routing problems where technicians (mainly) work fixed scheduling during business hours. For due date service systems we can assume continuous operation by leaving out the non-business hours. Assuming (almost) homogeneous Poisson arrivals, an $M/M/s$ queue can be used to give a rough estimate of the performance.

Example 15.5.1 A service organization makes its technician start from their homes (spare parts are delivered overnight). The 8 hour working day starts at the first customer, thus the travel time to the first customer does not count as working time. Travel times between customers and back home are 45 minutes on average, service takes 60 minutes. On average $8/1.45 \approx 4.5$ customers can be visited each day.

The required service level is 1 business days, in 95% of the cases, and there are on average 50 service requests per day. Now we enter the data in the Erlang C/$M|M|s$ calculator at the www.gerkoole.com/OBP, taking the arrival rate equal to $50/8 = 6.25$ per hour, an average service time of 1.75 hours, and we take an acceptable waiting of 7 hours: 1 business minus the travel time, taking 15 minutes into account for travel time variability. This leads to a required number of technicians of 12, given an occupation rate of $6.25 \times 1.75/12 \approx 91\%$.

The outcome of the queueing formula s does not take into account training time, holidays, illness, etc. Thus extra technicians should be hired for that. Assume that the fraction of unproductive time is u. Then the total number of required technicians is $s/(1-u)$. In call centers the number u is called *shrinkage*.

The calculation above is, for many reason, an approximation. The $M/M/s$ allows for few generalization, thus more data is available on the process, the service location or on the distributions of durations, then the only appropriate solution method is simulation. This is also the appropriate method for analyzing other systems such as the ones with time slots.

Scheduling of ambulance crews

Ambulances workforce scheduling is different because the arrival rate of service requests is highly non-homogeneous and the time to answer is so short that the number of ambulances should follow the demand pattern. This creates a new problem: the scheduling of ambulance crews. This form of scheduling resembles very much the scheduling of call center *agents*, for this reason we refer to Chapter 17 for staff schedules in ambulance services.

15.6 Car stock management

A problem different from tour assignment and routing problems is that of determining the car stock of a technician. This is an inventory problem (see Chapter 6). The car stock enables the technician to repair equipment by replacing parts without having to go back to the base location. Going back indirectly costs money: by dividing the total operational costs by the number of visits a price per visit v can be computed. A

'lost sale' therefore approximately costs v. On the other hand, there are holding costs for having items in the car stock, and the amount of space is limited.

The replenishment policy can differ, but often the stock is replenished overnight or in the morning when the technician visits the spare parts warehouse. When we do not take the delivery costs into account then the usage is replenished directly every day. In that case the days do not depend on each other and it suffices, for each possible item, to solve a single-day problem, with the following parameters:
- costs per lost sale v;
- holding costs c, typically equal to the price of the product times 0.1, the annual holding costs per unit price, divided by 200, the number of working days;
- demand D per day;
- car stock size S.

For these parameters the total expected daily costs are given by:

$$C(S) = Sc + q\mathbb{E}(D-S)^+ = c(\mathbb{E}D + \mathbb{E}(S-D)^+ - \mathbb{E}(D-S)^+) + v\mathbb{E}(D-S)^+ =$$

$$c\mathbb{E}D + c\mathbb{E}(S-D)^+ + (v-c)\mathbb{E}(D-S)^+.$$

Because $c\mathbb{E}D$ is a constant independent of S, the car stock problem fits the conditions of the newsvendor, and thus the optimal S can be found using Theorem 6.2.1.

Example 15.6.1 An item with a value of \$200 is needed on average once every 100 machines. A return visit is estimated to costs \$100. Based on the value of the item Because the daily holding costs are \$0.10, according to the newsvendor model we should take stock according to the 99.9 percentile of the demand distribution. This makes sense: a return visit is much more expensive then keeping stock. If a technician visits 4 customers per day, then it is optimal to take 1 item in the car stock.

The model that includes replenishment costs is much harder to analyze, especially since a delivery consists of items of multiple types. Another complicating factor is the fact that it might occur that the rational car stock does not fit into the car anymore.

Modern electronic equipment make the car stock problem less relevant. Electronic self-diagnosis of the equipment transmits the components to be replaced automatically to the supplier, who can then these components after the next supply of the technicians. It also allows for automatic condition monitoring (see page 220), making preventive maintenance more efficient.

15.7 Further reading

The MCLP was introduced in Church & ReVelle [38]. For more location covering problems see Daskin [49]. Recently there has been much scientific attention to the scheduling of ambulances. See for example Jagtenberg et al. [80] and references therein.

Lawler et al. [104] is entirely devoted to the TSP, with a chapter on vehicle routing. Toth & Vigo [154] is the standard reference for vehicle routing. Several chapters in OR/MS Handbook 8 [17] deal, in detail, with different aspects of vehicle routing. Gendreau et al. [63] discusses different versions of stochastic vehicle routing problems.

The time slot model of Madsen et al. [108] is the basis of Section 15.4. A more recent article on car stock management is Bijvank et al. [24].

15.8 Exercises

Exercise 15.1 Solve the problem of Figure 15.1 by implementing and solving the ILO formulation.

Exercise 15.2 Consider a VRP with 2 tours, 4 service requests, 0 travel times and stochastic service times. Find service times such that the two objectives in (15.1) give different optimal tours.

Exercise 15.3 Consider a technician who needs a certain spare part on average once a year. A customer visit costs on average 40 euro. The part cost 100 euro. Is it economically rational to put the part in the car stock?

Exercise 15.4 A technician visits every day 4 customer locations. At each location there is a probability of 0.01 that a certain part is needed. Replenishments occur daily.
a. How many parts should the technician have in his car stock to have a probability lower than 0.01 per visit that he cannot replace the item when needed?
b. The item costs 50 euro. What is a rational car stock?

Exercise 15.5 A service system has 20 technicians. Average travel times between customer locations is 25 minutes, service times are on average 40 minutes. Technicians travel to the first customer and from the last customer in their own time; the average working day is 8 hours.

a. How many service requests can this system handle on average daily?
b. Describe a methodology for obtaining approximations for the time a customer must wait on average.
c. The technicians prefer to work 10 hours during 4 days instead of 8 hours during 5 days. What do you think of this proposal?

Exercise 15.6 Verify the computation made in Exercise 15.6.1. Repeat it when the item is required for 10% of the machines.

Chapter 16

Health Care

In developed countries the health care sector may account for up to 20% of the gross national product, and its share is still increasing. This has put cost reductions in health care expenses high on the political agenda in many countries. The idea that much can be gained without reducing the quality of care is shared by many, but seems hard to realize.

Although many general operations management principles apply, there are a number of aspects that make health care, from a planning and scheduling point of view, different from manufacturing. The way health care is financed and the fact that health care is a service are among the most important ones.

In this chapter we discuss those planning and scheduling problems in health care that occur most often. The focus will be on hospitals, but we will cover the whole sector.

16.1 Introduction

In manufacturing the goal is to make as much profit as possible. A company makes products that customers buy. As long as the total production costs are lower than the revenue from sales the company makes profit. Health care is fundamentally different. Although commercial aspects play an increasingly important role, one of the aspects that is essential is that governments finance the health sector to a large extent in order to provide all citizens with a certain level of health care. For this reason health care has three conflicting objectives: price, quality and accessibility. The latter is new in comparison to manufacturing, and relates to the fact that all citizins should have access to high-quality care. Unfortunately, the way in which the sector is financed

not always stimulates health care institutions to behave according to these general objectives.

Quality of health care services falls apart in two aspects: those that represent the medical outcome and those that are related to the process such as the length of waiting times. Note that both types of quality are related: think about a cardiovascular patient who has to wait long before being some intervention. Evidently this is process-related, and leads to dissatisfied patients. But there is also statistical evidence that waiting leads to worse medical outcomes and higher mortality. The same holds for example *nurse-to-patient ratios* at nursing wards: less nurses per ward leads to longer waiting and worse medical outcomes, thus the process and the medical outcomes are strongly related. In this chapter we mainly focus on the process-related definition of quality, in relation to efficiency of the health-care delivery process. Of course, efficiency is strongly related to costs. Thus, we focus on the quality-efficiency trade-off, as we did for manufacturing. However, the fact that health care is a *service* makes it crucially different from manufacturing.

Health care is a service, which means that the customer or patient is part of the process. Thus, if we see health care delivery as a production process, "inventory" equals patients waiting. Consequently, even more than in manufacturing, there is a need to reduce inventory. However, this puts pressure on the productivity of the process, because inventory serves as lubricant in production chains, unless the variance in the process steps is very low. Radical solutions to avoid waiting, such as the combination of process steps (e.g., creating a "one-stop shop", combining multiple appointments in one visit), call for completely different planning approaches.

Another aspect of the human factor is that variability cannot be reduced to the extent as it can in manufacturing. People are different in all aspects of their behavior: some do not show up for an appointment and others come early, they require a different length of stay in the hospital to recover from an operation, and so forth. Fluctuations can be reduced and predicted to a certain extent (for example, by sending a reminder for an appointment, or by statistically differentiating between patients), but there will always remain fluctuations and uncertainty, more than in manufacturing. Planning in this situation is a major challenge for the health care sector.

Not related to the service aspect, but also different to manufacturing, is the organizational structure of most health care institutions. Other than in most production facilities there is a flat organizational structure, known as a *professional bureaucracy* (Mintzberg [115]). It is *professional* because the expertise of the hospital professionals is prevalent in the decision process; it is a *bureaucracy* because it is organized according to fixed rules and positions. To give an example of the latter, the process of

becoming a doctor is completely regulated. This structure makes it difficult to implement management decisions that encompass different organizational units, and improvements that focus on an efficient use of resources usually do require a broad focus. This does not facilitate changes in the process.

In the last decades health care expenses exploded, although the increase seems to slow down the last couple of years. It is commonly believed that the ageing population is the main cause, but technological advances are at least as important. This drove governments in developed countries to focus on cost reductions and the introduction of a regulated market. Formerly this was less of an issue. Because of the lack of competition and the availability of sufficient government funding the health care sector was, logistically speaking, one or two decades behind of manufacturing. Due to the increased focus on efficiency and quality many concepts that had been proven useful in manufacturing have recently been adopted to health care. We discuss them when relevant. The remainder of this chapter is focused on mathematical modeling of health care processes.

16.2 Overview

In this section we give an overview of the whole health care system. From this we derive a number of areas in which typical planning problems play a role. Every country has its own way of classifying the different forms of care, and also the paths that patients follow through the system might be different: for example, in some countries general practitioners (GPs) serve as gatekeepers for more specialized care, in other systems patients can go directly to specialized care. But no matter how the system is organized or financed, we can make a difference between *cure*, often short-term interventions aimed at curing patients of some illness, and (long-term) *care*, aimed at caring for people without the objective of curing from some disease, as is done in nursing homes. Some care and cure takes place at the patient's home, *home care*, but most of the health care activities take place in specialized institutions, of which hospitals are the most complicated ones. Other institutions include practices for GPs, practices for physical therapy, etc. *Mental health care* has its own institutions, but from a planning perspective it has much in common with other practices and hospitals. Important health care providers not yet mentioned are ambulance service providers and institutions active in prevention.

We introduce some useful terminology and concepts. The first distinction is between *scheduled patients* and *unscheduled patients*. Scheduled or *elective* patients are scheduled through an appointment system, unscheduled patients can be assumed to

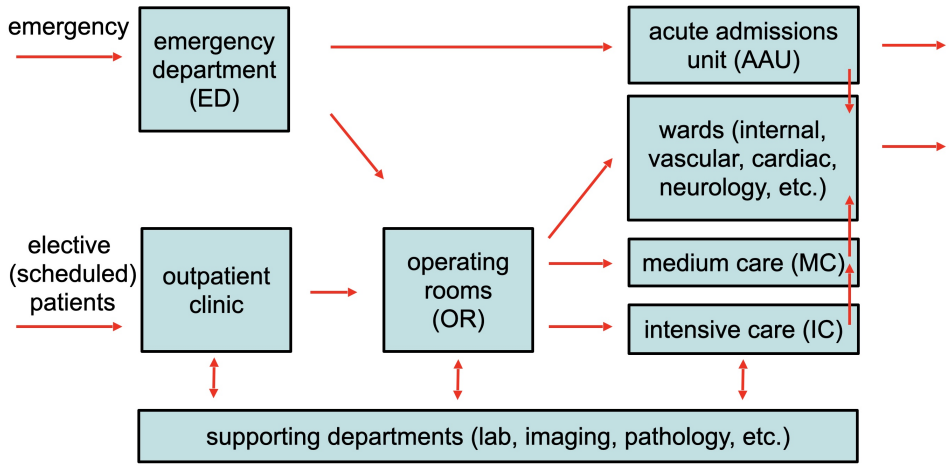


Figure 16.1: Typical entities of a hospital including main patient flows

arrive according to a Poisson process, with an arrival rate that depends on the time of day (as discussed in Section 2.4). Unscheduled patients fall apart in two groups: *urgent* patients and *emergency* patients. The difference between the two is that, for example, urgent patients can wait for the next day to be operated, while emergency patients have to be operated in say 8 hours.

An elective patient has two types of waiting times: the *access time*, which is the time between the moment the appointment is made and the actual appointment, and the *waiting room time*, which is the time the patient spends waiting in the waiting room at the health care facility. Another distinction is between *clinical* patients and *ambulatory* patients, those that stay in a hospital, and those that visit during the day a health care institution. Ambulatory patients are also called *out-patients*, and clinical patients *in-patients*.

In Figure 16.1 we show the main components of a hospital with the main patient flows in it. The full flow of a group of patients is called a *clinical pathway*. A typical pathway for a *trauma patient*, e.g., after a serious (traffic) accident, is: ambulance - emergency department - operating room (OR) - intensive care - medium care - regular ward - discharge to (nursing) home. Another pathway for emergency patients not involving surgical intervention could be: ED - acute admissions unit - internal medicine ward - discharge. A typical elective pathway would involve several visits to the outpatient clinic, including use of the lab for blood tests and the imaging department for X-rays or other scans, the OR and then a specialized ward. This would typically be the case for orthopaedic interventions such as knee operations.

As such a hospital, or even the whole health care system, can be seen as a complicated job shop with many planning problems at the different components of the system and at the level of the clinical pathways. We will discuss first capacity planning and scheduling of wards, relevant to all wards at the right of Figure 16.1. Then we move to capacity decisions for outpatient clinics, which is also relevant for other forms of capacity such as imaging. The focus will be on appointments, but the case of *walk-ins* will also be considered. Next we discuss *appointment scheduling*, which is the questions at what times the appointments should be scheduled. Both sections are also relevant for other health care institutions, such as GP posts, mental health care clinics, physical therapists, etc. Then we move to operating room planning. We highlight the differences outpatient capacity planning and discuss the main planning principles. We finish with a section on clinical pathways, i.e., issues involving multiple types of resources at the same time.

With that we cover most of the health care sector, with home care and ambulance services as most notable exceptions, but they are already covered in Chapter 15 on distribution and field service.

16.3 Bed planning

This section consists of three parts. First we look at capacity models for nursing wards. The main conclusion will be that the Erlang B model is appropriate in many cases. Then we study properties of the Erlang B system, especially related to scale. In general, scale is better from a planning perspective, but it requires nurses to be multi-skilled. That will bring us to the third part: how can we increase scale without requiring full flexibility from the nurses?

Let us consider the size of clinical wards. Every ward contains a number of beds. Note that only a number of the physical beds might be available due to personnel issues. The beds that can be used because personnel is available are called *operational beds*. Capacity decisions and, eventually, admission decisions at wards are complicated by the fact that the time that patients spend on wards for medical reasons are highly variable. This time is called the *length of stay* (LOS), its average is denoted by ALOS. Arrival times are also random, up to a certain degree. Emergency patients can be assumed to arrive according to non-homogeneous Poisson processes with a daily cycle. Usually the ALOS is considerably longer than 1 day, making the fluctuating arrival rate of limited relevance for capacity decisions. For elective patients the moment that patients arrive at a ward can often be planned. The most common example is a patient who arrives at a ward the day before operation and enters a (possibly

different) ward right after the operation. One could expect that the resulting arrivals are very regular, for example 1 patient per day. However, data analysis shows that the variation of the number of admissions per day on many wards is comparable to that of a Poisson distribution, making the Poisson process also a reasonable approximation for the case of elective patients. The randomness in both arrival moments and lengths of stay implies that either the bed capacity at wards cannot be fully used, or that there is a high blocking probability. The latter situation occurs regularly in hospitals: too often scheduled operations have to be cancelled because of lack of capacity at clinical wards. Next to that it often happens that a patient is not admitted at the right ward, in the "wrong bed". The reason is that hospital professionals, responsable for capacity decisions, do not recognize the impact of the variation: they have the tendency to take the number of beds s equal to the average numbers of arrivals λ times the ALOS β. This is an example of the *Flaw of Averages* (see Section 1.2). In other situations they account for some slack capacity, but this percentage is usually not related to the size of the ward or the number of refused or delayed admissions. For example, the number of beds is considered to be right if the occupancy is close to 85%.

A reasonable model for helping to make ward capacity decisions is the Erlang B model. See Theorem 5.4.3 for the calculation of performance measures, and Exercise 5.8 for the method to build your own calculator. You can also use the calculator at www.gerkoole.com/OBP. Note that this model assumes Poisson arrivals but does allow for general service time distributions. Erlang B also assumes that arrivals that find all beds occupied leave the system, and are possibly redirected elsewhere. See Table 16.1 for some numerical examples. Note that the rejection probability is equal to $\pi(s)$, the fraction of time that all s beds are occupied. The average number of occupied beds $\mathbb{E}L$ is equal to $\lambda(1-\pi(s))\beta$ (cf. Equation (5.7)). In the Erlang B model it is customary to write $\pi(s) = B(s,a)$, with $a = \lambda\beta$. Note what happens when we double both the size of the ward and λ: the rejection probability $\pi(s)$ decreases and the number of occupied beds $\mathbb{E}L$ more than doubles. These are the economies of scale of the Erlang B model.

s	10	10	10	10	10	20	20	20	20	20
λ	2	4	5	6	8	4	8	10	12	16
$\pi(s)$	0.005	0.12	0.21	0.30	0.44	0.0002	0.06	0.16	0.26	0.41
$\mathbb{E}L$	4.0	7.0	7.9	8.4	9.0	8.0	15.0	16.8	17.8	18.8

Table 16.1: Erlang B examples for $\beta = 2$ (all units in days)

A complicating factor in practice, when applying the Erlang B formula to ward dimensioning, is that usually only admissions are counted. Refused admissions are often not registered and therefore we do not have a direct view of the actual demand. Let us define λ_e as the expected number of admissions per unit of time: the *effective* arrival rate. Usually we measure λ_e, β and s. From this λ can be computed, but there is no closed-form expression known. Using the fact that λ_e is increasing in λ we can find a simple numerical procedure to obtain λ. Another method to find λ is measuring $\pi(s)$. Using the PASTA property (see Section 3.6) we find that $\pi(s)$ is equal to the probability that an arbitrary arrivals is refused admission, and thus $\lambda = \lambda_e/(1-\pi(s))$.

To calculate $\pi(s)$ we need the actual arrival and departure times to calculate the number of occupied beds as a function of time. Note that we assumed s to be fixed. In reality s, the number of operational beds, fluctuates to a certain extent. This can be on purpose, for example in order to anticipate on a lower load during weekends, or unpurposely, because of illness of nurses.

Example 16.3.1 For a ward with 5 beds we analyzed the average numbers of arrivals as a function of the number of occupied beds. Data was collected for a year on a half-hour basis. The average number of arrivals for $s = 0, \ldots, 7$ was as follows: 0.24, 0.21, 0.22, 0.17, 0.11, 0.05, 0.01, 0.00. These data show that sometimes there are as little as 3 operational beds, and that on the other hand there are sometimes as much as 7 beds occupied.

The Erlang B model gives us a tool to rationalize decisions concerning ward sizes. At the same time, this model can give us a lot of insight concerning decisions related to merging of wards or *bed pooling*, the idea that dynamically, on the basis of occupation, wards can share beds. We already saw in Table 16.1 an example of economies of scale: when a ward is doubled in both size and demand then the occupation goes up and, equivalently, the rejection probability goes down. Note that doubling the scale is equivalent to merging two identical wards. Using Theorem 5.4.5 we can see that this property holds in general: if wards with identical ALOS ($\beta_1 = \beta_2$) are merged then, due to Equation (5.8), the overall expected number of occupied beds goes up. If we divide (5.8) by $\beta = \beta_1 = \beta_2$, then we get:

$$(\lambda_1 + \lambda_2)B\big(s_1 + s_2, (\lambda_1 + \lambda_2)\beta\big) \leq \lambda_1 B\big(s_1, \lambda_1\beta\big) + \lambda_2 B\big(s_2, \lambda_2\beta\big). \tag{16.1}$$

Because $\lambda B(s, a)$ is the expected number of rejected patients per unit of time, we also see that the total number of refused patients decreases. This does not mean that the blocking probabilities of both patient flows go down; one might increase, but the weighted average decreases.

Example 16.3.2 Consider two intensive care units (ICUs) with equal ALOS: one with a load of 20 and also 20 beds, and a specialized ICU with a load of 8 and 12 beds. Assuming Poisson arrivals the Erlang blocking model can be used to determine the blocking probabilities: 0.16 and 0.05, with a weighted average of 0.13. A single ICU with load 28 and 32 beds has a blocking probability of less than 0.07: half the value for the separate ICU! However, the blocking probability for the specialized ICU increased.

Up to now we discussed merging wards with the same ALOS. Usually wards with different types of patients have a different ALOS. Equation (5.8) applies also in this situation: merging wards leads to a higher overall occupancy. If a hospital is being paid by the days that beds are occupied, then the revenue is maximized by merging as many wards as possible. On the other hand, Equation (16.1) does not hold anymore, because $\beta_1 \neq \beta_2$. Table 16.2 gives a counterexample. In column 3 we see the weighted averages over the two wards: $(\lambda_1 B(s_1, a_1) + \lambda_2 B(s_2, a_2))/(\lambda_1 + \lambda_2)$ is the weighted average rejection percentage, $(\mathbb{E}L_1 + \mathbb{E}L_2)/(s_1 + s_2)$ is the overall occupancy (which is also the weighted average occupancy, weighted with respect to the number of beds). Column 4 are the numbers for the merged ward. We see that not only the occupancy but also the rejection percentages increase.

	ward 1	ward 2	weighted average	merged ward
arrival rate (λ)	1.00	5.00		6.00
ALOS (β)	5.00	1.00		1.67
# of beds (s)	3	7		10
rejection % ($B(s,a)$)	53%	12%	19%	21%
occupancy % ($\mathbb{E}L/s$)	78%	63%	67%	79%

Table 16.2: Example of merging wards with $\beta_1 \neq \beta_2$

The counterexample is not very realistic, with a 53% rejection probability in the first ward. In most practically relevant cases merging leads to a decrease of the overall rejection rate. The fact that the merged wards do not necessarily profit equally from merging, or even have an increasing rejection rate, needs more attention.

We saw that merging wards leads to an overall increase in occupation and usually also to an overall decrease in refused admissions. However, it can happen that the blocking percentage of one of the types of patients increases. In most cases this is undesirable, as the number or percentage of refused admissions per type of patient is one of the key performance indicators of hospitals. Usually it is formulated as a constraint: it should remain below a certain specified percentage.

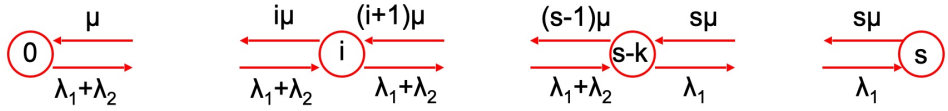


Figure 16.2: Transition diagram of the birth-death process of a threshold policy

Example 16.3.3 An intensive care unit admits emergency patients from the Emergency Department and patients that underwent scheduled surgery. Emergency patients can be sent to another hospital, and scheduled operations can be cancelled. This is allowed to happen only for 1% of the emergency patients and 5% of the patients that have to undergo a scheduled operation.

