# Inventory control in multi-item production systems

*Citation for published version (APA):*
Bruin, J. (2010). *Inventory control in multi-item production systems*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven.
https://doi.org/10.6100/IR689802

*DOI:*
10.6100/IR689802

*Document status and date:*
Published: 01/01/2010

*Document Version:*
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

*Please check the document version of this publication:*

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

*General rights*
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

*Take down policy*
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 04. Oct. 2023

# Inventory control in multi-item production systems

THOMAS STIELTJES INSTITUTE
FOR MATHEMATICS

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Bruin, Josine

Inventory control in multi-item production systems / by Josine Bruin.
A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-2337-5
NUR 919
Subject headings: multi-item production systems / queueing theory / Markov Decision theory

Printed by Proefschriftmaken.nl
Cover design by Myrthe Isthar Maters

# Inventory control in multi-item production systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 12 oktober 2010 om 16.00 uur

door

Josine Bruin

geboren te Zaanstad

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J. van der Wal
en
prof.dr. A.G. de Kok

# Acknowledgements

This thesis is the result of more than four years of work in which I got support and inspiration from a number of people who I would like to thank. It started in 2005, when Rein Nobel, supervisor of my master's thesis, asked me whether I was interested in starting a PhD project in Eindhoven. He and Henk Tijms introduced me to Ton de Kok, who later became my supervisor at the department of Technology Management. It was a curious step to cross the rivers and although I had to explain this step to many people, I am grateful for the opportunity to live and work in the friendly environment of Eindhoven and its university.

This environment includes the following people. First of all, my promotor Jan van der Wal, who I want to thank for his support, patience and constructive comments, especially when I was nervous to give a talk. I enjoyed the conversations in which he shared his strong intuition for stochastic processes. I am also indebted to Ton de Kok for his enthusiasm and inspiring ideas, and to Onno Boxma for his guidance and pleasant collaboration that resulted in the work presented in Chapter 5. This work was a joint project with Brian Fralix, who I would like to thank for the sometimes confusing, but lively discussions on Laplace-Stieltjes transforms and generating functions in polling systems. I also want to thank Johan van Leeuwaarden, for the enthusiastic and fruitful discussions on the determination of the boundary probabilities in Chapter 6.

Further I am grateful to my colleagues, especially to those who have taught me how to play table tennis and foosball, and to Ingrid & Ingrid for the weekly chocolate breaks. In addition, I thank the administrative staff of EURANDOM for the perfect organization of workshops, conferences and social events like the Sinterklaas gatherings when we got the most wonderful gifts.

Lastly, I thank my family and friends for their support and interest.

# Contents

Chapter 1

# Introduction

## 1.1 Problem

Multi-item production systems find many applications in industry, for instance glass and paper production or bulk production of beers, see Anupindi and Tayur [4]. These systems are characterized by the fact that multiple product types can be made to stock, but have to share the capacity of a single machine. It is difficult to decide which product type to produce next, because often the characteristics for each product type (holding costs, production times, etc.) are different and future demand is not known in advance. Further, production times may be stochastic, due to possible breakdowns or human interference. More importantly, switching times or costs can be incurred for switching from one product type to another, thereby losing time for producing products. The production manager has to come up with a production plan that tells us whether to produce, to switch or to idle the machine.

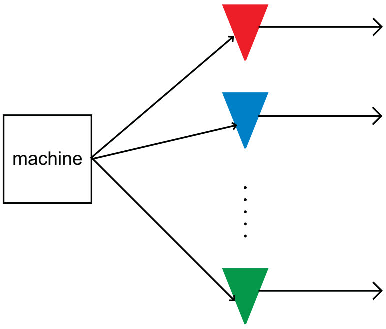


**Figure 1.1:** A multi-item production system

### Objective

The objective for the production manager could, for example, be the minimization of the holding costs under the condition that a certain service level is met, or the minimization of the average waiting time of a customer.

In this thesis, a multi-item production system with set-up times is studied. The production manager deals with the production of multiple items on one machine and has to find a delicate balance between the average number of products on stock and the average number of (arriving) customers who see no stock. Depending on whether the system is dealing with backlogged demand or lost sales, a cost function is considered which consists of holding and backlogging costs (per backlogged unit) or holding and penalty costs (for every lost sale). Because of the stochasticity of the demand, one would like to switch often so that the system quickly reacts to changes in demand. However, a production plan with a lot of switching also means loss of capacity, leading to more backlogged demand or lost sales.