The challenge is to find a way to assign beds of a ward to different types of patients having different parameters, in order to profit from the economies of scale and to satisfy the constraints on the rejection percentages. The way to do this is *dynamic bed allocation*: the idea that beds should be assigned to patients on the basis of current occupancy information. A possible consequence is that a patient can be refused while there are still beds free at the ward.

In this section we consider two types of dynamic bed assignment policies: one that uses *threshold policies*, often studied in the context of admission control to queueing systems, and one that uses *earmarked* beds, an idea coming from the health care domain. With threshold policies a patient of a certain type is only admitted if the total number of beds occupied does not exceed a certain type-dependent number, the threshold. When earmarking is used then a certain number of each type of patients is guaranteed a place, and when less places are occupied, then beds are kept free for these patients.

Let us go into more detail about threshold policies. We consider the case of two types of patients and (almost) equal ALOS. The occupation can be modeled as a birth-death process with states $\{0, \ldots, s\}$ and departure rates $\lambda(x, x-1) = x\mu$ for $0 < x \leq s$, with μ as usual equal to the reciprocal of the ALOS. For a threshold policy the arrival rates are as follows: $\lambda(x, x+1) = \lambda_1 + \lambda_2$ for $0 \leq x < s'$ and $\lambda(x, x+1) = \lambda_1$ for $s' \leq x < s$, with $k = s - s'$ the number of beds that we try to reserve for type-1 patients. See Figure 16.2.

It can be proven that such a threshold policy minimizes a weighted sum of the rejection rates, assuming that the costs for rejecting type-1 patients is higher than rejecting type-2 patients. The optimal threshold level can be determined using a technique called *dynamic programming* or simply by comparing the costs for different threshold levels using the birth-death formulation. In practice, there are usually

constraints concerning the percentages of refused admissions. A comparison for different threshold levels is then required to find the best solution. Outcomes of such an analysis can be found in Table 16.3.

scenario	# of beds	rejections type 1	rejections type 2	overall occupancy
separate wards	{5,10}	11.0%	16.8%	67.7%
no threshold	15	8.6%	8.6%	73.1%
threshold $= 1$	15	2.3%	13.7%	71.3%
threshold $= 2$	15	0.6%	18.7%	68.6%

Table 16.3: Threshold policies with $\lambda_1 = 1$, $\lambda_2 = 3$ and $\beta_1 = \beta_2 = 3$

The threshold policy can be easily generalized to more than 2 patient types, every type having its own threshold level. Determining the right mix of threshold levels does become a little more involved. More challenging is the case when the ALOS of the different patient types are very different. In that case a one-dimensional birth-death process does not suffice to model the system: a separate state variable is needed for every patient type. With two types, the state is thus of the form (x_1, x_2), with x_i the number of patients of type i, and $\mathcal{X} = \{(x_1, x_2) | x_i \in \mathbb{N}_0, x_1 + x_2 \leq s\}$. The optimal admission policy is a function of the state and can be rather complicated. Simple threshold policies which depend only on $x_1 + x_2$ perform quite well and are much easier to implement.

Threshold policies are well suited for nursing wards with a single specialty and where no different skills are needed for different types of patients. In a ward shared by several specialties and different types of patients threshold policies have the disadvantages that:
- there is no upper bound (other than s' or even s) on the number of beds that can be occupied by a certain type of patients making that all nurses should be able to handle all patients, even if they are assigned to a subset of the beds;
- there is no lower bound on the number of beds available for a specialty, making it, for example, difficult to plan operations in advance that have the ward as next process step.

To avoid both disadvantages we can assign beds of the shared ward in the following way: there are number s_1 and s_2 with $s_1 + s_2 \leq s$ that are the beds reserved for type 1 and 2, respectively. The number of shared beds is $s - s_1 - s_2$. The admission policy is as follows: in state (x_1, x_2) with $x_1 + x_2 < s$ type 1 is admitted if $x_1 < s - s_2$, and vice versa for type 2. See Figure 16.3. For a numerical example, computed using a 2-dimensional Markov chain, see Table 16.4.

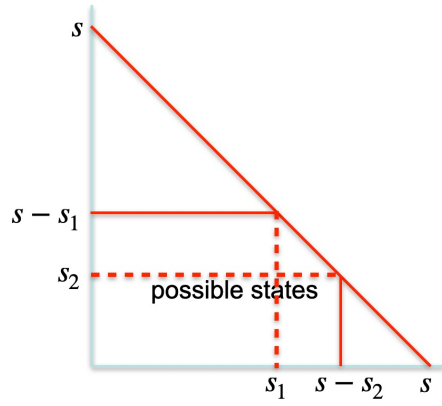


Figure 16.3: State space of the Markov process of earmarking

s	s_1	s_2	rejections type 1	rejections type 2	overall occupancy
15	5	10	11.0%	16.8%	67.7%
15	3	9	9.2%	10.2%	72.0%
15	1	4	8.5%	8.7%	73.1%
15	0	0	8.6%	8.6%	73.1%
15	5	7	3.5%	17.2%	68.9%

Table 16.4: Earmarking policy with $\lambda_1 = 1$, $\lambda_2 = 3$ and $\beta_1 = \beta_2 = 3$

We see that a high occupancy can be reached without too many flexible nurses. For example in the second scenario there are 12 specialized nurses out of 15 and the occupancy is only 1% lower than with only flexible nurses. Earmarking is less appropriate for protecting type 1, we need quite a number of earmarked beds to get the rejection % for type 1 below 5% which has its repercussions on the occupancy, as is shown in the last scenario. Note that for the computations we had to analyze a more-dimensional Markov chain, even though the ALOS are equal, because the policy and thus the transition rates depend on the numbers of beds occupied by different specialties.

16.4 Capacity decisions for appointment systems

Elective care patients have to wait twice before seeing a doctor: the time between the moment the appointment is made and the time of appointment (usually measured in days or even months), and the waiting room time (measured in minutes). The

waiting room time will be discussed in the next section. Here we discuss the time until appointment or, depending on the way in which the appointment system works, the time patients spend on a waiting list. We will call it the *access time*.

An appointment with a doctor is the main example, but the method we will develop in this section can be applied to appointments with many other sorts of health care professionals, tests such as imaging (X-rays, MRIs, CT-scans, etc.), surgical interventions, etc. For simplicity we will use the example of a doctor's appointment throughout the text.

It is obvious to use queueing models for modeling the access time. It would make sense to use a discrete-time model, with time measured in slots, but because of the availability of formulas for continuous-time models it is wiser to a continuous model as approximation. Let s be the number of *slots* for appointments on a day. Ideally, we would like to use an $M|D|s$ model with $\mathbb{E}S = 1$ and λ the number of arrivals per day. Because no formula exists for this model we can either approximate the performance using an $M|M|s$ queue or an $M|D|1$ queue with the service time equal to 1 over the number of slots.

Example 16.4.1 A doctor see on average 10 patients per day and has every day 12 slots available for seeing patients. According to the Pollaczek-Khintchine formula (5.4) with $\lambda = 10$, $\mathbb{E}S = \beta = 10/12$, $\sigma(S) = 0$, the expected waiting time $\mathbb{E}W_Q = 0.21$ days, which can be verified using the online calculator on www.gerkoole.com/OBP.

According to the Erlang C approximation of Theorem 5.4.1 with $\lambda = 10$, $\mathbb{E}S = \beta = 1$ and $s = 12$ the expected waiting time is $\mathbb{E}W_Q = 0.22$ days.

Note that λ needs to be determined, for example using the techniques discussed in Section 2.5. The forecast can be varying along the season, it might be desirable to compute the model for each month separately using different values of λ. Evidently, the models predict both waiting times and unused slots. However, this is not always observed in practice: we often see capacity used at 100% and a stable access time of most often several weeks or months. This is in contradiction with standard queueing models such as the $M|M|s$ and $M|D|1$: either the system is stable and there are free slots now and then and therefore 0 waiting time, or demand is too high and the waiting time grows towards ∞.

There are three main reasons why this can happen. The first is that in health care many capacity decisions are taken in an ad-hoc manner, for example appointments are often scheduled beyond the planned capacity, leading to long waiting-room times but keeping access times short. Thus the capacity is adapted to the demand.

The second reason is that the demand is often not independent of the access time. In many situation patients have a choice of facility, and short access times will lead to

an increase in demand. Thus it might well be the case that an increase in capacity has little impact on the access time: the arrival rate is a decreasing function of the queue length, and the probability mass of the resulting birth-death process is concentrated around the point where arrival and departure rates are close to each other. Determining the rate as a function of the state is difficult because of lack of reliable data. But the effect is to be taken into account when decisions about capacity are taken.

Advanced access

In the beginning of this century many health care institutions tried to reduce waiting times using lean principles (see the box on page 193). Activities that did not add value, such as repeat visits without complaints, were cancelled and this way capacity was increased. Then an additional effort was made to work away the backlog, and from that moment on the system would be stable and very short access times could be guaranteed. However, many of the *Advanced Access* projects failed because demand grew after its introduction and access times increased slowly to their original level.

Sometimes it is possible to avoid the growth of demand even when access times are short. This is already the case when it does not concern the primary specialty: few people choose an orthopaedic clinic because the X-rays are done quickly. When it concerns access to the primary specialty then the access time can be managed by limiting the patient pool. This is possible in specializations where the doctors keep a fixed group of patients, the *panel*, which visit them regularly. Examples are gynaecologists, general practitioners and doctors who treat patients with incurable diseases such as diabetes. For example, a general practitioner in the Netherlands usually has 2000 patients. Each of these patients has a small probability of making an appointment. The resulting arrival process is close to a Poisson process (see Section 2.1). The panel size can now be chosen in such a way as to avoid long waiting times.

The third reason is that patients might leave the waiting list, for several reasons. For example, the patient found what she searched for somewhere else, the need disappeared, or, on the contrary, the patient's condition became worse and the patient therefore fell in a different category and is perhaps being treated as an emergency. We can also model this situation. We start from the regular $M|M|s$ model that we extend by adding the possibility of patients abandoning the queue. This model has been extensively studied in the context of call centers, but less for health care. Define the time until patients leave the queue (due to whatever cause) supposing they are not being served as Y. Estimating the distribution or moments of Y can be done using the Kaplan-Meier estimator, see Section 1.8. Take $\gamma = 1/\mathbb{E}Y$, the rate at which a waiting patient leaves the queue, assuming exponentiality. Now we can

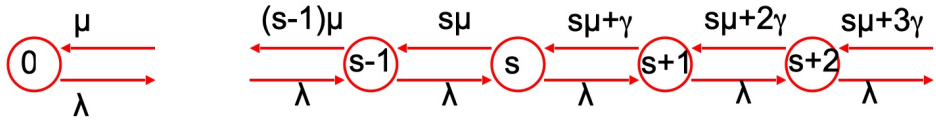


Figure 16.4: Transition diagram of the birth-death process of a system with abandonments

model the system as a birth-death process with rates $\lambda(x, x+1) = \lambda$ for $x \geq 0$ and $\lambda(x, x-1) = \mu + (x-1)^+\gamma$ for $x > 0$. See Figure 16.4. This system is known as the *Erlang A*, and sometimes written as $M|M|s+M$. This queue is always stable, because the departure rate is increasing and eventually gets higher that the arrival rate. Numerical experiments show what we observe in practice: in overload ($\lambda > \mu$) this queue hardly empties and the stationary distribution is concentrated around the state x with $\lambda \approx s\mu + x\gamma$.

Example 16.4.2 An intervention is done 5 times a day, but the demand is 6 per day on average. On average every day 1% of the waiting patients abandons. An analysis of the birth-death process, or the use of the online calculator at www.gerkoole.com/OBP, reveals that close to 1/6th of all patients abandon, and patients spend on average 18 days on the waiting list.

So far we have talked about models for determining capacity levels of outpatient department, and why standard queueing models do not apply. There is one further complicating factor that needs to be discussed: different types of patients, differing both in treatment times and urgency. It is possible to make separate slots or even different appointment blocks for different types of patients, but in general this is less efficient: then it might occur that slots of one type remain empty and there is a waiting list for the other slots. Smart planning is required here. We give a few examples.

Example 16.4.3 Appointments in outpatient clinics can often be of 2 types: a new patient or a repeat patient. New patients take more time, for example 30 against 10. Repeat patients are often planned long in advance. Separate slots for new and repeat patients are scheduled. To assure short access times for new patients repeat patients are rescheduled when required.

Example 16.4.4 For certain types of cancer short access times for an MRI scanner are required. However, numbers are small and because of that highly fluctuating. When the reserved slots are not filled in 24h before the slot then other types of patients from the waiting list are called to make sure that the MRI has a high utilization rate. When all slots are full then a patient might need to wait an additional day before the scan.

16.5 Appointment scheduling

In this section we consider again a doctor or some other form of capacity, but now we assume the number of slots fixed and we consider methods to minimize the waiting room time. Evidently, the same methods can be used when multiple resources have their own separate schedule. They can even be extended to the situation where the appointments are not made for a specific doctor, but we will not discuss the technicalities of such a generalization.

The appointment schedule that is most often used is called the *individual schedule*. It means that all appointments are made with equal distance. For example, consider a session of 3 hours where 12 patients are scheduled. Then they are scheduled at 15-minute intervals. Already in the fifties it was clear that this assignment rule is far from optimal. Often the doctor idles quite a lot the first hour, due to no-shows and possibly short treatment times. On the other hand, delays accumulate over the course of the session making the session often run late. For this reason a British mathematician and a British doctor, Bailey and Welch, worked together on this problem and came up with a rule that consists of taking the last patient of the individual schedule and putting it up front together with the first patient. This avoids the slow start that characterizes many clinics, and assures that there is on average some backlog without making the waiting room times much longer.

In this section we first consider methods to analyze this system. Evidently, simulation can be used, but as we also need to optimize over the possible schedules we have to rely on simulation optimization (see Section 7.6) which is both time-consuming and not that reliable. For this reason we also develop a time-inhomogeneous Markov chain model. It is the objective to schedule the appointments in such a way that the physician's idle time and the total patients waiting room time are minimized. These are conflicting objectives, and thus we have to find a trade-off between the two.

The model is as follows. We have a session consisting of T intervals, each of length d. d is a multiple of some time unit, say minutes. We want to schedule N patients during this period. We have a possibly random service time $S \in \mathbb{N}_0$, $\mathbb{P}(S = s) = p(s)$. No-shows can occur. We denote by q the no-show probability. The schedule is represented by a T-dimensional vector, x_t indicating the number of arrivals at the beginning of interval t.

Example 16.5.1 For $d = 5$, $T = 36$, and $N = 12$ the individual schedule is geven by $x = (1,0,0,1,0,0,\ldots,1,0,0)$ and the Bailey-Welch rule by $x = (2,0,0,1,0,0,\ldots,1,0,0,0,0,0)$.

For a given schedule x, we denote with π_t^- (π_t^+) the distribution of the amount of work at t just before (after) the arrivals that are scheduled at t. Using ideas from

discrete-time Markov chains we find that $\pi_{t+1}^-(k) = \pi_t^+(k+d)$, $\pi_t^+(k) = \pi_t^-(k)$ if $x_t = 0$, and $\pi_t^+(k) = q\pi_t^-(k) + (1-q)\sum_{i=0}^k \pi_t^-(i)p(k-i)$ if $x_t = 1$. If $x_t > 1$ then the $p(k-i)$ in the last formula have to be replaced by a convolution. From π_t^- and π_t^+ all performance measures can be calculated: the average waiting time of arriving patients W, the idleness of the doctor, and also the *tardiness* of the doctor. The tardiness Z is the time the doctor finishes after the end of the session, earliness counting for 0. In our situation, the expected tardiness is equal to $\mathbb{E}Z = \sum_k k\pi_{T+1}^-(k)$ (note that $T+1$ denotes the end of interval T, a moment at which no arrivals occur). Finally, let I denote the idleness of the doctor up to Td, the planned end of the session. In rows 1 and 2 of Table 16.5 we give the outcomes for the numbers and rules of Example 16.5.1, with exponentially distributed service times with expectation 15 minutes. For the sequence-entry the part between brackets is repeated the number of times indicated by the index. Thus $(100)^{12}$ means repeating 100 12 times, the individual schedule. $200(100)^{10}000$ is the Bailey-Welch rule. We see that the Bailey-Welch rule reduces the tardiness considerably, at the expense of an increase in waiting time.

rule	q	N	sequence	$\mathbb{E}W$	$\mathbb{E}Z$	$\mathbb{E}I$
individual	0.05	12	$(100)^{12}$	16:33	28:50	35:33
Bailey-Welch	0.05	12	$200(100)^{10}000$	20:29	21:30	23:06
$\min_x\{\mathbb{E}W + \mathbb{E}Z\}$	0.05	12	$110(100)^7(010)^3000$	19:32	22:15	25:27
$\min_x\{\mathbb{E}W + 1.15\mathbb{E}Z\}$	0.05	12	$1101(010)^3(001)^500010^4$	20:28	21:28	24:09
$\min_x\{\mathbb{E}W + \mathbb{E}Z\}$	0.3	12	$201(001)^90^6$	12:43	7:57	43:00
$\min_x\{\mathbb{E}W + \mathbb{E}Z\}$	0.3	15	$210(10100)^4(10010)^20^3$	20:10	18:25	31:31
$\min_x\{\mathbb{E}W + \mathbb{E}Z\}$	0.3	16	$2(10)^4(10010)^3(10100)^20^2$	22:49	23:22	27:53

Table 16.5: Outpatient scheduling policies with $d = 5$ and $T = 36$

It is clear that patient waiting time and tardiness are conflicting objectives. For any given trade-off the best schedule has to be found. The trade-off can be formulated in different ways: as a linear combination of the form $\min_x\{\mathbb{E}W + \alpha\mathbb{E}Z\}$ for some $\alpha > 0$, or by using a constraint, for example $\min_x\{\mathbb{E}Z|\mathbb{E}W \leq \beta\}$ for some $\beta > 0$. The policies that are found are all located on the *efficiency frontier*, discussed in Section 8.4. Rows 3 and 4 of Table 16.5 give outcomes of this type of analysis. Note that the outcomes of row 4 are slightly better than those for the Bailey-Welch rule. Thus the Bailey-Welch rule is not always on the efficiency frontier. However, the differences are so small that this has little practical importance.

The current schedule has a scheduled load of 100% and 5% no-shows. Without the no-shows the performance would be worse, the optimal schedule minimizing $\mathbb{E}W + \mathbb{E}Z$ has $\mathbb{E}W \approx 21$ minutes and $\mathbb{E}Z \approx 26$ minutes. Thanks to the no-shows

the performance remains acceptable. In the examples we took 5%, but in practice the no-show percentage can be as high as 30%! Planning at 100% capacity now leads to very good performance, but evidently a low utilization. Thus we plan more than 100%, anticipating the fact that some will not show up, a practice known in airlines as *overbooking*: see rows 5, 6 and 7 in Table 16.5. It is interesting to see that more patients are scheduled early, anticipating no-shows. Experiments can be executed using the scheduling tool at www.gerkoole.com/OBP.

Walk-ins

In this section we assumed health care services are delivered on appointment. Another delivery model are *walk-ins*: a patient simply goes to the health care facility and waits until it is his or her turn. This reduces the access time to 0, but it might considerably lengthen the waiting room time as the smoothing effect of the appointments is gone. When is this a realistic alternative? This method is evidently used by emergency services who have to adapt their capacity to the demand. It is also used by for example GPs and imaging facilities such as X-rays which are characterized by short service times and a load that is well below 100%.

16.6 Operating room planning

The scheduling of surgical operations, *operating room scheduling*, looks like appointment scheduling as discussed in the previous section. However, there are a few differences, the most important one being the following. In the appointment scheduling problem, when a doctor is ahead of time, he or she idles while waiting for the next patient. When performing operations, it is usually possible to call the next patient earlier if the operations are ahead of schedule. This means that the time to execute a schedule is simply the sum of the operation times. This makes that operating room scheduling is of a really different nature than appointment scheduling, it has more in common with machine scheduling.

Although the determination of the starting times is therefore less of an issue, OR planning has other challenges. Usually the schedule is made per specialty, thus the OR sessions (half a day or a full day in an OR) have to be assigned to specialties, usually on a weekly or bi-weekly basis. Then each specialty or groups of specialties have to plan in more detail the sessions they have, after which patients can be assigned to slots. Finally, at the day itself, depending on no-shows and fluctuations in durations, rescheduling between ORs has to be done to make sure that all operations are performed and finish on time. An important aspect are emergency operations, which

cannot be scheduled in advance and arrive during the day. There are two different model for this: having separate ORs for emergency, or keeping time free on all ORs for fluctuations and emergencies. There is evidence that the latter model is better.

Remark 16.6.1 To be able to plan it is crucial to have reliable unbiased data about durations. However, there is evidence that surgeons consistently underestimate the durations. For this reason it is better to use historical data for planning purposes. The durations evidently depend on the type of intervention, but also on the surgeon who is executing the procedure. There is evidence that durations are best approximated by a shifted lognormal distribution.

16.7 Clinical pathways

Delivering health care to a patients often consists of multiple step involving different types of capacity and specialties. Coordinating capacity decisions and planning and scheduling improves health care operations.

Example 16.7.1 Open heart surgery typically requires an operating room for a number of hours and an IC bed at the day of surgery. When one of the two is not available the operation needs to be cancelled. After discharge from the ICU the patient needs a bed at the medium care, etc. Planning this in advance leads to less cancellations and a better use of resources.

Health care delivery is often a sequential process consisting of different steps, such as visits to a doctor in the outpatient clinic, a diagnostic phase with exams such as X-rays, a surgical procedure, and time spent at possibly different clinical wards. Usually different departments are responsible for the different steps in the process, without any overall coordination. In the last decades in many different hospitals (and other health care institutions) *clinical pathways* or *integrated care pathways* have been created, following the ideas of the *coordinated supply chain* in manufacturing (see Chapter 12). It consists of coordinating the typical path of a homogeneous group of patients from admission to discharge, which stands perpendicular to the departmentental structure of most hospitals. If the group of patients is homogeneous and their resource usage predictable enough, then resources can be allocated over the whole path and not just step by step. Quality control is also easier because the group is well identified.

Certain types of treatment better fit in a clinical pathway than others: the more predictable the patient group, the easier it is. Setting up a clinical pathway is worth the investment when the patient group is big enough. It must be understood that some of the patient demand can never be organized via a clinical pathway: the paths

that these patients follow are so unpredictable that resources have to be allocated step by step. It is an interesting question what the influence of the pathways is on the remaining group of patients that are not organized according to clinical pathways. Many hospitals struggle with this question.

Some institutions go a step further: they specialize on certain clinical pathways. The term *focused factory*, introduced by W. Skinner, is often used in this context, refering to the fact that a production plant, in his opinion, should focus on a few process steps and excel in these. In such a focused factory a patient-centered pull-strategy can be used: all steps in the care process are planned at once. Not the seperate organizational units (radiology, operation rooms, etc.) play the central role, but the health care delivery process does. This stands in contrast with the traditional push policy. Focused factories have been successful in manufacturing, and also in health care remarkable results have been achieved using these ideas.

Delivering health care often goes through multiple stages, such as visits to a doctor, tests, surgery, time spent at wards, etc. We first consider the situation where it is allowed to wait between different stages. This occurs with outpatients where the different resources are planned one by one. It also occurs sometimes with inpatients: think of a patient lying at a ward waiting for a surgical slot, although this is usually undesirable, but not always avoidable (using *lean* terminology, waiting is considered waste).

Multi-stage processes with waiting in between ressemble flow lines or job shops, depending on their characteristics. Waiting is seldomly evenly distributed, most of the waiting usually occurs at the *bottleneck*. For this reason, attention should be focused at the bottleneck, following the ideas of the Theory of Constraints (see page 200). At the bottleneck the available capacity should be used as efficiently as possible. As a result, waiting room time at the bottleneck resource is probably higher than at other resources. In the case the bottleneck is only used by a single type of patients, then it is possible to schedule using methods such as the one discussed in Section 16.5. In case of multiple types of patients we have to decide how many slots to allocate to different types of patients. If we do not do this then resources are used by the patients that make their appointment longest in advance, and these are usually not the most urgent or profitable. Especially if a resource is used by outpatients on one hand and emergency and/or inpatients on the other, then slots should be reserved for the latter categories. In the case of emergencies, there is a medical need for this; in the case of inpatients the costs for waiting for the first free slot is much too high because they are occupying expensive resources such as a bed.

Remark 16.7.2 In a simple ∞-buffer flow line the throughput is determined by the slowest

server. However, in the case of finite in-process inventory, the throughput is a complicated function of all parameters, meaning that also servers that perform close to the bottleneck also have a big impact on the total performance. The same happens in health care, and therefore we should make sure there is sufficient slack on the less expensive resources to make sure the most expensive resource, the bottleneck, is used as efficiently as possible.

This is not always the case: for example, we still see hospitals reducing the number of meeting rooms while this leads to a less efficient use of doctors, which is a more expensive resource. What is introduced as a cost-saving measure actually leads to less throughput and less income.

As the concept of planning the steps in the health delivery process one by one is abandoned ever more often, the need to plan the different step consecutively increases. Next to that, for inpatients, this always has been the case in many situations. For example, after open heart surgery an IC bed is necessary. If none is available at the time the operation is supposed to start, then the operation has to be cancelled. This leads to challenging planning problems for which simulation is basically the only evaluation method. An example is the simulation tool at www.gerkoole.com/OBP, in which bed capacity and the OR schedule having to be determined. Output is the capacity usage and the number cancellations.

16.8 Further reading

Brandeau et al. [27] contains a collection of papers on OR and health care. A journal at the interface of health care and OR/MS is *Health Care Management Science*.

Gallivan et al. [60] considers a mathematical model for a ward with a single class of patients. Further work in this area can be found in De Bruin et al. [30], dynamic bed assignment in Bekker et al. [20]. Evidence that higher nurse-to-patient ratios lead to higher mortality is documented for example in Needleman et al. [118].

The idea of determining the optimal panel size is introduced by Green & Savin, see Green [69] for a tutorial on this and other subjects. More information on advanced access can be found on the web site of the IHI, `ihi.org`.

The Bailey-Welch rule was introduced in [161]. Koeleman & Koole [95] gives a procedure to find the optimal outpatient schedule under fairly general conditions. A paper in which the term overbooking is used in the context of patient scheduling is Laganga & Lawrence [102]. They also state that in the case they considered 30% no-shows is not exceptional.

Information on project management for process improvement projects in health care can be found in Belson [21]. Toussaint & Gerard [155] is an excellent introduction

to the application of lean principles in health care.

Cardoen et al. [33] review the literature on operating room planning and scheduling. A big program to improve operating room planning in the Erasmus Medical Center in Rotterdam lead to a number of publications, see for example Van Houdenhoven [76] and other references in [33].

16.9 Exercises

Exercise 16.1 Consider Exercise 3.5. Under the Poisson assumption, calculate β and the occupancy.

Exercise 16.2 Reproduce the numbers of Table 16.2.

Exercise 16.3 Reproduce the numbers of Table 16.3. This requires solving a birth-death process.

Exercise 16.4 Reproduce the numbers of Table 16.4. This requires solving a 2-dimensional Markov process.

Exercise 16.5 Consider Example 16.3.2. We implement the idea of protecting type 2 patients by introducing a threshold policy.
a. Develop a birth-death process that models this situation.
b. Compute the blocking probabilities for various values of the threshold and report on the consequences.

Exercise 16.6 A small IC unit has 6 beds and two types of patients. The average LoS is 3 days, and on average one patient arrives per day in each class. Compute the blocking probabilities for each class in the following situations:
a. No admission control is used, i.e., every patient is admitted unless all beds are occupied;
b. When exactly 1 bed is available then type 2 is blocked and type 1 is admitted.

Exercise 16.7 Consider a ward with s beds and two types of patients where s_i beds are earmarked for type i, $s_1 + s_2 \leq s$.
a. Make a drawing of the state-transition diagram of the two-dimensional Markov chain of this system.
b. Compute the stationary rejection probabilities for some realistic numbers.
c. Vary the values of s_1 and s_2 and report on the outcomes.

Exercise 16.8 Use an Erlang C model to model a doctor having 10 slots per day and a demand that is state-dependent: it decreasing linearly from demand equal to 20 per day when there is no waiting to 0 when the waiting time is 100 days.
a. Model this as a birth-death process.
b. Solve the birth-death process, make a histogram of the stationary probabilities, determine the most likely state and the average waiting time.

Exercise 16.9 Reproduce the numbers of Exercise 16.4.2 by solving a birth-death process.

Exercise 16.10 How could you analyze the system of Example 16.4.4? Implement your solution and analyze different strategies for an average demand of 2 slots per day. Output should consist of the demand fraction that cannot be scheduled the next day and the average number of slots that needs to be filled with other patients.

Exercise 16.11 A reservation system for MRI slots works as follows. Every day k slots are reserved for semi-urgent patients. Patients (all seen in the morning) are booked for slots for the same day, or, if no slots are available the same day, then for the next day. When there are even no slots the next day then an ad hoc solution is sought with cost c_1. At the end of a day slots are possibly given back for the next day to make sure that the number of free slots does not exceed $m \leq k$. Giving back a slot costs c_2. Any left-over slots at the end of the day cost c_3. Demand is assumed to be Poisson distributed.
a. Model this process as a Markov chain and indicate how to compute the costs from the stationary distribution.
b. Compute these numbers for $k = 2$, $m = 1$, an average demand of 1, and $c_1 = 20$, $c_2 = 1$, and $c_3 = 10$.
c. Formulate your ideas on how to find optimal values for k and m.

Exercise 16.12 Simulate in python an appointment schedule of length 3 hours, 5-minute intervals, and 12 patients, with lognormally distributed service times with mean 13 and standard deviation 5. (Make sure your distribution has the right parameters.) No no-shows (all patients show up), patients are on time.
a. Use simulation to estimate the expected average patient waiting time and the tardiness of the doctor (the time the doctor is busy after the end of the appointment block), first for the individual schedule (where patients are equally spaced).
b. Try to find the best possible schedule, counting the doctor's tardiness 2x heavier than the average patient waiting time. You can automate the search or use a trial-and-error procedure. In both cases, describe your procedure. To get the maximal number

of points you should implement a scientifically sound algorithm and not just trial-and-error.
c. Calculate for the optimal schedule the average and each 10th percentile of the waiting time of each patient, and put them in a plot with the patient number on the x-axis.
d. Try to find the best possible schedule for minimizing the sum of the doctor's tardiness and the maximal expected patient waiting time.
Pay attention to the statistical correctness of your findings, and try to program as efficient as possible (e.g., by using appropriate vector functions).

Exercise 16.13 The imaging department of a hospital offers walk-ins for CT scans of MRI scans for which they both have 1 scanner. Demand is on average 4 per hour for CT scan, 0.8 for MRI scan. A CT scan takes exactly 12 minutes, an MRI scan 1 hour. What are the expected waiting times and occupancies? Where do you advice to use walk-ins?

Exercise 16.14 Consider 2 operating rooms each open for 9 hours. On OR1 2 operations of each exactly 4.5 hours are planned. On OR2 8 operations are planned which are i.i.d. with lognormal durations with mean and s.d. 1. These durations include switch-over times, cleaning times, etc.
a. Simulate the expected tardiness.
b. Simulate the expected tardiness using a normal approximation of the durations and verify this using a formula.
A consultant wants to split the variability: each OR does 1 operation of the first type and 4 of the second type.
c. Repeat questions a and b for this situation.

Chapter 17

Call Centers

In this chapter we study call centers. In call centers there are many interesting modeling questions, often related, but not restricted, to queueing models. Studying quantitative issues of call centers is not just a matter of solving mathematically challenging problems: there is a big economic interest because of the large number of call centers in operation nowadays.

We give an overview of the practice and science of call center workforce planning. It consists of multiple steps, which we discuss one by one, after a general introduction.

17.1 Introduction

Call centers are a fascinating area for (stochastic) modelling. In manufacturing most production is being done before the demand occurs, the product is lying on a shelf in a shop or a distribution center waiting for customer demand. In (non-urgent) health care production is smoothed in time to meet capacity: a patient makes an appointment with a health care provider at a moment that suits above all the provider. In aviation and hospitality demand is pushed by financial incentives towards low-demand time slots. Inbound call centers have in common with emergency health care that demand has to be met almost instantaneously by supply. And while a hospital has at least 15 minutes to prepare for the arrival of a trauma patient, a call center often has to answer a call within 20 seconds. And it can be life-saving, as is the case with an emergency call center.

To be able to deliver this type of service planners have to deal with fluctuations, unforeseen (such as the variability of the Poisson process, or illness of employees,

often called agents) and foreseen (such as intra-day and intra-week seasonality in demand). Call centers cannot react instantaneously to all fluctuations, and therefore have to schedule overcapacity. Designing the call center in such a way that little overcapacity is needed, and planning the right amount and types of overcapacity is the essence of workforce planning.

Nowadays many call centers handle contacts through different communication *channels*, such as chat and email. However, inbound calls is often the most prominent channel. To give credit to the different channels the term *contact center* has been introduced. Few people however use it, thus a call center is most of the time a contact center mixing contact from different channels. A notable exception are call centers dedicated to outbound marketing campaigns. Through *predictive dialing* they deal with fluctuations in the fraction of calls that are answered and the speed at which this is done. Quite a number of patents for algorithms can be found on Google Scholar.

Will there still be call centers in say a decade? We see a tendency for offering automated customer service by using for example AI in chat bots and call avoidance by for example improved web sites. Indeed, there is evidence that making calling unnecessary is the best customer service (Dixon et al. [54]). And if people call, avoid that they have to make another call later on. Avoiding calls is also cheaper, and as most call centers are seen as *cost centers*, there is a strong incentive to reduce costs. However, there is no evidence that the call center market is shrinking, on the contrary (Mazareanu [112]). A possible explanation is the popularity of shared service centers which operate effectively as call centers (e.g., the human resources department at our university). As such, we see a tendency across industries, from decentralised service to centralised service (operated as a call center, potentially *offshored* to a country with lower wages) to self-service.

This chapter focuses on the practice of workforce management (WFM). (A better name would be workforce planning, but we will stick to the commonly used terminology.) As framework we use the different steps in the WFM processes. The three central planning processes are: budget planning, capacity planning, and agent scheduling. See Figure 17.1. "x" refers to the day of execution, "x+1Q" for example means 1 quarter before the day of execution. Note that many companies use *business process outsourcers* (BPOs) to handle (parts of) their call volume. To allow them to prepare for their job forecasts or required staffing levels are communicated at multiple moments in time.

As can be seen from the figure every step starts with forecasting. An exception is intra-day management: in practice the forecast is rarely updated after the agent schedule is made, although that would likely result in an increased accuracy. There

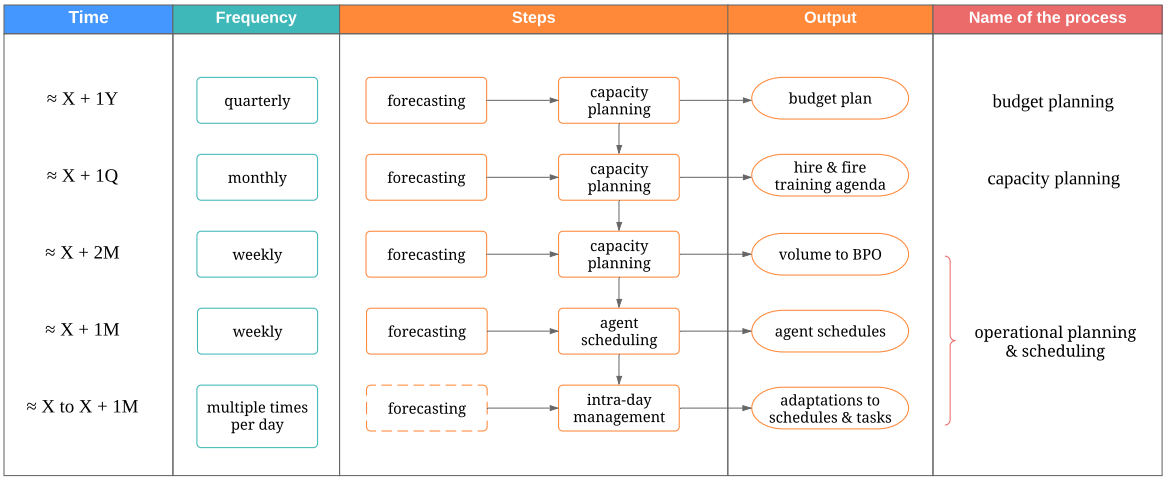


Figure 17.1: The WFM processes

are three processes:
- the long-term budget process, which is input for the corporate management which sets the financial boundaries;
- the tactical capacity planning process, where decisions concerning the agent pool are made, mostly the hiring of new agents and the training of new skills;
- the short-term operational planning process, which starts with deciding which volume goes to the external partners, and then consists of agent scheduling (which need to be communicated a few weeks in advance), and finally intra-day management which is done at the day itself.

Depending on the particular call center the situation might be slightly different, and smaller call centers with stable volumes might not execute the long-term steps explicitly. Note also that this scheme is biased towards the European situation with its strict labor laws forcing call centers to schedule carefully and publish schedules well in advance.

Next to the three processes from Figure 17.1 call centers have two more less explicit processes: a long-term, ad-hoc process which is about improving the overall design of the call center: the shift structure, the way forecasts are made, opening times, channels that are offered, etc. This process also designs all underlying processes. Finally, there is the real-time routing, the assignment of customer contacts to agents, which has a big impact on the performance. This is automated and part of the telephony/omnichannel switch. Although this could also benefit from updated forecasts and other real-time information this is rarely done.

Note also that we left out the connections with other departments. Forecasting for example takes input from marketing and sales to obtain the dates of marketing campaigns and sales forecasts, and the budget plan is used in negotiations with higher management to set the final budget. Furthermore, the processes are not always as linear as they seem: there might be interaction between forecasting, scheduling and marketing about the feasibility of marketing campaigns; capacity planning might lead to adaptations to the budget, etc.

WFM plays a supporting role in a call center. It helps achieve the goals of the three stakeholders, customers, employees and management: giving good service by satisfied employees at a reasonable price. Good service is usually defined by service level agreements (SLAs) which serve as constraint in all WFM steps. Employee satisfaction is represented for example in the types of possible shifts and fairness among agents, achieved by the routing rules. The financial side is the main reason for having the budgeting cycle, and the budget is discussed regularly to see if there are any exceptions.

In the next sections we discuss one by one the different steps of the WFM processes, starting with forecasting.

17.2 Forecasting

In call centers the numbers of arriving contacts is the most important time series to forecast. Many call centers have different lines or types or calls, each line should be forecasted separately. Other time series to forecast include the handling times and the fraction of sick agents.

Call center arrivals are well modeled by an inhomogeneous Poisson process, as is made plausible by Kim & Whitt [88]. The parameter is determined by trend, intra-year, intra-week and intra-day seasonalities, and events. This is the framework as discussed in Section 2.5, and the methods and related theory developed there largely applies to call centers. In practice however, most call centers either use a self-made spreadsheet or judgemental forecasting. Forecasting is done first at the daily level, for example, by a simple decomposition approach that adds the increase over a year to last year's volume. Written in a formula, with h the historical volumes, $\hat{h}$ the forecast, and w and y time periods of a week and a year:

$$\hat{h}_t = h_{t-y} \frac{h_{t-w}}{h_{t-y-w}}.$$

This forecast is adapted using estimations of the impact of events on t, $t-w$, $t-y$,

and $t - y - w$. It can be made more sophisticated by separately predicting weekly volumes and intra-week profiles, and by estimating the yearly increase by averaging over multiple weeks. Some call center scheduling tools offer forecasting functionality but rarely more advanced than this. Very few call centers employ advanced forecasting methods such as linear regression as discussed in Section 2.5. An alternative method that deals with all aspects is described in Hyndman [77], using smoothing methods as described in Hyndman & Athanasopoulos [78], and a separate regression for events using dummy variables.

Decomposition methods, by which we mean methods that determine the factors that influence volume one by one, only work in a multi-pass setting because of the dependencies of the underlying variables. For example, the occurrence of outliers can only be determined if you know the seasonalities. But the seasonalities can be better estimated if the events and outliers are filtered out. While some form of decomposition is commonly used by forecasters in practice, multi-pass methods are rarely used and neither studied in the literature.

Finally, data scientists have the tendency to use recently developed black-box methods, often neural nets, for forecasting. However, the lack of interpretability and adaptability makes it hard to get their outcomes accepted in practice.

It was argued in Section 2.5 that the WAPE is the best error measure. Because of the Poisson noise there is a minimal absolute percentage error (APE) of $\sqrt{2/(\lambda\pi)}$ for rate λ. By weighing minimal APEs over multiple intervals the minimal WAPE for a longer period can be obtained. It is interesting to note that managers sometimes require a WAPE which is lower than this minimal WAPE: "strive for 5" is often said, but for volumes below 250 this is impossible because the minimal WAPE is higher than 5%.

Remark 17.2.1 Note that the *average handling time* (AHT) often also needs to be forecasted. Again, the WAPE can be used to determine the accuracy of the AHT forecast, but the minimal APE now depends on the variability of the handling time distribution S and the number of arrivals in the interval n. It is given by:

$$\frac{\sigma(S)}{\mathbb{E}S}\sqrt{\frac{2}{n\pi}}. \tag{17.1}$$

In practice the WAPE is often considerably higher than the minimal WAPE. To determine the intra-year seasonality or the impact of events such as Christmas accurately many years of data is required, which is rarely available. Other events such as the impact of good weather or a system problem are not known at the day the schedule and therefore the forecast was made. For these reasons WAPEs of 20% or higher are not exceptional.

In the urge to explain the forecast, forecasters, and especially their managers, like to include time series such as sales in the forecast. Forecasts made this way are called "ratio forecasts", because it is supposed that a certain fraction of new customers call. However, again, a forecast of the external variable is needed, while the trend most of time shows considerable collinearity with the sales. Furthermore, the fraction might change. Therefore, it is questionable whether including a sales forecast improves the forecast. Testing it is the only way to find out, and often it is indeed not the case. Adding a variable such as a sales forecast is useful if it contains information not yet contained in the call volumes, such as qualitative opinions. This is often the case with long-term forecasts, made for budgeting reasons or capacity planning. An additional advantage is that it helps explain the forecasts and also their errors.

Note that these errors will be considerable, especially for long-term forecasting (Makridakis [109]). Therefore, the question should not be whether the forecast is reliable enough, but if we have enough flexibility to deal with the inevitable error. Call center managers recognise the importance of flexibility, but hardly ever make the connection between forecasting errors and the amount of flexibility required. This should be part of *capacity planning*, the long-term determination of the required capacity.

Forecasting is often done for the total number of *offered* calls. However, this includes retrials: callers who abandoned earlier and called again later, see Figure 17.2. Data analysis shows that retrials often occur shortly after the first attempt, almost all within the same day (Ding et al. [51]). Usually one knows the numbers of connected and abandoned calls, but the fraction of retrials is not known, unless the callers can be identified. Ding et al. [51] propose a statistical method to determine the "fresh" volume by using the retrial percentage as a variable in the forecasting model. In practice, taking the average between the offered and handled numbers of call often works well, corresponding to 50% retrials. Note that there are also *recalls*, callers who call a second time to get further advice. Recalls add to the call volume and therefore to the workload, but they are also a very important driver of customer dissatisfaction (Dixon et al. [54]). Reducing it however is outside the scope of WFM.

After determining daily volumes, they have to be drilled down to the intra-day volume. Typically forecasters will base themselves on what they consider to be similar days as the one they are about to forecast: same day of the week, not too long ago, similar events. Then they take the average of the *profiles*, the normalised volumes, and multiply that with the daily forecasts. However, this method leads to considerable overfitting, the average profile often shows quite some variability. Much better results are obtained by using the fact that you expect neighbouring intervals not to

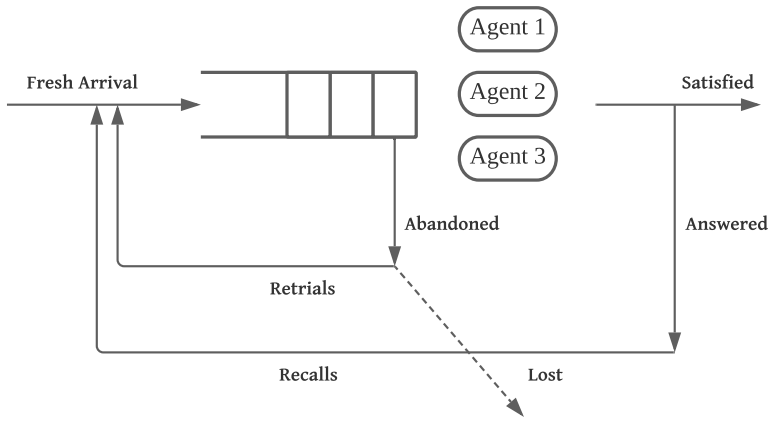


Figure 17.2: Retrials and recalls

vary that much, and by fitting a polynomial or a smoothing spline. This proves to work quite well.

17.3 Staffing

One of the main objectives of a call center is to offer a reasonable waiting time to the customers. The measurement on the waiting time is called *service level*, often defined as one minus the tail of the waiting time distribution. A SL of 80/20 means that 80% of the customers waits less than 20 seconds. The expected waiting time (aka *average speed of answer*, ASA) is also often used.

Agent scheduling concerns the construction of schedules such that, amongst other objectives, SLAs are expected to be met to the extent possible. The most commonly used SLA is 80/20. With the SLA as constraint the minimum required staffing in every interval is determined, sometimes explicitly, or implicitly in the scheduling algorithm. If it is done explicitly, and then given as input to the scheduling algorithm, then it is often done by the forecasters and even called *workforce forecasting*. We will call it *(safety) staffing*, as it entails planning overcapacity to deal with fluctuations in workload, terminology inspired by the safety stock which is additional stock to deal with fluctuations in demand. Staffing is probably the best-studied part of WFM, and the starting point of many scientists interested in WFM, explaining why many queueing scientists (used to) work on call centers.

We will make a difference between single and multi-channel and single and multi-skill operations. Staffing is done at the interval-level, usually 15 minutes. Even though agents can often handle multiple skills and/or channels, they are often scheduled during one or more intervals to work on a single skill and/or channel. We will first look at staffing for these single-skill single-channel operations, starting with inbound. Then we will look at staffing in a *blended* multi-channel environment and in the presence of *skill(s)-based routing*, where in real-time a contact from the optimal channel or skill is being pushed to the agent.

It is commonly assumed that arrival rates and numbers of agents are stepwise constant functions, constant during each quarter. This is motivated by the fact that arrival rates are expected to change little during each quarter, and that schedule changes are only possible at the quarter. In this situation the so-called SIPP approach (Green & Kolesar [68]) is an obvious choice: you assume constant parameters in each interval, and use a stationary queueing model. The $M|M|s$ or Erlang C model (see Theorem 5.4.1) is most commonly used in practice. Let us first look into its properties and then ask ourselves the question how good it is.