In principle, a minimization of the average costs is possible by modeling and solving the system as a Markov Decision Problem (MDP). Unfortunately, the complexity of the MDP grows exponentially in the number of product types and the number of product types quickly becomes so large that the optimal policy is intractable. The reason for this is that the calculation of the relative values (and thus optimal actions) requires the solution of a set of linear equations. The number of these equations equals the number of possible states. Because the number of possible states grows exponentially in the number of product types, the calculation time of the optimal policy also grows exponentially in the number of product types. For the same reason, the construction or analysis of a policy in which decisions depend on the complete state of the system becomes too complex if the system is too large.

### Stochastic economic lot scheduling problem

The stochastic economic lot scheduling problem (SELSP) is the name for all problems that consider the production of $N$ standardized products on a single machine with limited capacity and set-up times under random demands and random production times. Because the machine can only produce one unit at a time, the production system we consider is an example of a stochastic economic lot scheduling problem. In Winands et al. [93], an extensive literature overview is given on the SELSP and different approaches are discussed which can be divided into different categories, based on the following characteristics.

The first characteristic is the order in which the different product types are produced. In nearly all existing policies this order is fixed, because the analysis of a policy with a dynamic order of product types is often too complex for large values of $N$. This was also seen in Sox and Muckstadt [78] and Qiu and Loulou [69], who look for optimal and near-optimal strategies for small systems. Qiu and Loulou [69] study a system with limited stock space and show that for a 2-item production system the optimal decisions on production depend on the stock levels of both product types. Although this is intuitively easy to understand, it also tells us that the optimal production strategy for systems with more than 2 product types will also depend on the stock levels of all product types.

This observation brings us to the second characteristic, which indicates whether decisions depend on the complete state of the system or not. Following the definitions in the SELSP literature overview of Winands et al. [93], we will distinguish between global and local lot sizing policies. In a global lot sizing policy, decisions on production depend on the complete state of the system, whereas decisions in a local lot sizing policy only depend on the stock level of the product type currently set-up. Besides global or local lot sizing policies, it is also possible to construct other policies where decisions depend on more than one stock level, but not on the complete state of the system. For example, if a fixed order of production is considered, the decision to switch to the next product type may depend on both the stock level of the item currently set-up and the stock level of the next item.

The third and last characteristic is the cycle length of a policy. The cycle length of a policy is the time interval between the starts of two successive production series. Based on this characteristic, policies with a fixed order of production can be divided into two groups, namely one with policies with a fixed cycle length and one in which policies have a dynamic cycle length. Notice that policies with a dynamic order of production automatically have dynamic cycle lengths. Examples of production strategies with a fixed order of production and a dynamic cycle length are gated and exhaustive base-stock policies (see for example Krieg and Kuhn [54] and Federgruen and Katalan [37]), time- and quantity-limited base-stock policies (see de Haan et al. [30] and Eliazar and Yechiali [34]).

For both the gated and the exhaustive base-stock policy, the order of production is fixed and all product flows are served exactly once during one cycle. The difference between the two strategies lies in the fact that under the gated base-stock policy one produces exactly the number of products short to the base-stock level seen by the system just after it was set-up for the current product flow. The system then switches to the next item, while under the exhaustive base-stock policy one produces until the stock level equals the base-stock level before switching to the next item.

Time- and quantity-limited base-stock policies are characterized by the fact that, according to these policies, the machine basically produces according to a gated or exhaustive base-stock policy, but switches earlier if a certain time or production quantity limit is reached. The gated, exhaustive and time- and quantity-limited base-stock policies are all local lot sizing strategies, because the decisions on production only depend on the stock level of the item currently set-up.

An example of a policy with a fixed cycle length is a fixed cycle strategy (see for example Erkip et al. [35]). The structure of this strategy is illustrated in Figure 1.2. The order of production is fixed, but product flows may get more than one production period in one cycle. The lengths of these production periods are fixed, so that each of the product flows experiences a (single-item) periodic production system.