It is hard to obtain qualitative insights from the Erlang formulas, for example how they behave when you increase scale. There is a simple explicit approximation that is more insightful, the *square-root staffing* formula. For λ the arrival rate and β the average handling time, it says that staffing should be approximately at $\lambda\beta + \alpha\sqrt{\lambda\beta}$ with α a parameter depending on the SL only. The square root can intuitively be interpreted: if you add 2 i.i.d. random variables then the standard deviation is multiplied by $\sqrt{2}$. The same holds for safety staffing, because it is there to handle fluctuations in load. This clearly shows the economies of scale which is one of the reasons why we want agents to be multi-skilled. It also shows decreasing returns, as $\lambda\beta + \alpha\sqrt{\lambda\beta}$ is concave in $\lambda\beta$, which tells us that not all agents need to be multi-skilled. In Figure 17.3 we plotted for fixed AHT and SL and varying FC both the staffing levels for Erlang C, square-root staffing, and the corresponding occupancy. It is hard to see the non-linearity in the staffing plot, but it is clear from the occupancy plot.

One of the main features of customer behavior that is lacking in the Erlang C is *abandonment*: the fact that customers get annoyed waiting and hang up before they get service. We assume every customer has some random patience P, and define W_v as the *virtual waiting time*, the waiting time of the customer who is willing to wait as long as necessary to get service. If $P < W_v$ then the customer abandons. The model including abandonments is commonly written as $M|M|s + G$, with $+G$ denoting the generally distributed patience P. In the case of exponential patience we

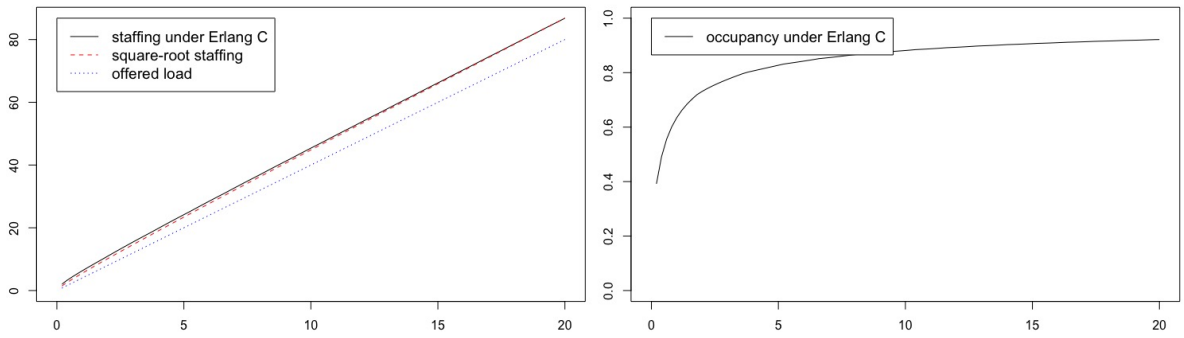


Figure 17.3: Staffing and occupancy for Erlang C with varying arrival rate, AHT 4 minutes and 80/20 SL

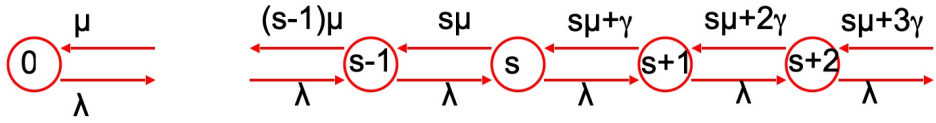


Figure 17.4: Transition diagram of the Erlang A birth-death process

call it Erlang A.

Let us model the Erlang A. The arrival rate is $\lambda(x, x+1) = \lambda$ and the departure rate for state $0 < x \leq s$ is equal to $\lambda(x, x-1) = x\mu$, as in the Erlang C model. However, for higher states the departure rate is different: $\lambda(s+x, s+x-1) = s\mu + x\gamma$ for all $x > 0$, with γ the patience parameter. See Figure 17.4.

Using Equation (4.8) we can find the stationary distribution. Note that it always exists because the sum in Equation (4.9) always exists (as long as $\gamma > 0$). This means that the system is always stable, independent of the values of λ, μ and s.

The derivation of the waiting time distribution is more involved than that of the $M|M|s$ queue (Section 5.4), because the waiting time, conditioned on the state, does not have a gamma distribution as is the case for the $M|M|s$ queue. If a customer arrives in state $s+k$, i.e., there are k waiting customers in front of him or her, then this customer has to wait a sum of exponentially distributed random variables with rates $s\mu + k\gamma, s\mu + (k-1)\gamma, \ldots, s\mu$ before being served. Such a distribution, a sum of exponentials with different rates, is known as a *hypoexponential* distribution. We derive the tail distribution for hypoexponential distributions for which all rates are different.

Let $X_i \sim \exp(\mu_i)$, $i \leq K$, and $\mu_i \neq \mu_j$ for $i \neq j$. We define F_k as the distribution

function of $X_1 + \cdots + X_k$, $k \leq K$, $\bar{F} = 1 - F$. The result we show is as follows:

$$\bar{F}_k(t) = \sum_{i=1}^{k} \prod_{\substack{j \leq k \\ j \neq i}} \frac{\mu_j}{\mu_j - \mu_i} \mathrm{e}^{-\mu_i t}, \quad 1 < k \leq K. \tag{17.2}$$

Of course, $\bar{F}_1(t) = \exp(-\mu_1 t)$.

From properties of the exponential distribution we find for $h > 0$ small, $k > 1$:

$$F_k(t+h) = \mu_k h F_{k-1}(t) + (1 - \mu_k h) F_k(t) + o(h). \tag{17.3}$$

Rewriting and taking the limit as $h \to 0$ gives

$$\bar{F}_k'(t) = \mu_k (\bar{F}_{k-1}(t) - \bar{F}_k(t)), \tag{17.4}$$

for $k > 1$.

We prove (17.2) by induction to k. Note that for $k = 1$ the result is correct. Assume it holds up to $k-1$ for some $k > 1$. As a solution to (17.2) we try $\bar{F}_k(t) = \sum_{i=1}^{k} \alpha_{ik} \mathrm{e}^{-\mu_i t}$. By differentiating we find $\bar{F}_k'(t) = -\sum_{i=1}^{k} \mu_i \alpha_{ik} \mathrm{e}^{-\mu_i t}$. Equation (17.4) must hold for all t, and therefore the coefficients of all power functions should be equal. This leads to:

$$-\mu_i \alpha_{ik} = \mu_k \prod_{\substack{j \leq k-1 \\ j \neq i}} \frac{\mu_j}{\mu_j - \mu_i} - \mu_k \alpha_{i,k}$$

for $i \leq k-1$. From this it follows, for $i \leq k-1$, that

$$\alpha_{ik} = \prod_{\substack{j \leq k \\ j \neq i}} \frac{\mu_j}{\mu_j - \mu_i}.$$

The same form for $i = k$ follows, after some tedious calculations, from the fact that $\bar{F}_k(0) = \sum_{i=1}^{k} \alpha_{ik} = 1$.

A web-based calculator for the Erlang A model in which this method is implemented can be found on www.gerkoole.com/OBP. Figure 17.5 gives SLs and abandonment percentages for several values of γ and varying numbers of agents.

We see that adding abandonments to the Erlang model is a valuable extension that gives really different outcomes. It is clear that the Erlang A model is less sensitive to parameter changes than the Erlang C model, the SL graph is less steep. For example, Erlang C predicts a bigger difference in SL when a forecasting error is made than

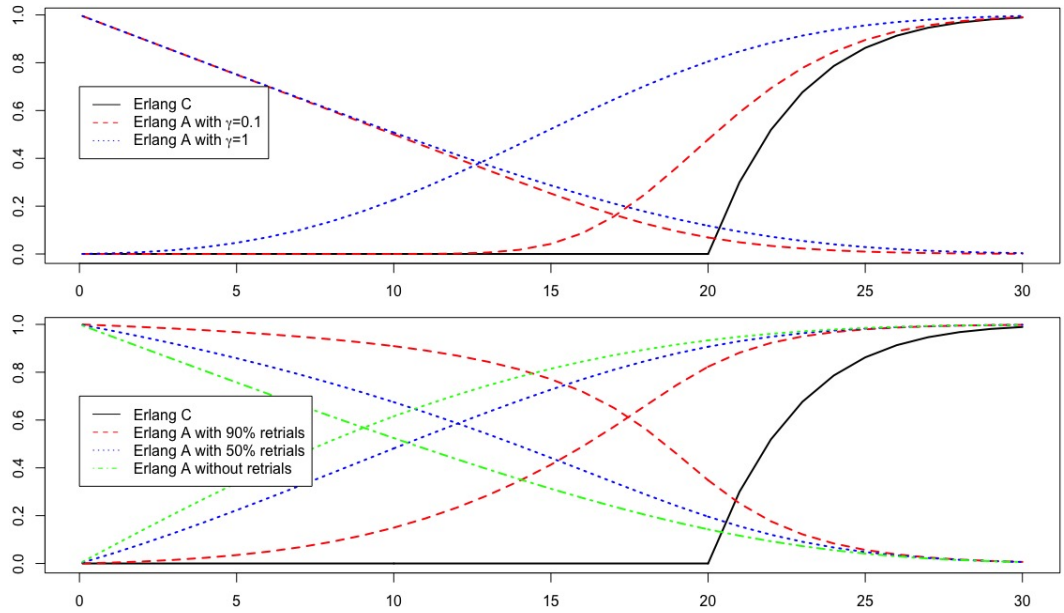


Figure 17.5: SL (increasing) and abandonment percentage (decreasing) for various levels of average patience (left) and retrial rates (right) and $\lambda = 5$, $\beta = 4$, and TTA 1/3

Erlang C. By comparing the Erlang models to reality it was found that Erlang A gives a better fit. Thus the reality is more robust to errors and changes than Erlang C suggests.

Compared to the Erlang C, the Erlang A requires one more parameter: the patience distribution or its expectation, depending on the exact model used. The waiting time of a customer W_q is the minimum of its patience and the time to service, $W_q = \min\{P, W_v\}$, thus the patience is a *(right-)censored variable*, you do not always observe (the tail of) it. The famous Kaplan-Meier method can be applied, which constructs P from censored observations. When P is exponential, then $\mathbb{E}P$ can be found from the observed waiting time and the fraction of abandonments: $\mathbb{E}P = \mathbb{E}W_q / \mathbb{P}(P < W_q)$ (see Example 1.8.1). It needs to be said that using Erlang A with an expected patience of 5 or 10 minutes is already much better than Erlang C.

You can also use the patience as a tuning parameter, then it models right away other features which are not part of the model. Among the features that are not modeled are non-exponential handling times and patience. It is known that multi-server queues are almost insensitive to the service time distribution, thus taking them exponential is a sensible assumption. This is less the case for the patience distribution:

because eventually any waiting customer will get served, it is the initial hazard rate that is most important, not the expectation. Thus is an additional argument for fitting the Erlang A to the data and using the abandonment rate as the tuning parameter.

Erlang A gives also the possibility to include the abandonment rate in the SLA. For example, the abandonments needs to stay below 5%. If the SL is based on the waiting, as in the regular, SL, we have to decide how to count abandoments. It is common in science to use the *virtual waiting time*: the time an arbitrary customer would have to wait if her patience were ∞. However, this measure is not measured in a call center, thus the performance cannot be verified. In practice other definitions are used, for example the fraction of all calls being answered within the *time to answer*.

Weighted averages

The SIPP approach splits the day in intervals, for each of which the SL is predicted using different parameters. To obtain daily values weighted averages have to be taken. Because of the non-linearity of the Erlang formulas (see Figure 17.3) we cannot use average values as input and get average staffing as output. However, numerical experiments show that using averages gives nearly the same results. This is due to the fact that staffing is only slightly non-linear as a function of the forecast. Because of the concavity of the staffing function it slightly overestimates. This makes staffing simpler, especially in the case of capacity planning, where only aggregated outcomes are required. No intraday profiles are needed for the computations.

To avoid employee fatigue some call centers prefer staffing at a certain occupancy percentage, which means that staffing is linear in the forecast. Then averaging and staffing can be interchanged and both methods give the same outcome.

The widely used Erlang97 Excel add-in (Bromley [28]) also has the option to compute abandonments. It is based however on a waiting-time quantile of the Erlang C, thereby making two errors: it does not model the fact that Erlang A generally has a better SL than Erlang C because some customers leave the queue, and it assumes the patience is the same for all customers.

The next step in refining the Erlang C model is adding retrials. Abandonments will lead to retrials. See the previous section on forecasting on how to estimate the retrial fraction. As most retrials occur quickly we assume the retrials occur in the same interval, by increasing the arrival rate with the abandonment rate. This increases the abandonments, which leads to an increase in retrials, etc. This procedure converges quickly. Note that this procedure requires as input for the Erlang model the rate of new, *fresh*, requests.

Without abandonments the occupancy has a simple formula: $\lambda\beta/s$. For the model with abandonments retrials this is more complicated, λ needs to be replaced by $\lambda(1-$

$a)/(1-pa)$, with λ the fresh arrival rate, p the retrial fraction, and a the fraction abandoned calls. Furthermore, it is important not only to add safety staffing to deal with fluctuations in arrivals and handling times, but a planner should also account for unplanned break, absence of agents due to illness, etc. This is commonly known as *shrinkage*, and can be considerable. Thus s should be replaced by $s/(1-u)$ with u the shrinkage. We introduce the following terminology:
- $\lambda\beta$ is called the net workload;
- by adding safety staffing we get s, the gross workload but also the net workforce;
- adding the shrinkage leads to the gross workforce.
To avoid rounding twice a "fractional" Erlang is useful, which interpolates linearly between the neighboring integers.

Example 17.3.1 An interval has the following parameters (all in minutes): $\lambda = 5$, $\beta = 4$, $\mathbb{P}(W_v \leq 1/3) \geq 0.8$, $\mathbb{E}P = 5$, $p = 0.5$, and $u = 0.3$. Then, using the calculator on www.gerkoole.com/OBP, we find that the net workforce is 23.08 and the occupancy 85.1%, see Figure 17.6. The gross workforce is 32.97.

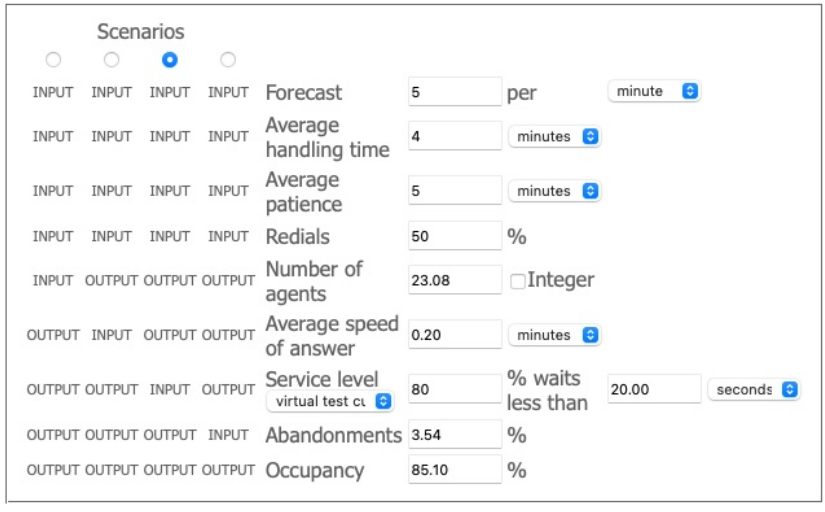


Figure 17.6: Example of an Erlang A calculation

It is interesting to note that delay announcements, which provides waiting time estimates, influences customer patience. Although customers might appreciate it, the delay announcement can also induce some customers to abandon earlier, leading to peaks in abandonments.

Although SIPP combined with an Erlang model is the commonly used method in practice, there are a number of problems with such an approach. We discuss two of them.

In the first place, SIPP uses a stationary model. But the queue is not in a stationary situation at the beginning of each interval. Depending on the parameters of the previous intervals, you might, for example, expect a backlog. There are a number of methods available to handle this, of which the *stationary backlog carryover* (SBC) approach from Stolletz [147] performs best. And a stationary model predicts expected performance. If you schedule using SIPP at the level of your SLA, then in roughly 50% of the cases, you won't reach your daily SLA. The error can be quite big (Roubos et al. [133]). In practice, this is unacceptable. The problem is usually solved by intra-day management.

In the second place, there are many factors that influence performance that are not modelled by Erlang C nor Erlang A, such as the impact of short unscheduled breaks and the behavior of agents under longer periods of high workload. Only recently the first attempts to validate the Erlang models based on realized service levels were undertaken (Ding et al. [53]). This is a good reason to use the abandonment as a tuning parameter, or to move to a different model that has the potential to model more factors, such as a statistical/machine learning approach.

Next to inbound a variety of other channels are used. They can be divided into synchronous and asynchronous communication. Email, webforms and old-fashioned mail and fax are asynchronous. Usually the time-to-answer is multiple hours or days, at least multiple intervals. This means that fluctuations have to be dealt with by flexibility in scheduling, not by safety staffing. Inbound is synchronous. Another noteworthy synchronous channel is *chat*. Its difference with inbound from the point of view of WFM is that a chat agent can do multiple chats in parallel, usually 2 or 3. When all agents are saturated, customers wait in the queue, just like inbound. This parallelism increases efficiency. When an agent answers one customer, the other(s) can formulate their responses. However, it makes the total handling time per chat longer: sometimes, a customer has to wait for a chat to become available. Quantifying the durations are somewhat challenging but can then be used to extend the Erlang models for chat systems. At www.gerkoole.com/OBP a chat calculator can be found.

Moving decisions to a later moment when better information is available is a general principle to improve decision making. One way to do this is to move the decision which type of task to do from the schedule to the routing. In a multi-channel environment this leads to *blending*, in a multi-skill environment to *skill-based routing* (SBR).

We now discuss how staffing can be done in these environments.

Blending is usually executed by blending synchronous and asynchronous channels, such as inbound and email or outbound. When the asynchronous channel can be interrupted to deal with priority with inbound, then staffing is easy: inbound is staffed as discussed, and the overcapacity with respect to the expected load is filled with email. Things get more complicated when the asynchronous channel cannot be interrupted as in the case of outbound. Scheduling to 100% occupancy will lead to a low SL on inbound. The solution is a *threshold policy*: inbound has priority, when more than a certain level of agents are free then an outbound call is assigned. When the handling times are equal then this system can be modeled as a birth-death process, otherwise as a 2-dimensional Markov chain. See the calculator on www.gerkoole.com/OBP and Exercise 17.3.

Now we move to SBR. Specialization, one of the driving forces of efficiency and quality of economic activity, makes that agents are rarely generalists but specialized in one or a few skills. Too few would lead to lack of flexibility. For this reason different agents have different subsets of all skill, and skill-based routing is required. Because of the lack of closed-form formulas, simulation is the only viable option for SBR, apart from some crude approximations. In practice most often *static* rules are used, meaning that they are independent of current staffing levels, SL, etc. This leads to the necessity to correct the routing rules through intra-day management, which is often too late and certainly not optimal. Typical rules include priority policies, *overflow rules* (where calls after some waiting time are routed to new groups of agents), and *LIA/LWC*: on arrival a call is assigned to the *longest idle agent* with the right skill, on finishing a call an agent gets the *longest waiting call*, among those calls for which he or she has the right skill. Adaptive policies would adapt the priorities or the time until overflow, but this is rarely used in practice and little studied. On www.gerkoole.com/OBP a 2-skill calculator can be found by which static routing policies can be simulated. By varying the numbers and skills of agents figures such as Figure 17.7 can be constructed. It shows that the economies of scale and decreasing returns of increasing the level of multi-skilledness of agents. As costs increase approximately linearly in the number of multi-skilled agents there is an optimum in the number of multi-skill agents.

These simulations are purely based on fixed arrival rates and known (exponential) service time distributions. Note that call centers using SBR are less vulnerable to forecasting errors than call centers operating in a single-skill mode. Thus SBR not only reduces staffing needs related to short-term Poisson fluctuations, but also additional staffing or flexibility to deal with long-term fluctuations.

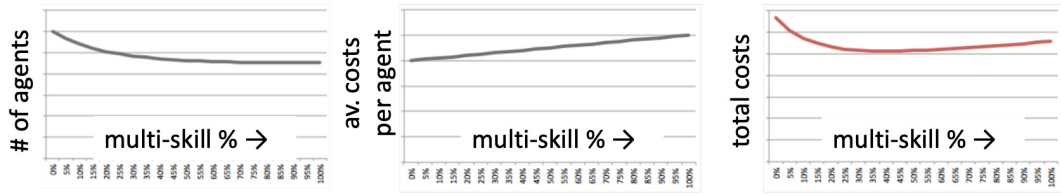


Figure 17.7: Economies of scale and costs for different levels of multi-skilledness

XXX NEW:

17.4 Agent scheduling

Agent scheduling is the operational process in which agents get assigned to shifts and activities during these shifts. Activities include the channel and/or skills they have to work on, but also paid breaks, meetings, training sessions, etc. Next to the routing, which is part of the telephony/omnichannel switch, it is the part of WFM that is most often supported by specialized software. There is a wide choice of software vendors, the bigger ones include Genesys, Injixo, Nice, Teleopti, and Verint. See for example [156] for a list. However, little is known about their exact workings. Erlang C and simulation are used, the latter often leading to very long run times. Fukunaga et al. [57] give some details about Verint (called Blue Pumpkin at the time). Smaller call centers, and also the ones with less scheduling issues (for example, because they are only open during business hours), often schedule using a spreadsheet. Agent scheduling in its generally is hardly studied in the literature: usually unpersonalized shifts are determined, without activities within the shifts, which is actually shift scheduling.

In its simplest form, agent scheduling consists of three steps: for each interval the required staffing levels is determined (e.g., using an Erlang formula), the most efficient way to cover the staffing needs by the available shifts is determined (potentially using integer linear programming (ILP)), and these shifts are assigned to agents in some way (for example, by letting them choose in the order of seniority). The first to formulate a solution for the covering problem of the second step was Dantzig in [48], in which he considered toll booths at a US bridge.

There are various reasons why such an approach is highly suboptimal and even infeasible. Often employees have different types of contracts, therefore in step 2 different groups of shifts should be identified, otherwise no match between agents and

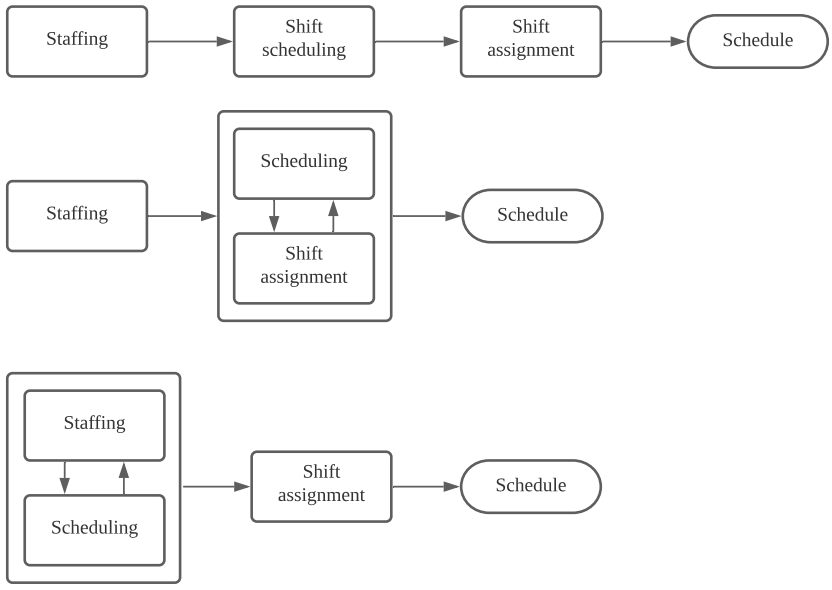


Figure 17.8: Short-term operational planning process

shifts can be made. Furthermore, many agents have personal preferences. Satisfying them as much as possible is crucial for employee satisfaction, making that the schedule should be made at the individual level, integrating step 2 and 3. Dealing with these personal preferences is evidently part of WFM software, but hardly studied scientifically.

A much better studied subject is the integration of step 1 and 2, as in the lower part of Figure 17.8. The reason for combining them is that the staffing levels of step 1 are hard to cover with shifts, leading to considerable overstaffing. Often SLAs are formulated at the daily level, thus SLs are allowed to fluctuate a bit, certainly if that leads to more efficient schedules and if the daily constraints are met. Integrating step 1 and 2 makes the optimization problem highly non-linear. It can be rewritten as ILP but at the cost of having many binary variables. Add to this the fact that we should schedule at the weekly level (necessary because of constraints on the schedules related to numbers of working days per week and start times), then we are stuck with heuristics such as local search. In multi-skill settings we need simulation to get reliable evaluation of possible solutions, leading us to simulation-optimization with stochasticity on the SL constraints, problems which are known to be notoriously difficult.

In a single-skill situation, Koole & van der Sluis [101] uses SIPP and shows that under a very simple shift structure, a suitable local algorithm can find optimal schedules. In a transient single-skill setting, Atlason et al. [11] use simulation to generate cutting planes used in the shift optimization module. Liao et al. [**?**] also use simulation. They combine stochastic programming and robust optimization to work out scheduling with uncertain arrival rates. Robbins & Harrison [**?**] solve a stochastic scheduling problem to minimize the combined cost of agents and missing QoS targets. Later, Gans et al. [**?**] considered a two-stage scheduling problem which allows adding and removing agents based on updated forecasts at midday.

In a multi-skill situation, Pot et al. [125] & Bhulai et al. [23] use an overflow approximation for SBR, similar to Chevalier & Tanbordon [37], to build a multi-skill scheduling algorithm. Bodur and Luedtke [**?**] solve a two-stage stochastic programming for scheduling with Benders decomposition. A main drawback of these approximations are unrealistic assumptions or unrealistic fluid approximations. Moreover, service levels cannot be approximated, because the models are based on rejection models. Again using simulation, Cezik & l'Ecuyer [34] extends the approach of Atlason et al. [11] to multi-skill staffing. Avramidis et al. [13] extends the cutting plane method to solve scheduling problem over a day (i.e., multiple periods). Running times however are very long.

The current state-of-the-art is Li et al. [106] which uses machine learning (ML) to speed up the simulations. This makes it possible to solve industrial-size weekly multi-skill multi-channel problems in several minutes. Solving the same problem without using ML takes much longer, see Li et al. [105]. Note that it is inevitable that the SL fluctuates, because of our transient daily SL objective. In call centers planners spend long hours adapting schedules manually to get smooth service levels, from a mathematical perspective a useless and expensive practice.

Note that all these problems consider *shift* scheduling: they determine shifts, but do not determine the activities within the shifts. This adds a layer of complexity far beyond the current state-of-the-art, but it is required in the operations and done by WFM software. On the other hand, it can be argued that the activity assignment should be done at the routing level, although some activities (such as meetings) need to be planned in advance. The fact that these methods are not at the level of agents but at best at agent group level makes them better suitable for capacity planning, which is the subject of the next section.

The big remaining challenge not yet addressed in the literature is the construction of even faster simulation-optimization methods or (meta)heuristics using very accurate approximations that solve the combined problem of shift scheduling and task

scheduling in several minutes. Most WFM systems either use very crude approximations based on Erlang C or have excessive run times of for example a whole night on a fast computer. These methods should work for a multi-skill multi-channel environment and take all shareholder interests (agent satisfaction, costs, and SL) into account.

17.5 Capacity planning

Capacity planning is the holy grail of WFM. To be able to do long-term planning, you have to take into account how you deal with all the shorter-term processes. Thus all decisions at all levels impact capacity planning. On the other hand, it does not have to be done at the same level: while agent schedules need to be determined at the 15-minute level, capacity planning can often be done at the week level.

Let us first consider capacity planning used for budget planning. The long-term forecast is an essential element for this process. Based on the forecast schedules could be made, just as for agent scheduling. Then the costs of these schedules could be determined leading to the budget. However, apart from some practicalities such as the lack of information on agents still to be hired, runs are often too long to compute the multi-year horizon needed for the budget, especially because Excel, which is used for this in 99% of the organizations, is not appropriate for this kind of calculation. A simple fast calculation is to estimate the budget proportional to the volumes: if the volume increase by x% then the costs will also increase by x%. Of course we make an error: costs are not linear in the forecast, but for small changes the error is expected to be small, probably much smaller than the forecasting error. More advanced methods, such as an ML model to estimate costs based on the forecast and other parameters, have also been successfully used in practice.

More complicated are decisions related to the hiring (and perhaps firing) of agents and decisions about the training of new skills for existing agents. Hiring and training new agents is a lengthy process that can easily take 3 months or more, thus the capacity has to be planned well in advance. In the simplest case it is just deciding how many agents are needed, but often there are choices in types of contract and initial skill sets. To determine which types of agents to hire shift scheduling has to be done, over a longer period, starting from the current pool of agents, taking agents resignations and *shrinkage* into account. Shrinkage is the term used for all activities that prevent agents from being available for phone work (or other types of contacts), from holidays and illness to meetings and paid breaks. The operational schedule should take activities like meetings and short breaks into account, capacity planning

all of them. Note that they are sometimes unpredictable, such as illness, and sometimes planable, such as when agents go on holidays or when meetings take place. Both types complicate capacity planning. Many call centers do capacity planning in a grossly simplified way by replacing all randomness and advanced calculations by fractions as explained in the previous paragraph on budget planning. Probably even more call centers use no calculations at all but make rough estimates, potentially making big errors in the optimal amount of agents and especially in the optimal contract and skill mix. Very few utilise more advanced technology, finding the optimal agents pool and determining which agents is the best to be added to the current pool is hardly done.

There are no papers solving the pool optimization problem completely. Some papers, such as [13, 23, 105], as discussed in the previous section, solve the shift scheduling problem for a week or a day, but methods have to be found to extend this to longer periods or to somehow aggregate weekly results to say a year. Furthermore, all forms of shrinkage have to be added. In our opinion, this is the biggest remaining challenge in WFM, and the only possible solution method we see is a time-consuming simulation-optimization procedure, possibly sped up using ML as in [106].

A simpler solution to the pool composition problem might be to use some rule of thumb. Chevalier et al. [36] studies, using approximations based on networks of overflow queues, that 80% specialized and 20% fully flexible agents works surprisingly well in many situations. This holds for the staffing problem, random forms of shrinkage will likely make the need for flexible agents higher in the pool composition problem. Also Wallace & Whitt [158] show, using simulations, that a little flexibility goes a long way, in a situation where agents have 1 or 2 skills and a topology that "connects" all skills.

17.6 Intra-day management

Intra-day management are changes made to the deployment of agents during the day of execution (or just before). These changes can be to the activities they do. Sometimes this is motivated by the SL: agent priorities can be changed, or for example meetings can be cancelled to improve the SL or even scheduled at the last moment when many agents are idle. The changes in activity can also have other motivations, such as the urgent need to schedule a meeting. At all times the consequences to the SL should be taken into account.

Next to changes in activity, intra-day management deals with changes in working hours. This starts as soon as the schedule is published by the planners, when for ex-

ample agents request schedule changes for personal reasons, or when the forecast has changed significantly and more or less agents are needed. This continues throughout the day of execution, many call centers have a flexible workforce layer through which they can up or downscale on a short notice, even during the day itself. The management of this is often not based on SL predictions, and also few papers address this type of issue. An exception is Roubos et al. [132] in which the staffing levels are adapted during the day in an optimal way as to obtain the required SL by the end of the day.

17.7 Design

We start this section with some general guiding principles on how the workforce should be planned.

Decisions that limit flexibility should be taken as late as possible. That way we can better deal with fluctuations, because for all types of fluctuations it holds that over time more information becomes available, i.e., the variability of the unknown variable decreases over time. E.g., take a multi-skilled call center. During the scheduling phase skills can be assigned to agents blocking them for other skills, unless traffic management changes the schedule. Letting SBR do the assignment is much more efficient, even a fixed assignment at the last moment is better because the latest forecast can be used, and availability is fully known, you known for example who is ill. A re-assignment could be part of intra-day management, but why then schedule in the first place?

An often-heard objection against SBR is that agents have to change skill (e.g., move from one language to another) frequently, which can be annoying. Similar objections hold against blending, especially when email handling is interrupted for inbound calls. Good routing however can avoid that: in certain systems you can limit the number of times that an email might be interrupted, and one can think of similar solutions for blending.

When the decision is related to something that influences employee satisfaction then making decisions later might be more efficient but at the same time decrease employee satisfaction which negatively impacts the performance of the call center. However, flexibility is not always required at the maximum level, asking only a fraction of the employees to be flexible might give you the majority of the advantages of flexibility, which is the next guiding principle.

A little flexibility goes a long way. Wallace & Whitt [158] observed this for the number of multi-skilled agents in an SBR setting, but this holds in general: a few

agents with part-time shifts, a few back-office agents who can help in the front-office (the call center), etc., can help to obtain the biggest part of the advantages of flexibility. Another way to state it is that flexibility shows decreasing returns. From this it also follows that you can better have a bit of multiple types of flexibility, than a large amount of one type of flexibility. However, using all these forms of flexibility together in the smartest way possible is a challenging task that requires appropriate tooling. Nobody can immediately grasp the consequences of one agent less on all skills, even the ones he or she does not have. That brings us to the final guideline.

Automated decision making is preferred over manual. Few decisions are fully manual or automated, for most decisions there is some tool (which can be a spreadsheet) that supports the decision. The better the tooling, the less human interference is required. We argue that a higher degree of automation is usually better: advanced knowledge in the form of algorithms can be implemented in software, knowledge that most planners will never obtain. An important constraint is that the outcomes should be transparent, for the planner to be able to explain the outcomes and interact with it. As an example, take an agent who wants the afternoon off. Usually an intra-day manager looks at the current SL, and makes a decision on the basis of that. It would be much better to have a SL prediction for the rest of the day to base the decision on, but that requires advanced tooling. From there it is a small step to an automated system that compares the predicted SL with the SLA and that makes a decision on that basis. Clearly this makes the call center more efficient but also makes the WFM team smaller, leading to additional costs savings which are usually higher than the software costs.

When designing a call center WFM is only one of the aspects that have to be taken into account, each decision should also be evaluated from the point of view of WFM: what are the consequences for the SL, the agent satisfaction, and the efficiency, i.e., the costs? Quite often WFM is not in the loop when design decisions are made. E.g., when the decision is made not to look at AHT anymore to allow agents to give the best possible service, what are the consequences on the required workforce if the AHT gets much longer? It might be customer-friendly to expand the opening hours, but what are the consequences for the SL? There are many questions of this type. The tools described in the previous sections can often be used to solve these problems.

OLD:

17.8 Improving workforce management

The standard approach to WFM leaves much to be desired. In the first place, scheduling at least the minimum number of agents in each interval gives an overall service level that is well above the minimum. This is for two reasons: in the first place because of the integral nature of the number of agents, in the second place because of the fact that the optimal way of covering the highly varying staffing requirements with shifts of fixed length (e.g., 4 or 8 hours) might introduce some slack. This calls for an approach that considers the average daily service levels, that allows intervals with a low SL to be compensated by intervals with a high SL. Note that we should account for the arriving rate when calculating the expected daily service level. Let λ_i be the arrival rate in interval $i \in \{1, \ldots, I\}$, $S_i(s_i)$ the SL as a function of the number of agents, then the expected daily SL S is given by

$$S = \sum_{i=1}^{I} \frac{\lambda_i}{\sum_{j=1}^{I} \lambda_j} S_i. \tag{17.5}$$

When shifting from interval SL constraints to overall SL constraints, we can profit from economies of scale in the following way. The offered load to a call center typically varies heavily over the day. This means that the agents-SL curve is much steeper during the busy hour than during less busy hours. This, together with the higher weighing factors in (17.5) for busy intervals, makes it interesting to overstaff during busy periods and to understaff during quiet periods. A small example with only two intervals is worked out in Table 17.1. By comparing the first two rows it can be seen that the service level improves considerably by shifting an agent to the busy interval, although the SLs in both intervals are less balanced.

s_1	s_2	TSF interval 1	TSF interval 2	overall TSF
13	3	89.51	95.33	90.04
14	2	95.41	76.12	93.66
13	2	89.51	76.12	88.29
13	1	89.51	0	81.37

Table 17.1: The effect of rescheduling; $\lambda_1 = 10$, $\lambda_2 = 1$, $\beta = 1$, $a = 0.333$, $\alpha = 0.8$

Table 17.1 also nicely illustrates the effect of the integer constraints for numbers of agents on the overall service levels. From the last line it can be shown that with two agents less we still satisfy the overall SL constraint.

Another drawback of the current WFM practice is the fact that steps three and four, determining shifts and coupling them to agents, are separated. As agents often have different types of contracts, resulting in different shifts, it is not possible to determine the right mix of shifts before actually assigning agents to them. The same holds for the personal preferences, such as employees which have to start at the same time due to car pooling, or agents which are available only part of the time due to obligations such as meetings. This calls for an integration of step three and four, determining the shifts and assigning them to agents.

In summary, we see that ideally the whole planning process should be done in a single step. From a model-solving point of view this is impossible. Of course there are simple call centers with a single type of shifts and no additional preferences, in such a case the standard approach can be followed. But in our opinion, for a complex call center, scheduling should start from the personnel preferences and the expected call volume, and should consist of a DSS in the real sense, that allows the human scheduler to evaluate changes made to the schedule.

A final way to improve WFM is using a more sophisticated model than the Erlang C. This will be discussed later.

17.9 Variability and flexibility

In Section **??** the lack of robustness of the Erlang delay system was discussed. Although it looks worse than it is in reality, thanks to the abandonments, measures need to be taken to be able to react to changes in load and changes in workforce availability. In practice the latter is usually done by introducing *shrinkage*: this is the difference between planned workforce and the workforce actually needed. It often includes also training, meetings, and so forth. A usual value is 30%. But evidently, the actual shrinkage is unpredictable, for example illness cannot be predicted the moment the schedule is made. Thus it rarely occurs that exactly the right number of seats is occupied. For this reason another solution is needed.

By introducing flexibility at all time levels of the operation we can offer the required SL while keeping a high productivity at the same time, independent of changes in parameters: it makes the call center more robust. At the highest level we have flexibility in contracts. With this we mean that for certain agents we can decide on a very short notice (e.g., at the beginning of the day) whether we require them to work or not. This is an excellent solution to deal with variability in arrival rate and absence. For the latter this is obvious; for the former we have to realize that the arrival rate during the first hours of the day often gives a good indication of the load during

the rest of the day. Thus early in the morning it can already be decided whether additional agents are needed.

When trying to quantify this, we start with a minimum number of fixed contract agent. This minimum is based on some lower bound on the arrival rate and a minimal absence. Then we assure that there are enough agents with flexible contracts such that we can get the number of agents equal to the number required in the case of a maximal arrival rate and maximal absence.

Example 17.9.1 A call center has an arrival rate that falls between 4 and 4.8, with 90% probability. For the lower bound 50 agents are needed, for the upper bound 9 more. Out of these 50 agents between 1 and 6 agents are absent, on average 3. Thus we schedule at least 51 agents, and in the "worst" case we have to hire 14 more, on average 6.

Introducing flexible contracts gives us the possibility to handle days with a higher than usual traffic load. If the peaks are shorter, in the order of an hour, then we cannot require agents to come just for this short period of time. In this it is possible to mobilize extra workforce by having personnel from outside the office work into the call center.

Example 17.9.2 Stocks trading lines of banks have scenarios in which many people from other departments can be mobilized, in case of for example a stock market crash.

A final type of flexibility is flexibility in task assignment. This is a method to react to load fluctuations that can even work at the finest level of fluctuations, that the Erlang formula accounts for. For this it is necessary that there are, next to the incoming calls, other tasks that have less strict service requirements. Examples are outgoing calls and faxes, and more recently emails and messages entered on Web pages. They have service requirements that range from hours to days, thus of a totally different scale than the requirements of incoming calls. To be able to satisfy the service requirements for these so-called channels it suffices to schedule just enough agents to do the work. Scheduling overcapacity, as for incoming calls, is not necessary. It also doesn't matter when outgoing calls or emails are handled, as long as they are handled in the required time interval. This makes it possible to use outgoing calls to fill in the gaps left by a low offered load, and allows in case of undercapacity agents originally scheduled for emails or outgoing calls to work on incoming calls. Thus instead of assigning in a fixed way agents to ingoing or outgoing calls, they are assigned dynamically (either by the supervisor or automatically) to a certain channel. This assignment should be done carefully. A free agents should obviously be assigned to

a waiting incoming call if any are present. A way to maximize productivity is by assigning free agents to outgoing calls if there are no waiting incoming calls. However, then every incoming call has to wait for a free agent. In most situations this will lead to a very low SL. The solution is to keep a number of agents free for incoming calls when none are waiting. This rule works when changing from ingoing to outgoing calls takes relatively little time. It is known as call blending, as it was originally intended for call center dealing with inbound and outbound traffic. Simply *blending* seems a more appropriate name given the recent focus on communication over the internet.

The advantages of most other channels compared to inbounds calls are clear. therefore the robustness and the SL are increased if inbound call are exchanged to emails or outbound calls. An active policy on this might reduce costs significantly.

Example 17.9.3 To make reservations for international travel the Dutch railways has two options. The first is calling the contact center by dialing an 800-number. The second is entering your travel data and the moment at which you want to be called back (a four-hour interval) on a web page. Potential travelers are thus financially stimulated to enter their data on the web page, thereby turning an inbound call into an outbound call. This allows the contact center to contact you at some quiet moment during your preferred time interval. Often the call takes little time as the agent already known the travel options, based on the data that you entered.

17.10 Multiple skills

Introducing a differentiation in skills has many managerial advantages. Training costs are lower compared to a call center where all agents should be able to deal with all calls, and the acquisition of new skills after some time offers call center agents a career path, that can help reduce turn-over. But there are also dangers related to the introduction of multiple skills. In the first place, it increases the complexity of the call center. Next, it has a lower flexibility compared to one big single-skill call center, and, finally, one might loose the economies of scale. To illustrate the latter point, let us look at the numbers in Table 17.2. Here we see a call center with two skills and a total of 24 agents, $\lambda_1 = \lambda_2 = 5$, $\beta = 2$, $a = 0.333$. The advantages of multiple skills but also of cross-training are obvious.

The results in Table 17.2 are obtained through simulation, except for the first and last line. This illustrates the lack of useful solution methods for these types of problems. In this section we give an overview of the problems and possible solutions.

skill 1	skill 2	skill 1 & 2	SL
0	0	24	84.7%
2	2	20	84.4%
4	4	16	83.8%
6	6	12	83.1%
8	8	8	81.4%
10	10	4	77.5%
12	12	0	67.8%

Table 17.2: The effect of cross-training: $\lambda_1 = \lambda_2 = 5$, $\beta = 2$, $a = 0.333$

In fact, there are two types of problems related to multiple skills. One is of a design nature, namely how many agents with certain skills are needed to obtain the desired service level for the skills. The other is an online control problem: to which agents to assign which calls. Both problems are extremely difficult, and only partial solutions and rough estimates exist in the scientific literature. We start by discussing online call routing, which is known as *skill-based routing*.

There are two types of skill-based routing: Static and dynamic. For both types of routing, each agent is member of one agent group. Groups are characterized by one or multiple skills that all agents in the group have (although it can occur that there are multiple groups with the same skills). Now if a call for a certain skill arrives, it is offered to one or more groups having this skill. The order in which this is done is determined by the skill-based routing. We call it the routing list.

Static routing means that the order in which calls are offered to groups is fixed and does not depend on current information, only on the call type. If all agents in all groups of the route are occupied, then the call is offered again to all groups in the same order, or, equivalently, to the first available agent within one of the selected groups. Of course the call center manager can change the routing at the beginning or even during the day; in practice we often see that the routing is changed only when groups are introduced or deleted.

No closed-form solutions exist for performance measures in static routing situations. Very good approximations exist if delayed customers abandon immediately. The delay case is still open. Also the scheduling problem is still open, although the approximation for the static case can be used as the basis of a local search algorithm. For certain standard situation it is observed in the literature that about 20% of cross-trained agents suffices to obtain most of the economies of scale.

Dynamic routing means that there is an online algorithm that determines how to route each call using current information such as availability of agents. A static algorithm is a special case of a dynamic algorithm; therefore dynamic algorithms have (in theory) a better performance. The dynamic routing algorithm depends on the numbers of busy agents in all groups. This state space description has therefore as many entries as there are skill groups. Thus the state space is, in general, high-dimensional. This makes that general methods to solve this type of dynamic decision problems cannot be used, due to the so-called *curse of dimensionality*. Approximation methods are currently being developed.

A type of multi-skill situation that desires separate attention is when different types of calls have different SL requirement. E.g., one might have B2B customer that require a faster answer than B2C calls; the same might hold for a group of premium customers, or sales calls require a faster answer than after-sales calls. In the case of a multi-skill operation one wants to pretect the SL of premium customers by reserving in some wat capacity for their calls. This can be done by placing cross-trained agents in the single-skill premium group. A more flexible way in which a better SL is obtained is by reserving a number of cross-trained agents to premium calls: if less than a certain number of cross-trained agents are available then regular calls are not assigned to multi-skill agents.

The numbers in Table 17.2 illustrated the gain obtained from cross-trained agents as compared to only having specialists, assuming that the call handling times were equal. However, often this is not the case: agents that receive multiple types of calls are often less efficient than agents that only receive a single type of call. This counterbalances the loss of economies of scale, certainly in the case of large call centers. Let us illustrate this with a numerical example.

Consider a call center with 2 skills. Specialists of either skill have $\beta = 5$, generalists have $\beta = 5.5$ (because they have to switch skill regularly). For $\lambda = 1$ for each skill we need or 15 generalists or 16 specialists, thus cross-training agents saves one agent. For $\lambda = 5$ these numbers are 62 and 60, and thus working only with specialists requires two agents less.

The implication of the numerical example is that, in big call centers, the advantages of skill-based routing are limited. However, this does not mean that call centers should only employ specialists: cross-trained agents increase the flexibility of the call center. This flexibility in task assignment (see Section 17.9) can be used in many different ways. Multi-skilled agents might be scheduled to work on different skills during the day, but sometimes it is used to deal with the seasonality of certain call types.

Example 17.10.1 An assurance company has a peak on travel insurance calls right after the summer, while at the same time there is a dip in call about housing assurances. This gives no problems because there is a group of agents that have both types of skills.

Cross-trained agents are also very useful in the case of unpredicted fluctuations. Peaks in the volume on one type of calls can be attenuated by decreasing the number of agents scheduled to work on other skills.

Example 17.10.2 Consider again two types of calls, with both $\lambda = 5$ and $\beta = 5$, and twice 30 agents. If they are all specialists, and $\lambda = 5.5$ for one of the skills, then the SL on that skill reduces to 54%. The average SL over both skills is 67%. Moving one or two agents from skill one to skill two (who therefore need to be cross-trained) moves both SLs to around 70%.

Evidently, the situation will become even more advantageous if peaks in one type of traffic are accompagnied by dips in other types.

17.11 Shift scheduling

Machines and other technical equipment is often constantly available for use within a company, with the exception of repair and maintenance (see Chapter 14). This is not the case for the prime resource in most modern companies: people. This chapter deals with mathematical models for the (optimal) employment of personnel.

Standard shift scheduling We consider problems with T time intervals. For every time interval t a number s_t is given representing the minimum number of employees needed in interval t. The simplest shift scheduling problem has a constant shift length of K intervals without (planned) breaks. It can be solved by the following simple algorithm. Let x_t be the number of employees that start working at time t. To determine a schedule that uses the minimal number of employees one takes $x_1 = s_1$ and for $t = 2, \ldots, T$ x_t such that there are at least s_t employees working:

$$x_t = \max\{0, s_t - (x_{t-K+1} + \cdots + x_{t-1})\},$$

where we assumed for convenience that $x_t = 0$ for $t \leq 0$.

For several reasons this situation hardly ever occurs. Usually we encounter more complicated shifts (with for example scheduled breaks) and variations in shifts (e.g., shifts with different lengths). We formulate a mathematical programming formulation of this problem. Let K now be the number of different possible shifts, and x_k

the number of of people that are scheduled for shift type k. We define the constraint matrix A by $a_{tk} = 1$ if shift k works during interval t, 0 otherwise. Let c_k be the costs of shift k. Then an optimal schedule can be obtained from:

$$\min \left\{ \sum_{k=1}^{K} c_k x_k \;\middle|\; \begin{array}{ll} \sum_{k=1}^{K} a_{tk} x_k \geq s_t, & t = 1, \ldots, T \\ x_k \in \mathbb{N}_0, & k = 1, \ldots, K \end{array} \right\}. \tag{17.6}$$

Several solution methods exist for this type of integer programming problem. Note that the integer constraint is essential, without it non-integer solutions can be found (see Exercise 17.19).

This formulation leads to an optimal choice of shifts. This choice however is often infeasable in practice, due to limitations in the possible shifts. For example, an additional constraint could set the number of full-time shifts equal to the number of full-time employees. Many other types of constraints can be thought of.

Shift scheduling with a global constraint Previously we assumed that there is a separate constraint for each interval: in interval t at least s_t employee should be scheduled. In certain situations however the situation is different. Then the number of employees is allowed to be lower than s_t in certain intervals as long as this is compensated for in other intervals. This is for example the case in call center scheduling, if we take the daily average service level as objective. Then a low SL in certain intervals can be compensated for by high SLs in others. This allows for more flexibility when scheduling, leading to cost reductions.

Indeed, scheduling at least s_t employees in interval t gives an overall service level well above the minimum. This is the case for two reasons: in the first place the integral nature of determining s_t, in the second place the fact that the optimal solution of (17.6) might have $\sum_{s=t-K+1}^{t} x_s > s_t$ for certain t. This calls for an approach that considers the average daily service levels, but research on this type of solution is still in its infancy. Such an approach would integrate the second and third step of the decision process, namely determining the minimum levels s_t and determining the shifts.

To formalize this, let $L_t(s)$ be the service level at interval t if there are s employees working. Assume that $L_t(s)$ is increasing in s, this is for example the case if L is the percentage of customers waiting shorter than a seconds. We pose a restriction on the overall service level, which is defined as the weighted average of the interval service

level. With weighting factor w_t for interval t, this gives:

$$\min \left\{ \sum_{k=1}^{K} c_k x_k \;\middle|\; \begin{array}{l} \sum_{t=1}^{T} w_t L_t\Big(\sum_{k=1}^{K} a_{tk} x_k \Big) \geq \alpha \\ x_k \in \mathbb{N}_0,\ k = 1, \ldots, K \end{array} \right\}. \tag{17.7}$$

Example 17.11.1 In call centers (see Chapter 17) we consider the service level of an *arbitrary* customer. Customer calls arrive in interval t with rate λ_t. With probability λ_s/λ (with $\lambda = \sum_{t=1}^{T} \lambda_i$) this call arrives in interval s, and thus $w_t = \lambda_t/\lambda$. The service level L_t is given by the Erlang formula or one of its generalizations. To avoid understaffing we probably want to add constraints of the form $\sum_{k=1}^{K} a_{tk} x_k > \lambda_t \beta$, with β the average service time.

Problem (17.7) is non-linear, as L_t is in general non-linear. (In the case of tail probabilities, L is concave.) This calls for solution methods that are based on other techniques than branching to non-integer values. Such a technique is local search, where shifts are added, deleted and shifted until a local minimum is found. Numerical results show that this can lead to significant cost reductions.

Another option is adding additional variables to make (17.7) linear. Introduce the variables n_{ts}: $n_{ts} = 1$ if during interval t exactly s employees are scheduled, 0 if this is not the case. Thus, for every t, exactly one n_{ts} should be one. This is obtained by requiring $n_{ts} \in \{0, 1\}$ and $\sum_{s=0}^{S} n_{ts} = 1$ for every t, with S the maximum number of employees that can be scheduled.

Problem (17.7) is now equivalent to:

$$\min \left\{ \sum_{k=1}^{K} c_k x_k \;\middle|\; \begin{array}{ll} \sum_{k=1}^{K} a_{tk} x_k = \sum_{s=0}^{S} n_{ts} s, & t = 1, \ldots, T \\ \sum_{t=1}^{T} w_t \sum_{s=0}^{S} n_{ts} L_t(s) \geq \alpha & \\ \sum_{s=0}^{S} n_{ts} = 1, & t = 1, \ldots, T \\ x_k \in \mathbb{N}_0, & k = 1, \ldots, K \\ n_{ts} \in \{0, 1\}, & t = 1, \ldots, T,\ s = 1, \ldots, S \end{array} \right\}. \tag{17.8}$$

Note that this problem is linear in all its variables.

Assigning shifts to agents Making shifts is not the end of HR planning: shifts have to be assigned to people. The first requirement to make this possible is that the right numbers of the right shifts have been generated. In the total pool of employees there are often different types of contracts, which differ for example in the number of working hours per day. The shifts should be generated in the right numbers. This can be

ontained by adding constraints of the form $\sum_{k \in \mathcal{K}} x_k = N_{\mathcal{K}}$ with $\mathcal{K} \subset \{1, \ldots, K\}$ a set of shifts and $N_{\mathcal{K}}$ the number of shifts of this type that should be generated. $\mathcal{K}$ could for example be the set of 4 or 6-hour shifts start between 8.00 and 10.00.

Thus, for a single day shifts can be assigned to agents taking all constraints into consideration. However, the shift assigned to a particular agent on one day might influence the possible assignments on another day. For example, an agent might work 4 days per week, on days to be determined by the call center. Or an agent works 32 hours a week, assigned in a flexible way to the days. To deal with this scheduling sometimes needs to done at the agent level for a whole week at once. Work during weekends sometimes even leads to planning periods that are longer than a week. For example, in a certain industry sector in Holland there used to be a regulation that employees should have at least 4 free Sundays in every block of 13 weeks. To utilize fully the possibility of scheduling employees on Sundays the planning period should be 13 weeks in this case.

In practice we encounter situations where shift scheduling and shift assignment are completely separated, and situations where they are completely integrated. The former is computationally less demanding but gives suboptimal or even unfeasable solutions in many cases; the latter is computationally much more demanding, certainly if the planning period is long and if the intervals are short, i.e., T is big.

So far we described a way for an organization to plan its workforce given all constraints. A completely different approach, with its own advantages and disadvantages, is to have the employees choose their own shifts. Then the mathematical analysis stops after shift generation, and an especially designed (web-based) user interface allows the employees to choose their own shifts.

17.12 Long-term planning

Over a longer time period companies have to deal with changing need for employees and with *attrition*, the fact that people leave the company. The *turnover*, the ratio of new hires to the number of employees, can be well over 100% in for example call centers.

The challenge is the fact that offered load and turnover is unpredictable, and that hiring and training of new employees costs time. This requires stochastic models to determine the right moment to start hiring new employees. For costs reasons new employees are trained in groups.

TO ADD:

the WAPE is linear in the intra-day management costs ([52]). intra-day management is always needed, see Roubos.

17.13 Further reading

NEW:

This chapter is based on [97]. We start with some general call center WFM references. The following are academic overviews: Gans et al. [61], Avramidis & l'Ecuyer [14], and Akşin et al. [5]. More practitioner-oriented text books are Cleveland & Mayben [40] and Koole [100].

There is a huge literature on call center forecasting. Ibrahim et al. [79] is a recent overview.

Halfin & Whitt [71] introduced square-root staffing for Erlang C which was later extended to Erlang A and many other models. Seminal work on models with abandonments was done by Palm [122] and Baccelli & Hébuterne [16]. Zeltyn & Mandelbaum [166] give simple formulas for the $M|M|s+G$ using integrals over the patience distribution. For definitions and ways to compute the SL see Jouini et al. [84]. Sze [148] includes retrials.

OLD:

Brown et al. [29] contains a statistical analysis of data from one particular call center. All aspects of workforce management are discussed in Reynolds [128]. Fukunaga et al. [57] presents an overview of the scheduling modules of one of the major wfm systems. A more mathematically oriented text book that include the basic models but also some advanced skill-based routing models is Stolletz [146]. The underlying ideas of Section 17.9 hold for services in general, see Sasser [137].

Predictive dialers, in use in outbound call centers, are discussed in Samuelson [135]. The formula for quantiles of hypoexponential distributions comes from Ross [130, Section 5.2.4], the derivation resembles that of Koole [98].

Call center literature on shift scheduling, Ryan (INFORMS talk) on crew scheduling, Ortec tools.

17.14 Exercises

Exercise 17.1 Prove Equation (17.1).

Exercise 17.2 Reproduce the numbers of Example 17.3.1. Use rounding and do the same.

Exercise 17.3 Model blending inbound and outbound with equal average handling times and without abandonments as a birth-death process. Solve the b-d process and reproduce the numbers of the online calculator.

OLD:

Exercise 17.4 An important aspect of call centers are abandonments by people who do not want to wait any longer in queue. What do you think that is the influence of abandonments on the service level of the customers? And on the productivity?

Exercise 17.5 Consider a call center with on average 1.5 arrivals per minute, an average service time of 5 minutes, and 10 agents. The Erlang C model is used to compute the performance of this call center.
a. Compute the expected waiting time.
b. Give examples of an increase in scope and an increase in scale in the context of call centers.
Suppose the call center doubles in arrival rate and in number of agents.
c. Compute the expected waiting time.
d. How many agents do you need to make the average waiting time less than 90 seconds?

Exercise 17.6 A service level definitions avoiding some of the disadvantages of both the regular SL definition ($\mathbb{P}(W_q \leq t)$) and the average speed of answer ($\mathbb{E}W_q$) is the expected excess: $\mathbb{E}(W_q - t)^+$. Show that, for the Erlang C model, it is given by the following formula:

$$\mathbb{E}(W_q - t)^+ = \frac{\beta C(s,a)\mathrm{e}^{-(s-a)t/\beta}}{s-a}.$$

Exercise 17.7 Consider a call center where every agent that becomes available is assigned to an inbound call if one is present, otherwise to an outbound call. In other words: agents are never idle, inbound calls have priority over outbound calls of which there is an infinite supply. Assume that inbound and outbound calls have the same exponential service times.
a. Construct a birth-death process with as state the total number of calls in the systems (i.e., outbound calls in process and inbound calls). Give the transition rates.
b. Determine the stationary distribution.
c. Give a formula for the waiting time distribution for inbound calls.
d. Give a formula for the fraction of time that the agent is busy with outbound calls, and the average number of finished outbound calls per unit of time.
e. Compute these numbers for $s = 12$, $\beta = 3$ and $\lambda = 2$, 3, and 3.75.

Exercise 17.8 An Erlang calculator can calculate the service level, defined as the percentage of callers that waits longer than t seconds, based on the arrival rate λ, the average service time β, and the number of servers s.
In a call center there are on average 10 calls per minute, that require each on average 3 minutes to answer. The acceptable waiting time is 20 seconds, and the time between the moment a call is assigned to an agent and the moment it is answered by the agent is around 3 seconds.
a. Give the parameter values for the Erlang calculator by which you can calculate the service level in the call center.
b. To obtain a service level of around 80% 35 agents are needed. Define productivity and calculate it.
c. A model is not an exact description of reality. Give 3 aspects in which the Erlang system does not model the call center exactly.
d. The arrival rate doubles to 20. The manager decides to double the number of agents. What do you expect to be the consequences for the costs and the service level? Motivate your answer!
e. Estimate without using the Erlang formula how many agents need to be scheduled to obtain a 80% service level. Motivate your answer.

Exercise 17.9 Consider a call center with the following parameters: $\lambda = 5$ per minute, $\beta = 3$ minutes. An acceptable waiting time is 18 seconds. Answer the following using the Erlang C calculator:
- What is the service level with 20 agents?
- How many agents agents are needed for a 95% service level?
- How many are needed if λ doubles?
- And what if the acceptable waiting time is only 9 seconds?

After each call the agents need on average 1 minute to enter data related to the call and they take on average 5 seconds to take up the phone. Answer the same questions as above for this new situation using the Erlang calculator. Note that the time that an agent uses to take up the phone is both waiting and service time!

Exercise 17.10 Consider a call center with the following parameters: $\lambda = 5$ per minute, $\beta = 3$ minutes, 18 agents. An acceptable waiting time is 20 seconds. Answer the following using the Erlang X calculator:
a. Without blocking or abandonments, what is the SL?
Customers abandon, on average after 2 minutes.
b. Without blocking, what is the SL?
The number of waiting places is limited to 8.

c. What is now the SL?
If blocking plus abandonment must be less than 5%, and the SL as high as possible, how many lines would you choose?

Exercise 17.11 Agents are absent (e.g., because they are ill) with probability 0.05. Assume that $\lambda = 10$ and $\beta = 3$.
a. How many agents do you need to have behind the telephone to assure a 80-20 service level (i.e., 80% answered within 20 s)?
b. For s agents, what is the probability that more than k of them are ill?
c. How many agents should you schedule, not knowing who will be ill, to have 95% probability that enough will show up to meet the service level?
d. How many agents with flexible contracts and with fixed contracts would you schedule? Explain. (Note that flexible contracts are somewhat more expensive that fixed contracts.)

Exercise 17.12 Consider a call center with on average 2.5 arrivals per minute, an acceptable average waiting time of 1 minute, and an average service time of 6 minutes. The Erlang C model is used to compute the number of agents.
a. Compute this number using the table.
It is observed that the service time consists of 2 minutes talk time and 4 minutes so-called wrap-up time. Two minutes of this wrap-up time needs to follow the call; the remaining 2 minutes can be done at another time, by another agent. Because of this two agent groups are created: one that handles calls (average service time 4 minutes) and one that does only the second half of the wrap-up (average service time 2 minutes).
b. Compute the number of agents needed in the first group to obtain an average waiting time of less than 1 minute.
c. How many agents are needed in the second group to have a 100% productivity?
Agents like to finish a call completely if possible. It is decided to implement this in the following way. All agents are in 1 group, and they handle a call entirely if the there are few calls waiting. When there are many calls waiting then only the first part of each call is done. The remaining second parts are distributed among free agents later on when the load is lower.
d. How many agents do you expect to need under this new situation? Motivate your answer!

Exercise 17.13 Consider a call center with 2 types of calls, A and B. Arrival rates are λ_A and λ_B, average service times β_A and β_B. There are specialists of both skills and

generalists available.
a. Take $\lambda_A = \lambda_B = 10$, $\beta_A = \beta_B = 3$. There are 20 generalists. How many specialists with skills 1 and 2 would you schedule to minimize costs under a 80-20 SL? Motivate your answer. (An exact calculation is not necessary, a motivated estimation is sufficient.)
b. The same question for $\beta_A = 2$ and $\beta_B = 3$.

Exercise 17.14 A call center has 12 agents, calls arrive at a rate of on average 15 per minute, and the average call duration is 25 seconds.
a. Calculate the expected waiting time using the Erlang calculator.
It was found that this does not match with reality. Further research showed that it takes on average 5 seconds before an agent pick up the phone after a call is assigned to an agent.
b. Calculate again the expected waiting time of an arbitrary call.
Again, there is a considerable difference between reality and the prediction of the model. Further research showed that agents take short breaks, totaling on average 5 minutes per hour.
c. Give an approximation for the expected waiting time for this new situation.

Exercise 17.15 Prove Equations (17.3) and (17.4).

Exercise 17.16 A company is changing its call centers operations. Instead of a regional approach (there are currently 3 regional call centers), they decide to build a single call center with skill-based routing (i.e., different groups of agents handling different types of calls). A model for the new call center is needed to determine the consequences for the workforce. Based on the model outcomes decisions are taken with regard to the possible employment of new agents and with regard to the training of agents for specific skills.
The average call duration β is 2 minutes, independent of call center and call type. In the next table the number of agents in the regional call centers and the expected waiting times during peak hours are given. The arrival rates are not known.

Call center	1	2	3
Number of agents	12	9	7
Expected waiting time (seconds)	27	78	5.5

a. Determine the arrival rates, using the Erlang calculator.
There will be 2 call types, approximately 1 out of three calls is type 1. The average waiting time for both types should not exceed the current overall waiting time.

b. Determine the current average waiting time.
c. Determine the right number of agents for the two types.
d. How many agents would be needed in the case of only one single group?

Exercise 17.17 Answer the following questions using the multi-skill online simulator at gerkoole.com, using the following parameters: $\lambda = (5, 10)$ per minute, $\beta = (3, 3)$ minutes, patience $= 5$ minutes, AWT $= 20$ seconds, 29 single-skill agents with skill 2, 20 multi-skill agents.
a. What is the SL under the regular LIA/LWC routing policy?
b. Give a routing policy that obtains an 80% SL for both lines.

Exercise 17.18 Consider the same parameters as the previous exercise, but now the number of agents has to be determined.
a. Agents can have skill 2 or both skills. Single-skill agents cost 20 Euro per hour, multi-skill agents 23 Euro. What is the optimal combination of agents to obtain an 80/20 SL on both lines?
b. Now it is also possible to train agents only for skill 1. Their costs are 22 Euro per hour. What is now the optimal combination?
c. Compare the costs with the single-skill situation, computed using the Erlang A model.

Exercise 17.19 Consider a simple shift scheduling problem with $T = K = 3$, and $a_{tk} = 0$ if $t = k$, 1 otherwise, and $c_k = s_t = 1$ for all k and t.
a. Find the optimal schedule by solving the problem of Equation (17.6).
b. Do this again but without the integer constraint of Equation (17.6).

Exercise 17.20 A call center has inbound calls and emails. Every interval, agents either work on inbound calls or on emails. Emails should be answered within 24 hours. Suppose there are, over the whole day, u agent hours of work on emails to be done.
a. Extent the problem of Equation (17.6) to allow for emails that can be done by all agents.
b. Same as a, but assume that only agents doing shifts $1, \ldots, K'$, with $K' < K$, can do emails.
Make sure that the constraints remain linear.

Exercise 17.21 Consider a call center that is to be operated during 5 periods. There are three possible shifts, each working 3 consecutive intervals. The arrival rates are 5, 8, 10, 7, and 3 per minute. The average service time is 3 minutes. As standard 80-20

SL is chosen, performance is estimated using the Erlang C.
a. Compute a schedule that minimizes the number of agents if the SL has to be met every interval.
b. Compute a schedule that minimizes the number of agents if the SL has to be met on average over the whole day.

Exercise 17.22 A call center needs 100 employees. The turnover is 120%, equally spread out over the year, otherwise unpredictable. Employees leave on a very short notice.
a. Formulate a model for the attrition.
b. Hiring new employees and training them takes 2 months. At which moment should the call center start hiring to make sure that the probability that there are less than 95 agents is less than 5% the moment new agents become available?

Chapter 18

Revenue Management

The central theme of this book is optimally matching demand and supply. We have seen application areas with a fixed capacity where production was done before demand occurred (manufacturing) and after the moment demand occurred (elective health care). We've also seen that in certain areas capacity is adjusted to the demand (call centers and emergency health care). In this chapter we study systems where the match is made by adjusting the price of the product(s). Areas where this has proven to be successful include aviation, hospitality (i.e., hotels), and car rental. This activity including its methods is called *Revenue* or *Yield Management* (RM).

This method is based on the assumption that demand will decrease when the price goes up. We start this chapter with a short section on *pricing* in which we study the relation between price and demand. In the following section we consider the concepts of RM. The crucial difference between pricing and RM is that RM takes also the capacity into account, which is fixed and finite in the case of RM. After having obtained an understanding of the concepts we move to the technical details, first to newsvendor-type models, and then to models where the price can be changed dynamically over time. This requires both advanced forecasting and advanced optimization techniques, which are discussed in separate sections. We finish by looking into problems where multiple types of capacity are reserved simultaneously, which is not uncommon in aviation (where multiple flight *legs* constitute a single trip) and hotels (where customers often stay multiple nights).

18.1 Pricing

Consider a certain product. The (expected) demand D is (usually) a decreasing function of the selling price p. D is called the *demand curve*. We are interested in finding the price for which the revenue is maximized. Define

$$E(p) = \frac{pD'(p)}{D(p)} = \frac{p}{D(p)} \frac{dD(p)}{dp},$$

the relative difference in demand ($dD(p)/D(p)$) divided by the relative change in price (dp/p). It is called the *price elasticity*. We assume that $E(p) < 0$: when the price is higher, less items are sold (there are some exceptions, for example luxury items such as Gucci bags). When $E(p) \in (-1, 0)$, then an increase in price will lead to a relatively small decrease in demand, thereby increasing total revenue. When $E(p) < -1$, an increase in price will decrease revenue. The revenue is maximized if $E(p) = -1$: when the price increases by 1%, then the demand decreases by 1%, keeping the total revenue equal. The condition $E(p) = -1$ can also be obtained by differentiating the total revenue $pD(p)$ to p.

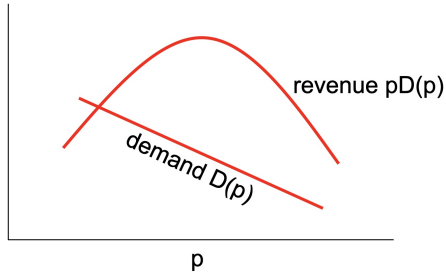
The difficulty in applying the above is that it is often very hard to estimate the form of the demand curve. In few situations it is possible to experiment with different prices in such a way that a reliable estimate of D can be obtained. Note that demand is a *censored* variable: when the price is p, you only observe the part of the demand willing to pay at least p. Furthermore, changes in demand can also be explained by other factors, such as the day of the week, prices of competition, etc. This means that a difference in sales can not always be contributed to the fact that different prices are used. Ideally, a statistical or machine learning model should be made which explains the demand as a function of price and all other variables that influence demand. However, to get a reliable model many data points are required, together with enough variability in price.

Example 18.1.1 For a certain parking lot at an airport it is possible to make advance internet reservations. By changing the price from day to day it is possible to estimate the demand curve and to determine the price that maximizes the revenue. This needs to be made for comparable days: for example weekends and special days (such as holiday) have a different demand.

Choosing p such that $E(p) = -1$ maximizes revenue. That is not the same as maximizing profit because of the costs. The total production costs C are a function of the number of produced items $n \in \mathbb{N}_0$. There are fixed costs $C(0)$, variable costs

Linear demand curve

The best we can obtain often consists of estimates of $D(p)$ for 1 value of p and its derivative $D'(p)$ for the same value of p. It makes sense to assume that D is linear for values close to p. Then the revenue $pD(p)$ is quadratic. See the figure below for an illustration. The top of the parabola is the point with $E(p) = -1$.



$C(n) - C(0)$, and marginal costs $\Delta C(n) = C(n+1) - C(n)$. When the total costs are linear then $\Delta C(n) = (C(n) - C(0))/n$ for all n, the marginal costs equal the average variable costs.

Example 18.1.2 Consider the rooms in a hotel for a particular night. n is the number of occupied rooms. In hospitality it is not unreasonable to take $C(n)$ linear. $C(0)$ consists of the investment in the building, and staff that has to be there no matter how many customers there are. The marginal costs consist for example of cleaning costs. Actually, the marginal costs might be negative, because visitors, even though they pay for the room, generate additional income, for example by dining in the hotel's restaurant.

The profit W under a price p can be calculated as follows:

$$\text{profit} = W(p) = \text{revenue} - \text{total costs} = pD(p) - C(D(p)).$$

When the variable costs are low, then maximizing revenue is (almost) equal to maximizing profit. When the costs are non-negligable but linear, then $E(p) = -1$ should be replaced by $(p-c)D'(p)/D(p) = -1$, with c the variable costs.

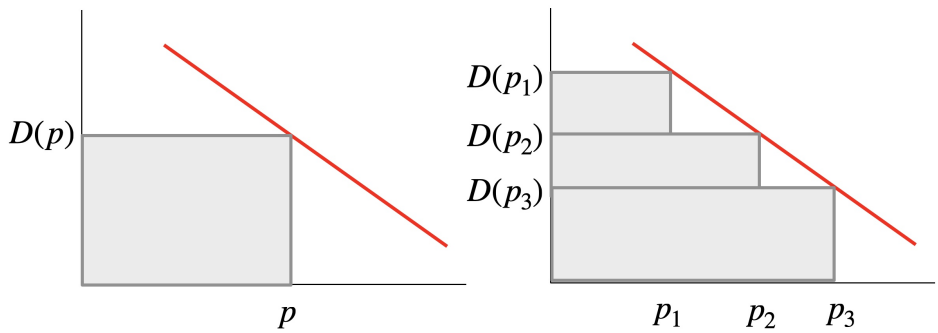
In the rest of the chapter we will focus on revenue, because in the most relevant application areas variable costs are negligible or linear otherwise, in which case we replace the price p by $p - c$.

Market segmentation So far we focused on a single price, i.e., everybody pays the same price. However, different customers have a different *willingness-to-pay*: otherwise the demand curve would be a step function with a single step. From the seller's

point of view it would be ideal if all customers paid their willingness-to-pay. This is the goal of *personalized pricing*, but in practice it is still very hard to achieve. What can often be done to get customers pay closer to their willingness-to-pay is *segmentation*. Suppose that a product is sold to different groups of customers, for example through different channels. Then, instead of selling the product for a single price, we could differentiate in price for the different segments of the market. Now for every segment the revenue should be maximized, leading to a higher total expected revenue than for a single price. A major concern is *buy-down*, when customers from one segment, willing to pay the price for that segment, buy the product for the price targeted at another market segment.

Example 18.1.3 Consider again the parking lot of Example 18.1.1. There are internet reservations and customer who drive up without reservation. The internet price is lower than the drive-up rate, the idea being that cost-sensitive customers take the trouble of going online to search for bargains. The numbers of parked cars are carefully monitored to make sure that the number of drive-ups does not decrease.

Example 18.1.4 Suppose we have a linear demand curve as below. In the left figure the case without segmentation is shown, the surface of the rectangle is the revenue $pD(p)$. Now we move from 1 price point to 3, and assume no diversion: everybody willing to pay between p_2 and p_3 pays p_2, and everybody willing to pay p_3 or more pays p_3. Thus $D(p_3)$ customers pay p_3, $D(p_2) - D(p_3)$ pay p_2, and $D(p_1) - D(p_2)$ pay p_1. The shaded area in the right figure shown again the total revenue. We clearly see the increase in revenue.



Example 18.1.5 With a student card you can a get a 10 Euro ticket for what is considered by many the world's best orchestra, the Amsterdam Concertgebouworkest. For many students a regular ticket is too expensive, thus this is a form of segmentation. But this way they also hope to attract future customers paying the regular price.

Example 18.1.6 In aviation segmentation is done in many ways. The best known form of segmentation is between business and leisure. Usually business customers have a higher

willingness-to-pay. Business flyers typically fly during the week and have short stays. Therefore, to get a cheap ticket on a regular airline, requires you to stay over the weekend and/or multiple days. Evidently some business customers stay during the weekend or longer periods, allowing them to get a cheaper ticket than they are willing to pay. Thus the segmentation is not perfect. Note that this is not about the difference in cabins, often called business and economy. They are usually priced differently and seen as different products.

Up to now we considered pricing in the context of a potentially infinite supply of products. When the number of products or underlying resources is limited, below the demand for the revenue maximizing price, and consumption cannot be delayed, then *revenue management* comes into play.

18.2 Revenue management concepts

In the literature different definitions of RM can be found. Two of them are as follows:

> "RM is the art and science of predicting real-time customer demand at the micromarket level and optimizing the price and availability of products" (Cross [45], p. 4),

and

> RM "is the process of understanding, anticipating and influencing consumer behavior to maximize yield or profits from a fixed, perishable resource" (Mauri [111]).

Combined they contain the most important elements of RM: *micromarkets* (i.e., segmentation), price optimization, and a perishable and fixed capacity. Whether it is a process, science or art is not clear: I would say it is a methodology based on scientific principles.

The crucial difference with pricing is that the product is perishable and its capacity fixed. Perishable products have the property that production and consumption must happen close to each other. Examples are cooled dairy products and meat. In the current context it is usually meant that production and consumption happen at the same time, i.e., it concerns a service. Indeed, a seat in an aircraft can also be seen as a perishable product: when the aircraft has left all empty seats are "perished".

RM is typically used in the situation of a service with a fixed capacity, such as hotels and aircraft. When capacity is flexible, then the solution to variable demand is adapting the capacity, as in call centers. Fixed capacity also implies high fixed costs,

the investments in aircraft and hotels are a major part of the costs in aviation and hospitality.

In aviation a standard figure shown in business reports is the *passenger load factor*, i.e., the number of occupied seats divided by the total number of seats "flown". However, a high load factor does not mean a high revenue, or v.v. The price at which the seats are sold is equally important. The objective of revenue management is to maximize revenue by deciding regularly if or how many seats are available at which price.

We discussed the main concepts segmentation, fixed and perishable capacity, and revenue maximization. There are a number of concepts we still need to discuss: *demand diversion*, *auxiliary items*, *overselling*, *overbooking*, and *multi-leg/night bookings*.

To discuss these items we have to define clearly what a product is. A product is defined by the use of a single unit of capacity at a certain point in time. Thus two tickets on the same flight of which only one is refundable concerns the same product but with different characteristics. Two tickets on different flights are different products.

Diversion is the fact that increasing the price of a product will decrease its sales and increase sales on other products. For example, when ticket prices are high on Friday night people will take an earlier or later flight. This is a desirable feature: probably the demand on the late Friday flight is higher than of other flight and this way people willing to pay the high price get on the flight and other may divert to a cheaper flight. If done well this increases the load factor and the total revenue. It should be avoided that people with a high willingness-to-pay divert by making the other flights not too cheap. From a technical point of view diversion complicates the analysis, demand of different products is not independent.

Auxiliary items play an increasingly important role in aviation. They are characteristics of products such as luggage allowance, conditions for a refund or rebooking, and priority boarding. Traditionally it is a way to segment: flexible tickets were priced higher, and were supposed to be preferred by business travellers, who need the flexibility. Then the conditions are attached to the *booking class*. However, more and more we see that airlines offer these auxiliary items separately, on top of the ticket price. This reduces their possibility to segment.

Whether sold separately or as part of the ticket characteristic, some tickets offer the possibility to rebook or even to get a full refund. In hospitality there is often the possibility to cancel the reservation. This means that less customers will show up than the number booked. To avoid consistently having empty rooms or seats more bookings than there is capacity should be allowed. We will call this *overbooking*.

Bundling

Bundling is a method in which the auxiliary items are not sold separately but in *bundles*. Bundling increases revenue. For example, instead of selling a luggage allowance and priority boarding separately for each 10 Euro, a bundle of both could be sold for 15 Euro. Suppose there are customers, one which values additional luggage at 10 Euro and priority boarding at 5, and the other the other way around. Then bundling leads to a revenue of 30 Euros instead of 20. It is possible to construct examples where bundling does not increase revenue. Consider for example another 2×2 problem with the following parameters:

		item	
		1	2
customer	1	30	5
	2	5	50

Unbundled the maximal revenue is 80, bundled it is 70. However, it is observed to work well in practice and most airlines do it.

Note that although all customers who show up are expected to fit, there is a positive probability that they won't. At first sight this probability can be computed using a binomial distribution, but the impact is worse because whether customers show up or not are correlated events. The consequences of more people showing up than there is capacity is denied boarding for some people. Often this is voluntary, denied passengers get a financial compensation for taking a later flight, but sometimes it has dramatic consequences, as with United flight 3411 on April 9, 2017, when a man was beaten and dragged out of the aircraft by security officers.

Example 18.2.1 A flight has 100 seats. 65 customers have non-reimbursable tickets, they all show up. 40 tickets are flexible, these customers show up with a 80% probability. The expected occupancy is 97%. If the events of them showing up are independent then with 4% probability there is not capacity. However, 80% is the overall average: on certain flights the probability is higher (for example, because there are less alternatives or because it is the last flight), on others lower. Suppose a certain flight has a 70% probability on one day, 90% on another day. When the probability is 70% there is ample capacity, it never happens (once in a 1000 times) that the capacity is not sufficient. When the probability is 90% however then almost half of the times the capacity is not sufficient. Thus when the airline does not know when its 70 or 90 or does not take it into account, and it happens both roughly half of the times, then there is not enough capacity is around 25% of the cases, a huge difference with 4%.

Cancellations force airlines and hotels to overbook but the remaining uncertainty

is a risk for which they pay in lost capacity or financial compensations to denied customers. Therefore a non-reimbursable ticket can be worth more than a higher-priced flexible ticket, especially when the chance of a late cancellation is high.

Although many people use the terms interchangeably, we will mean by *overselling* something else then overbooking. With overselling we mean the practice that airlines sell more tickets than they expect that will fit, meaning that they have to deny boarding to some customers. This practice is only profitable when the difference in ticket prices is higher than the compensation paid to the denied customer. A second reason for doing this is customer retention: customers with loyalty cards of a certain level are guaranteed to have a seat on any flight, even when the flight is fully booked. This forces airlines to oversell (unless they reserve capacity for them, but it is the same as with cancellations: this will lead to empty seats).

The last important property of RM systems is that customers often need multiple resources at the same time. In hospitality many customers book a room for multiple nights, and in aviation many airlines offer connections consisting of multiple flight *legs*. The availability of these different types of capacity cannot be considered separately: a customer wishing to fly from BCN to JFK through AMS won't fly the BCN-AMS leg if the AMS-JFK leg is not available. This makes the problem of determining the optimal pricing strategy highly multi-dimensional.

RM problems both have a stochastic and a dynamic nature: stochastic because demand is unknown, and dynamic because demand changes over time and because pricing decisions made now have consequences for the available of capacity later on. In the remainder of the chapter we discuss first simple static RM models, in which the dynamic aspect is largely simplified. Then we move to more advanced but still 1-dimensional revenue management, first forecasting and then optimization. Finally we discuss approximations for multi-dimensional problems.

18.3 Static models

Before going into the mathematical details let us go back to the early days of RM. Before the deregulation of the aviation section prices prices were fixed and regulated. After the deregulation of the 1970s new competitors entered the market. People Express was one of these, taking a considerable market share off from American Airlines (AA) by having much lower prices. One of the reasons they could do this were the lower costs. For example, they had no advanced reservation system, people paid cash in the aircraft. When AA realized the loss of revenue they came up with the following strategy. The introduced the *Ultimate Super Saver* tariff, below that of Peo-

ple Express, which was only available well beyond the day of departure (to avoid diversion) and with limited availability as to make sure that later customers paying the full price would still find a seat available. The rest is history, People Extress went bankrupt and its CEO said in 1985: "We were a vibrant, profitable company from 1981 to 1985, and then we tipped right over into losing $50 million a month. We had been profitable from the day we started until American came at us with Ultimate Super Savers." The first model is exactly the Super Saver price structure: we have 2 prices, cheap tickets are booked before the regular ones, we have to decide how many seats can be booked by the cheap fare (the *booking limit*) as to maximize expected revenue.

Let us introduce some notation. The capacity is integer and equal to C. The stochastic demand of class i is denoted by D_i, its revenue per sold item is y_i, with $y_1 > y_2$. Type 2 demand occurs before type 1. The question to answer is: how much capacity should be reserved for type 1? The demand distribution for type 2 is irrelevant, if the demand is small then there is no issue whatsoever. Thus we assume for the moment that $D_2 = C$. Our objective is to maximize expected revenue. Assume that the seats are numbered and we start selling them to type 2. For every next seat that we calculate the marginal expected revenue of selling the seat to type 2 or reserving it together will all remaining seats for type 1. Assume there are s seats remaining, thus we consider seat $C - s + 1$. If we sell it to type 2, then the revenue is y_2. With probability $\mathbb{P}(D_1 \geq s)$ we can sell this seat (and all higher numbered seats) to type 1 with revenue y_1. Thus the maximal optimal number of seats reserved for type 1, s_1, is the biggest number s for which:

$$y_2 \leq y_1 \mathbb{P}(D_1 \geq s),$$

thus

$$s_1 = \max_{0 \leq s \leq C} \Big\{ y_2 \leq y_1 \mathbb{P}(D_1 \geq s) \Big\} = \min_{0 \leq s \leq C} \Big\{ F_{D_1}(s) \geq 1 - \frac{y_2}{y_1} \Big\} = F_{D_1}^{-1}\Big(1 - \frac{y_2}{y_1}\Big), \tag{18.1}$$

where F^{-1} is the quantile function as defined in Equation (1.1). The number s_1 is called the *reservation* or *protection level* for type 1. The total revenue is given by

$$y_1 \mathbb{E} \min\{D_1, \max\{C - D_2, s_1\}\} + y_2 \mathbb{E} \min\{D_2, C - s_1\},$$

taking also the randomness in demand of type 2 into account. It is hard to give a closed-form expression for say Poisson distributions, a numerical procedure or simulation has to be used.

Note the similarity of (18.1) to the solution of the newsvendor problem of Theorem 6.2.1. If a seat too many is reserved then the costs are $h = y_2$. If not enough seats

are reserved then the costs are $q = y_1 - y_2$ per seat. $q/(q+h) = 1 - y_2/y_1$ which shows that indeed Equations (6.1) and (18.1) are the same.

Example 18.3.1 An airport parking lot has two classes of customers: those that drive up without reservation and those that make advance reservations over the internet. Let us simplify the problem and look at a single day (perhaps the most crowded day of the week). The discount internet rate is \$10, the drive-up rate \$15. Capacity is 200, the forecasted number of drive-ups is 150. Demand is Poisson distributed. How many internet reservations should we allow? Using a normal approximation of the demand we easily find using the inverse of the normal distribution (for example, using Excel) that we should reserve 145 places for drive-ups. Note that this is less than the expected number of drive-ups!

In reality type-2 customers do not all make their reservation before type-1 customers make theirs. To avoid diversion the possibility to book type 2 should be cancelled from a certain moment on, even if the booking limit has not yet been reached.

Overselling

Consider the same model, but assume that type 2 customers, that already have ordered the product, are willing to change product (e.g., take a later plane) for an indemnity $z_2 < y_1 - y_2$. Then type 1 customers will never be refused, and the reservation level s_1' is the biggest number s that satisfies:

$$y_2 + (y_1 - y_2 - z_2)\mathbb{P}(D_1 \geq s) \leq y_1 \mathbb{P}(D_1 \geq s),$$

which is equivalent to

$$y_2 \leq (y_2 + z_2)\mathbb{P}(D_1 \geq s).$$

It is readily seen that $s_1' \leq s_1$. Note also that the expected marginal revenue per seat is higher, thus the expected total revenue is higher. Thus overselling is profitable! The total revenue is given by

$$y_1 \mathbb{E}\min\{D_1, C\} + (y_1 - z)\mathbb{E}\max\{0, \min\{D_1, C\} +$$

$$\min\{D_2, C - s_1'\} - C\} + y_2 \mathbb{E}\min\{D_2, C - s_1'\}.$$

The outcome can again be simply interpreted by the newsvendor model.

Multiple classes An extension of the model (without overselling) to multiple price classes is the EMSR algorithm, which stands for *Expected Marginal Seat Revenue*. Assume there are n classes, with revenue $y_1 > \cdots > y_n$ and demand $D_1, \ldots, D_n$. The

idea of booking limits is extended to multiple classes, as follows. Consider bookings of type $j+1$. How much of the remaining capacity should we reserve for type $1, \ldots, j$? For every type $i \in \{1, \ldots, j\}$ we compute a booking limit as follows:

$$u_i^{j+1} = F_{D_i}^{-1}(1 - \frac{y_{j+1}}{y_i}).$$

The total number of seats that have to reserved for types 1 upto j when considering selling type $j+1$, written as s_j, is now given by

$$s_j = \sum_{i=1}^{j} u_i^{j+1}.$$

The disadvantage of this method is that the booking limits are simply added, and we would expect (from the central limit theorem, see Section 1.7) that when we add demand over multiple classes that we need relatively speaking less seats to deal with the variation in demand. To account for this, a second version of EMSR is designed, called EMSR-b (the old one becoming EMSR-a). For EMSR-b we add demand over the j types and we calculate the weighted average price $\hat{y}_j$ over classes $1, \ldots, j$:

$$\hat{y}_j = \frac{\sum_{i=1}^{j} y_i \mathbb{E}D_i}{\sum_{i=1}^{j} \mathbb{E}D_i}.$$

Now we take

$$s_j = F_{\hat{D}_j}^{-1}(1 - \frac{y_{j+1}}{\hat{y}_j})$$

with $\hat{D}_j = \sum_{i=1}^{j} D_i$. Instead of one of these approximations we could also use dynamic programming which gives a simple recursion for the optimal solution.

Example 18.3.2 Consider a revenue management instance with three classes, all with normally distributed demand of average 40. The prices are $y_1 = 100$, $y_2 = 80$, $y_3 = 70$. Let us first apply EMSR-a. Computations using Excel show that when selling type-3 seats we should reserve 37 seats for type 1 and 33 for type 2, thus 70 in total. When selling type 2 we should reserve only 35 for type 1. Under EMSR-b, when selling type-3 seats, in total 73 seats should be reserved. Now let the total capacity be 80. Using a Monte Carlo simulation Excel plug-in we find an estimation for the revenue under EMSR-a of 8313 and for EMSR-b of 8321.

18.4 Forecasting

Forecasting in revenue management goes beyond standard methods, such as presented in Section 2.5, for two reasons: observations are censored, i.e., we do not observe the demand but only the sales, and for each price we do not need a number, but the *booking curve*, containing estimates of the demand during the whole booking horizon.

Over time we see a decrease in low-priced sales. This is due to the fact that prices usually increase as we get closer to the check-in/departure date. We do assume that demand increases until the check-in date. For this reason it is not unrealistic to assume that demand, for all prices, has an exponential form.

Non-increasing booking curves

In this section we assume exponential booking curves. Other forms can be required, but are considerably more complicated to model. Especially in hospitality we regularly see decreasing demand. Off-season, when demand is lower than the capacity in a region or city, hotels tend to lower their prices close to the check-in date in an attempt to fill their empty rooms. This results in a drop in high-price demand close to the check-in date and thus a decrease of the high-price booking curves.

Many RM systems work with booking limits. This means that during the period between two runs of the optimization algorithm the availability is regulated by limits on the number of tickets that can be sold for a certain price. This also explains that during the day the price of a ticket can change, then the booking limit has been reached. This however complicates forecasting, because multiple prices are used during the same time interval. To simplify forecasting we assume that a single price is used during each interval. By making time intervals short enough this is a realistic assumption.

Let us formalize this. We assume we want to estimate the booking curves, one for each price or price range, for a day, and we have data of M comparable days. We propose the following model for the rate at dat $t \in \{1, ..., T\}$, with T the number of days a product can be booked, and $i \in \{1, ..., n\}$ the price categories:

$$\lambda_i(t) = \beta_i e^{\alpha_i t}.$$

Let's assume p_{tm} was the price class used on booking day t of departure/check-in day m. Then the sales x_{tm} are a realization of an exponential distribution with rate $\sum_{i=1}^{p_{tm}} \lambda_i(t)$. Note that we assumed diversion, all customers with a higher willingness-to-pay than $y_{p_{tm}}$ pay that price.

Evidently the challenge is to estimate α_i and β_i based on the data. We cannot use a method similar to the one introduced in Section 2.5 because of the exponential form. Therefore we propose to use a maximum likelihood approach for the current problem. The likelihood of observing x_{tm} is:

$$\frac{\left(\sum_{i=1}^{p_{tm}} \lambda_i(t)\right)^{x_{tm}}}{x_{tm}!} e^{\sum_{i=1}^{p_{tm}} \lambda_i(t)}.$$

Taking the log and summing over m and t gives the log-likelihood. Maximizing this over α and β gives their maximum likelihood estimators and with that the prediction for the booking curves. They are input to the optimization algorithm which we discuss in the next section.

To use the method developed so far in practice we have to consider differences in prediction depending on the day of the week, season, etc. A very simple approach could be to split the days in low, high and medium demand and use the algorithm for each set separately. A more advanced approach would be to decompose α and β into different components, comparable to Equation (2.1). Then we can maximize the likelihood using all data available in a single run.

18.5 Dynamic models

One can do revenue management in two ways:
- either one changes the price of the products, of which there is often only one;
- or the prices of the different products are kept fixed, but bying cheap products is made impossible under certain circumstances as to protect capacity for customers willing to pay higher prices.

Both situations can be treated in the same way, by interpreting a product with different prices as different products. However, one should realize that in this case customers make only reservations for the lowest price that is available, although they are perhaps willing to pay more. This also plays a role when the products differ not only in price, although to a lesser extend. In this situation it is called 'diversion'.

Example 18.5.1 A low-cost carrier such as Easyjet has a single product on each flight. The price that is offered for this product is varied depending on many factors such the time until the flight and the number of reservations made already. Traditional carriers have different types of tickets with different conditions. Although most people choose for the ticket with the lowest price, there are still people willing to pay higher prices, for example because it is refundable.

18.6 Multi-resource models

Here we consider the situation where customers require services consisting of multiple products, such as multiple flight legs for an origin-destination pair or multiple night of a hotel room. A way to solve this is by using *bid prices*. A bid price is a way to implement the booking limits of the previous section. A bid price is the level above which booking are accepted. Thus if the booking limit for a class is reached then the bid price is increased above its price. A customer requiring multiple products has a bid price that is the sum of the bid prices of the products.

18.7 Further reading

The book Talluri & van Ryzin [150] is currently the main reference for revenue management. The EMSR algorithm, forecasting and many other relevant details are discussed at length. Cross [45] is a non-mathematical book about the impact of revenue management. Simon [142] is a modern non-technical text on pricing.

The two-class model is introduced by Littlewood (see Brumelle et al. [31]). A good starting point for pricing and price elasticity is the Wikipedia page on "Price elasticity of demand", also for cases with a positive elasticity.

18.8 Exercises

Exercise 18.1 Consider a product with constant but positive marginal costs c.
a. Give an expression for $W(p)$ for this case.
b. Derive the optimal value for the price elasticity in terms of p and c.
c. Give an intuitive explanation for the result you found.

Exercise 18.2 Determine the maximal bundled and unbundled revenue for the next two problems.

		item	
		1	2
customer	1	20	15
	2	15	35

		item		
		1	2	3
customer	1	30	5	15
	2	5	20	20

Exercise 18.3 A railway dedicated to freight has two types of trains: bulk trains (type 1, for example coal) and container trains (type 2). The demand for each hour is Poisson, both with average 3. The variable costs are negligable, the revenue per type is

150 and 200 Euro, respectively. Reservations for bulk trains arrive before container trains.
a. The capacity of the line is 12 per hour. What is the maximal expected revenue, and how can it be obtained?
b. Due to safety regulations in tunnels the capacity is limited to 4. What is the maximal expected revenue, and how can it be obtained?
c. The same questions, but now under the assumption that the demand is deterministic with the same expectation.

Exercise 18.4 A small airplane has a capacity of 10 seats. There are 2 fare classes. The price of a type 1 ticket is E 200, of a type 2 ticket E 500. Type 1 customers book before type 2 customers. The demand for type 1 tickets is 10. The demand for type 2 tickets is distributed as follows:

demand	0	1	2	3	4	5	6	7	8	9	10
probability	0	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

a. Without overselling, how many seats should you sell to type 1 customers to maximize expected revenue?
b. What is the expected total revenue?
A third type of customers (last minute) is introduced with demand 10, price E 100 per ticket, that book after type 2 customers.
c. What is the expected total revenue when you reserve as many seats as determined under a) for type 2?
d. In this new situation, calculate how many seats should be sold to type 1 customers to maximize expected revenue.

Exercise 18.5 Consider Example 18.3.1.
a. Use an Excel Monte Carlo plug-in to simulate this model for various values of s_1, and make a plot with s_1 on the horizontal axis and the total revenue on the vertical axis.
b. Management wants to avoid sending drive-up customers away. What would you advice as a value for s_1?

Exercise 18.6 Consider an airplane with capacity 100, Poisson demand with average 90 and 30 for class 1 and 2, and prices 100 and 300. Class 1 books before class 2.
a. Simulate the demand and determine by trial and error the best booking level and determine the corresponding average revenue. How variable is the total revenue?
b. What is the average revenue if you take the booking limit equal to the average

demand?
c. Determine the optimal booking level by trial and error using the Poisson distribution in for example R or Excel.
d. Approximate the optimal booking level using the inverse of the distribution function.

Exercise 18.7 Make an Excel sheet in which EMSR-a and EMSR-b implemented for 10 booking classes and normally distributed demand. Try different numbers and try to find an instance for which the difference between both models is maximal.

Exercise 18.8 Consider demand in two classes with $T = 60$, and at every time unit the possible events demand in class 1, demand in class 2 and no demand have equal probability.
a. Implement (for example, in Excel) the dynamic programming recursion for this demand, no diversion and $C \leq 60$.
b. Give a table of the optimal policy for $y_1 = 300$, $y_2 = 100$ and $C = 30$. c. Give a demand realization for which the optimal policy first closes class 2 and then opens it again.

Bibliography

[1] Stafford hospital: what went wrong. Times Online, March 17, 2009. `http://www.timesonline.co.uk/tol/news/uk/health/article5925945.ece`.

[2] R.L. Ackoff. The future of Operational Research is past. *Journal of the Operational Research Society*, 30:93–104, 1979.

[3] H.A. Akkermans. Participative business modelling to support strategic decision making in operations — a case study. *International Journal of Operations & Production Management*, 13(10):34–48, 1993.

[4] H.A. Akkermans. *Modelling with Managers*. PhD thesis, Technical University of Eindhoven, 1995.

[5] O.Z. Akşin, M. Armony, and V. Mehrotra. The modern call-center: A multi-disciplinary perspective on operations management research. *Production and Operations Management*, 16:665–688, 2007.

[6] H.I. Ansoff and E.J. McDonnell. *Implanting Strategic Management*. Prentice Hall, 2nd edition, 1990.

[7] J.M. Anthonisse, J.K. Lenstra, and M.W.P. Savelsbergh. Behind the screen: DSS from an OR point of view. *Decision Support Systems*, 4:413–419, 1988.

[8] R.N. Anthony. *Planning and Control Systems: A Framework for Analysis*. Harvard University Press, 1965.

[9] J.J. Arts. Maintenance modeling and optimization. onderzoeksschool-beta.nl/wp-content/uploads/wp_526.compressed-1.pdf, 2017.

[10] S. Asmussen. *Applied Probability and Queues*. Springer, 2nd edition, 2003.

[11] J. Atlason, M.A. Epelman, and S.G. Henderson. Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science*, 54(2):295–309, 2008.

[12] T. Aven and U. Jensen. *Stochastic Models in Reliability*. Springer, 1998.

[13] A.N. Avramidis, W. Chan, M. Gendreau, P. l'Ecuyer, and O. Pisacane. Optimizing daily agent scheduling in a multiskill call center. *European Journal of Operational Research*, 200(3):822–832, 2010.

[14] A.N. Avramidis and P. L'Ecuyer. Modeling and simulation of call centers. In *Proceedings of the 2005 Winter Simulation Conference*, pages 144–151, 2005.

[15] A.O. Awani. *Project Management Techniques*. Petrocelli Books, 1983.

[16] F. Baccelli and G. Hébuterne. On queues with impatient customers. In *Performance '81*, pages 159–179. North-Holland, 1981.

[17] M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors. *Handbooks in Operations Research and Management Science, Vol. 8: Network Routing*. North-Holland, 1995.

[18] R.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, 1975.

[19] F. Baskett, K.M. Chandy, R.R. Muntz, and F.G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 248–260, 1975.

[20] R. Bekker, G.M. Koole, and D. Roubos. Flexible bed allocations for hospital wards. *Health Care Management Science*, 20(4):543–466, 2017.

[21] D. Belson. Managing a patient flow improvement project. In R.W. Hall, editor, *Patient Flow: Reducing Delay in Healthcare Delivery*, pages 429–452. Springer, 2006.

[22] S. Bhulai and G.M. Koole. *Stochastic Optimization*. MG books, Amsterdam, 2015. To appear.

[23] S. Bhulai, S.A. Pot, and G.M. Koole. Simple methods for shift scheduling in multi-skill call centers. *Manufacturing & Service Operations Management*, 10:411–420, 2008.

[24] M. Bijvank, G.M. Koole, and I.F.A. Vis. Optimising a general repair kit problem with a service constraint. *European Journal of Operational Research*, 204:76—85, 2010.

[25] Z. W. Birnbaum, J. D. Esary, and A. W. Marshall. A stochastic characterization of wear-out for components and systems. 37:816–825, 1966.

[26] J. Bramel and D. Simchi-Levi. *The Logic of Logistics*. Springer, 1997.

[27] M.L. Brandeau, F. Sainfort, and W.P. Pierskalla, editors. *Operations Research and Health Care*. Kluwer, 2004.

[28] L. Bromley. Erlang for Excel. www.erlang.co.uk (downloaded September 7, 2020), 2001.

[29] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100:36–50, 2005.

[30] A.M. de Bruin, R. Bekker, L. van Zanten, and G.M. Koole. Dimensioning hospital wards using the Erlang loss model. *Annals of Operations Research*, 2009. To appear.

[31] S.L. Brumelle, J.I. McGill, T.H. Oum, K. Sawati, and M.W. Tretheway. Allocation of airline seats between stochastically dependent demands. *Transportation Science*, 24:183–192, 1990.

[32] J.A. Buzacott and J.G. Shanthikumar. *Stochastic Models of Manufacturing Systems*. Prentice-Hall, 1993.

[33] B. Cardoen, E. Demeulemeester, and J. Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201:921–932, 2010.

[34] M.T. Cezik and P. l'Ecuyer. Staffing multiskill call centers via linear programming and simulation. *Management Science*, 54(2):310–323, 2008.

[35] P. Checkland and John Poulter. *Learning for Action*. Wiley, 2006.

[36] P. Chevalier, R.A. Shumsky, and N. Tabordon. Routing and staffing in large call centers with specialized and fully flexible servers. Downloaded from Google Scholar, September 8, 2020, 2004.

[37] P. Chevalier and N. Tabordon. Overflow analysis and cross-trained servers. *International Journal of Production Economics*, 85:47–60, 2004.

[38] R. Church and C. ReVelle. The maximal covering location problem. *Papers in Regional Science*, 32(1):101–118, 1974.

[39] E. Çinlar. *Introduction to Stochastic Processes*. Prentice-Hall, 1975.

[40] B. Cleveland and J. Mayben. *Call Center Management on Fast Forward*. Call Center Press, 1997.

[41] E.G. Coffman, Jr., J.K. Lenstra, and A.H.G. Rinnooy Kan, editors. *Handbooks in Operations Research and Management Science, Vol. 3: Computing*. North-Holland, 1992.

[42] J.W. Cohen. *The Single Server Queue*. North-Holland, 2nd edition, 1982.

[43] A. Cooper. *The Inmates are Running the Asylum*. Sams Publishing, 2004.

[44] R.B. Cooper. *Introduction to Queueing Theory*. North Holland, 2nd edition, 1981.

[45] R.G. Cross. *Revenue Management: Hard-Core Tactics for Market Domination*. Broadway Books, 1998.

[46] E.L. Crow. The mean deviation of the Poisson distribution. *Biometrika*, 45:556, 1958.

[47] Y. Dallery and S.B. Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems*, 12:3–94, 1992.

[48] G.B. Dantzig. A comment on Edie's "Traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.

[49] M. S. Daskin. A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, 1983.

[50] T.H. Davenport and J.G. Harris. *Competing on Analytics: The New Science of Winning*. Harvard Business School, 2007.

[51] S. Ding, G. Koole, and van der Mei, R.D. On the estimation of the true demand in call centers with redials and reconnects. *European Journal of Operational Research*, 246(1):250–262, 2015.

[52] S. Ding and G.M. Koole. Optimal call center forecasting and staffing. Submitted, 2020.

[53] S. Ding, S. Li, G. Koole, E.I. Yuce, R. van der Mei, and R. Stolletz. Data analysis and validation of call center staffing and workforce models. Working paper, 2020.

[54] M. Dixon, N. Toman, and R. Delisi. *The Effortless Experience*. Pinguin, 2013.

[55] M. El-Taha and S. Stidham, Jr. *Sample-Path Analysis of Queueing Systems*. Kluwer, 1998.

[56] M.C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14:192—215, 2002.

[57] A. Fukunaga, E. Hamilton, J. Fama, D. Andre, O. Matan, and I. Nourbakhsh. Staff scheduling for inbound call centers and customer contact centers. *AI Magazine*, 23(4):30–40, 2002.

[58] J. Galbraith. *Designing Complex Organizations*. Addison-Wesley, 1973.

[59] S. Gallivan. Challenging the role of calibration, validation and sensitivity analysis in relation to models of health care processes. *Health Care Management Science*, 11:208–213, 2008.

[60] S. Gallivan, M. Utley, T. Treasure, and O. Valencia. Booked inpatient admissions and hospital capacity: Mathematical modelling study. *British Medical Journal*, 324:280–282, 2002.

[61] N. Gans, G.M. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5:79–141, 2003.

[62] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman, 1984.

[63] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88:3–12, 1996.

[64] S.B. Gershwin. *Manufacturing Systems Engineering*. Prentice-Hall, 1993 or 1994.

[65] E.M. Goldratt. *Critical Chain*. North River Press, 1997. Dutch translation: *De Zwakste Schakel*, Het Spectrum, 1999.

[66] E.M. Goldratt and J. Cox. *The Goal*. Gower, 1990. Dutch translation: *Het Doel*, Het Spectrum, 1999.

[67] S.C. Graves, A.H.G. Rinnooy Kan, and P. Zipkin, editors. *Handbooks in Operations Research and Management Science, Vol. 4: Logistics of Production and Inventory*. North-Holland, 1993.

[68] L. Green and P. Kolesar. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science*, 37:84–97, 1991.

[69] L.V. Green. Using Operations Research to reduce delays for healthcare. In Zhi-Long Chen and S. Raghavan, editors, *Tutorials in Operations Research*, pages 1–16. INFORMS, 2008.

[70] D. Gross and C.M. Harris. *Fundamentals of Queueing Theory*. Wiley, 2nd edition, 1985.

[71] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–587, 1981.

[72] A.C. Hax and D. Candea. *Production and Inventory Management*. Prentice-Hall, 1984.

[73] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5:43–85, 1995.

[74] D.P. Heyman and M.J. Sobel, editors. *Handbooks in Operations Research and Management Science, Vol. 2: Stochastic Models*. North-Holland, 1990.

[75] W.J. Hopp and M.L. Spearman. *Factory Physics: Foundations of Manufacturing*. McGraw-Hill, 2nd edition, 2001.

[76] M. Houdenhoven, van. *Healthcare Logistics: the Art of Balance*. PhD thesis, Erasmus University Rotterdam, 2007. `http://repub.eur.nl/res/pub/10862`.

[77] R.J. Hyndman. Forecasting with daily data. robjhyndman.com/hyndsight/dailydata (downloaded June 26, 2020), 2013.

[78] R.J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. O Texts, 2018.

[79] R. Ibrahim, H. Ye, P. L'Ecuyer, and H. Shen. Modeling and forecasting call center arrivals: A literature survey and a case study. *The International Journal of Forecasting*, 32:865–874, 2016.

[80] C. Jagtenberg, S. Bhulai, and R. van der Mei. An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care*, 4:27—35, 2015.

[81] L.A. Johnson and D.C. Montgomery. *Operations Research in Production Planning, Scheduling, and Inventory Control*. Wiley, 1974.

[82] C.V. Jones. User interfaces. In E.G. Coffman, Jr., J.K. Lenstra, and A.H.G. Rinnooy Kan, editors, *Handbooks in Operations Research and Management Science, Vol. 3: Computing*. North-Holland, 1992.

[83] R.J. Jorna, H.W.M. Gazendam, H.C. Heesen, and W.M.C. van Wezel. *Plannen en Roosteren*. Lansa, Leiderdorp, 1996.

[84] O. Jouini, G.M. Koole, and A. Roubos. Performance indicators for call centers with impatience. *IIE Transactions*, 45(3):341–354, 2013.

[85] E.L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:457–481, 1958.

[86] F.P. Kelly. *Reversibility and Stochastic Networks*. Wiley, 1979.

[87] W.D. Kelton, R.P. Sandowski, and D.A. Sandowski. *Simulation with Arena*. McGraw-Hill, 1998.

[88] S. Kim and W. Whitt. Are call center and hospital arrivals well modeled by nonhomogeneous poisson processes? *Manufacturing & Service Operations Management*, 16(3):464–480, 2014.

[89] P.J.B. King. *Computer and Communication Systems Performance Modelling*. Prentice-Hall, 1990.

[90] T. Klastorin. *Project Management: Techniques and Tradeoffs*. Wiley, 2003.

[91] J.P.C. Kleijnen. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, 1987.

[92] J.P.C. Kleijnen. Verification and validation of simulation models. *European Journal of Operational Research*, 82:145–162, 1995.

[93] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications*. Wiley, 1975.

[94] L. Kleinrock. *Queueing Systems, Volume I: Theory*. Wiley, 1976.

[95] P.M. Koeleman and G.M. Koole. Optimal outpatient appointment scheduling with emergency arrivals and general service times. *IIE Transactions on Healthcare Systems Engineering*, 2(1):14–30, 2012.

[96] R. Koenker and K.F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15:143–156, 2001.

[97] Ger Koole and Siqiao Li. A practice-oriented overview of call center workforce planning. Submitted, 2021.

[98] G.M. Koole. A formula for tail probabilities of Cox distributions. *Journal of Applied Probability*, 41:935–938, 2004.

[99] G.M. Koole. *Call Center Optimization*. MG books, Amsterdam, 2013.

[100] G.M. Koole. *A Deep Dive into Call Center Workforce Management*. MG books, Amsterdam, 2020.

[101] G.M. Koole and H.J. van der Sluis. Optimal shift scheduling with a global service level constraint. *IIE Transactions*, 35:1049–1055, 2003.

[102] L.R. LaGanga and S.R. Lawrence. Clinic overbooking to improve patient access and increase provider productivity. *Decision Sciences*, 38:251–276, 2007.

[103] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, 1997.

[104] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, editors. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley, 1985.

[105] S. Li, G. Koole, and O. Jouini. A simple solution for optimizing weekly agent scheduling in a multi-skill multi-channel contact center. In *Proceedings of the 2019 Winter Simulation Conference*, 2019.

[106] S. Li, Q. Wang, and G. Koole. Optimal contact center staffing and scheduling with machine learning. Working paper, 2020.

[107] D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[108] O. Madsen, K. Tosti, and J. Vælds. A heuristic method for dispatching repair men. *Annals of Operations Research*, 61:213–226, 1995.

[109] S.G. Makridakis. *Forecasting, Planning, and Strategies for the 21st Century*. The Free Press, 1990.

[110] J. de Mast, R.J.M.M. Does, and H. de Koning. *Lean Six Sigma - for Service and Healthcare*. Beaumont Quality Publications, 2006.

[111] A.G. Mauri. Yield management and perceptions of fairness in the hotel business. *International Review of Economics volume*, 54:284–293, 2007.

[112] E. Mazareanu. Call center market size by region 2012-2017. www.statista.com/statistics/881033/call-center-market-size-region (downloaded June 22, 2020), 2019.

[113] J.R. Meredith. Reconsidering the philosophical basis of OR/MS. *Operations Research*, 49:325–333, 2001.

[114] J.A. Van Mieghem. *Operations Strategy: Principles and Practice*. Dynamic Ideas, 2008.

[115] H. Mintzberg. *The Structuring of Organizations*. Prentice-Hall, 1979.

[116] I. Mitrani. Computer system models. In E.G. Coffman, Jr., J.K. Lenstra, and A.H.G. Rinnooy Kan, editors, *Handbooks in Operations Research and Management Science, Vol. 3: Computing*. North-Holland, 1992.

[117] A. Myskja. The man behind the formula. Biographical notes on Tore Olaus Engset. *Telektronikk*, 94:154–164, 1998.

[118] J. Needleman, P. Buerhaus, V. Pankratz, C. Leibson, S. Stevens, and M. Harris. Nurse staffing and inpatient hospital mortality. *The New England Journal of Medicine*, 364:1037–1045, 2011.

[119] B.L. Nelson. *Foundations and Methods of Stochastic Simulation*. Springer, 2013.

[120] G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors. *Handbooks in Operations Research and Management Science, Vol. 1: Optimization*. North-Holland, 1989.

[121] W.G. Nickets, J.M. McHugh, and S.M. McHugh. *Understanding Business*. McGraw-Hill, 2002.

[122] C. Palm. Methods of judging the annoyance caused by congestion. *Tele*, 4:189–208, 1953.

[123] R. Pasupathy and S. Ghosh. Simulation optimization: A concise overview and implementation guide. In H. Topaloglu, editor, *Tutorials in Operations Research*, pages 122–150. INFORMS, 2013.

[124] M. Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2012.

[125] S.A. Pot, S. Bhulai, and G.M. Koole. A simple staffing method for multi-skill call centers. *Manufacturing & Service Operations Management*, 10:421–428, 2008.

[126] A.A.B. Pritsker. Modeling in performance-enhancing processes. *Operations Research*, 45:797–804, 1997.

[127] M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[128] P. Reynolds. *Call Center Staffing*. The Call Center School Press, 2003.

[129] S.M. Ross. *Simulation*. Academic Press, 6th edition, 1996.

[130] S.M. Ross. *Introduction to Probability Models*. Academic Press, 7th edition, 1997.

[131] S.M. Ross. *A First Course in Probability*. Prentice Hall, 6th edition, 2002.

[132] A. Roubos, S. Bhulai, and G.M. Koole. Flexible staffing for call centers with non-stationary arrival rates. In R.J. Boucherie and N.M. van Dijk, editors, *Markov Decision Processes in Practice*, pages 487–503. Springer, 2017.

[133] A. Roubos, G. Koole, and R. Stolletz. Service-level variability of inbound call centers. *Manufacturing & Service Operations Management*, 14:402–413, 2012.

[134] R.Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley, 1981.

[135] D.A. Samuelson. Predictive dialing for outbound telephone call centers. *Interfaces*, 29(5):66–81, 1999.

[136] R. Sargent. Verification and validation of simulation models. In *Proceedings of the 2005 Winter Simulation Conference, Orlando*, 2005.

[137] W.E. Sasser, Jr. Match supply and demand in service industries. *Harvard Business Review*, 54:133–140, 1976.

[138] S. Savage. *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty*. Wiley, 2012.

[139] J. Seddon. *Systems Thinking in the Public Sector*. Triarchy Press, 2008.

[140] P.M. Senge. *The Fifth Discipline*. Doubleday, 1990.

[141] E.A. Silver and R. Peterson. *Decision Systems for Inventory Management and Production Planning*. Wiley, 2nd edition, 1985.

[142] H. Simon. *Confessions of the Pricing Man*. Springer, 2015.

[143] H.A. Simon. *The New Science of Management Decision*. Prentice-Hall, revised edition, 1977.

[144] D.R. Smith and W. Whitt. Resource sharing for efficiency in traffic systems. *The Bell System Technical Journal*, 60:39–55, 1981.

[145] S. Spear and H.K. Bowen. Decoding the DNA of the Toyota Production System. *Harvard Business Review*, 77(5):96–106, 1999.

[146] R. Stolletz. *Performance Analysis and Optimization of Inbound Call Centers*. Springer, 2003.

[147] R. Stolletz. Approximation of the non-stationary $m(t)/m(t)/c(t)$-queue: The stationary backlog-carryover approach. *European Journal of Operational Research*, 190(2):478–493, 2008.

[148] D.Y. Sze. A queueing model for telephone operator staffing. *Operations Research*, 32:229–249, 1984.

[149] H.A. Taha. *Operation Research: An Introduction*. Prentice Hall, 6th edition, 1997.

[150] K.T. Talluri and G.J. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer, 2004.

[151] J. Taylor and N. Raden. *Smart (Enough) Systems*. Prentice-Hall, 2007.

[152] H.C. Tijms. *A First Course in Stochastic Models*. Wiley, 2003.

[153] H.C. Tijms, M.H. van Hoorn, and A. Federgruen. Approximations for the steady-state probabilities in the $M/G/c$ queue. *Advances in Applied Probability*, 13:186–206, 1981.

[154] P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, 2nd edition, 2014.

[155] J. Toussaint and R.A. Gerard. *On the Mend: Revolutionizing Healthcare to Save Lives and Transform the Industry*. Lean Enterprise Institute, 2010.

[156] TrustRadius. Call center workforce optimization software. www.trustradius.com/call-center-workforce-optimization (downloaded September 8, 2020), 2020.

[157] E. Turban. *Decision Support Systems and Expert Systems*. Prentice-Hall, 4th edition, 1995.

[158] R.B. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing & Service Operations Management*, 7(4):276–294, 2005.

[159] J. Walrand. *An Introduction to Queueing Networks*. Prentice-Hall, 1988.

[160] A. Warner. *The Bottom Line: Practical Financial Knowledge for Managers*. Gower, Aldershot, 1993.

[161] J.D. Welch and N.T.J. Bailey. Appointment systems in hospital outpatient departments. *The Lancet*, 259:1105–1108, 1952.

[162] T.R. Willemain. Insights on modeling from a dozen experts. *Operations Research*, 42:213–222, 1994.

[163] H.P. Williams. *Model Building in Mathematical Programming*. Wiley, 3rd edition, 1993.

[164] W.L. Winston. *Operations Research: Applications and Algorithms*. Duxbury Press, 1987.

[165] W.I. Zangwill. The limits of Japanese production theory. *Interfaces*, 22:14–25, 1992.

[166] S. Zeltyn and A. Mandelbaum. Call centers with impatient customers: many-server asymptotics of the M/M/n+G queue. *Queueing Systems*, 51:361–402, 2005.

[167] H.-J. Zimmermann. An application-oriented view of modeling uncertainty. *European Journal of Operational Research*, 122:190–198, 2000.

[168] P.H. Zipkin. *Foundations of Inventory Management*. McGraw Hill, 2000.