From a practical point of view, the fixed cycle strategy has several advantages for the production manager. For example, if the production system is just one stage

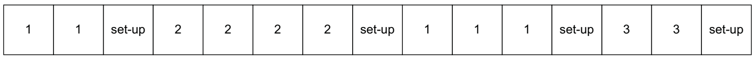


**Figure 1.2:** A fixed cycle for a 3-item production system

among a series of successive stages of production, it is easy to coordinate between the different stages if a fixed cycle strategy is followed. Furthermore, a fixed cycle planning leads to more reliable due dates for customer orders and the strategy is easy to implement on the production floor. However, there are some clear drawbacks of this policy (see Dellaert [32]), of which the most important one is that the system does not react to changes in stock levels of product types that are currently not set-up.

### Analysis

The analysis of the production system can be done with different methods, depending on whether the system deals with lost sales or with backlogged demand. For most systems with lost sales, it is hard to find analytic expressions for measures like the average number of products on stock, so the analysis of a lost sales model often requires numerical methods like successive approximations. Systems with backlogged demand can often be translated into queueing models with infinite buffers. Using methods from queueing theory, like generating functions, one can obtain analytic expressions for measures like the (average) number of customers backlogged, the average number of units on stock, etc.

Because the two systems are analysed with different methods, the thesis is divided into chapters for systems with lost sales and chapters for systems with backlog. Systems with lost sales are studied in the first part of this thesis, while systems with backlogged demand are studied in the second part.

### Translation to a queueing system

The approaches that are used in the backlog model are often studied from a queueing point of view, in which the focus lies on the analysis and minimization of the queue lengths or waiting times. Queueing systems find many applications in, for example, telecommunication systems, traffic lights and production systems. A queueing system is characterized by the arrival process(es) of the customers, the service time distribution(s) and the service discipline. A very basic queueing model is a system with one server and a single queue, as is shown in Figure 1.3. If the maximum stock levels in the production system with backlog are all equal

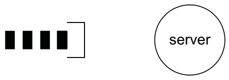


**Figure 1.3:** A queueing model

to zero, the production system becomes a queueing system with one server and multiple queues, as is shown in Figure 1.4. Such a system is also called a polling model. Polling models have been widely studied in the literature (see for example Browne and Weiss [20], Grasman et al. [45], Van der Mei and Borst[84], Resing [71] and Van Vuuren and Winands [87]) and it is often assumed that the server visits the queues in a cyclic order. Each queue has its own arrival process and service time distribution and between the different queues switch-over times can be considered. Polling models find many applications in, for example, communication systems, traffic and manufacturing systems. Two surveys are given in Takagi [79] and Vishnevskii and Semenova [88]. The approaches studied in this field can easily be translated into production strategies by setting a base-stock level for each item and considering the number of units short to these base-stock levels. This is called the *shortfall* of an item and can also be seen as the number of waiting customers.

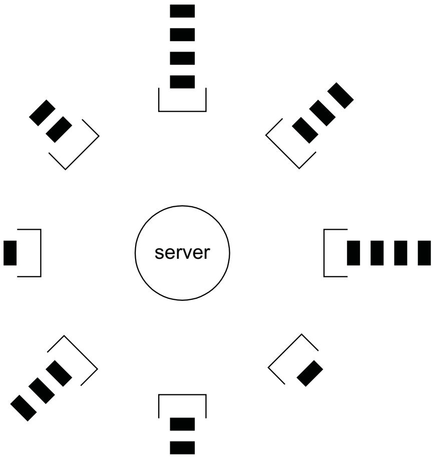


**Figure 1.4:** A queueing model with one server and 8 queues

A polling model with infinite buffer sizes (as is the case in the translated backlog model) is often analysed with a generating function approach. This approach will be explained in more detail in Chapter 4.

The translated lost sales model is a polling model with multiple finite buffer queues. The buffer size of each queue equals the base-stock level of that queue. The number of waiting customers in the queueing model corresponds to the shortfall level, but cannot become larger than the base-stock level, since customers in the production model do not wait but are considered as lost. The characteristics of the queueing model depend on the values of the buffer sizes and thus on the base-stock levels. Furthermore, only very few results (see for example Takine et al. [80]) are known on polling models with finite buffers. The processes at the different queues depend on the buffer sizes of all queues. Similarly, the processes at the different product flows depend on all base-stock levels in the lost sales production system, while in the backlog production system the base-stock level of a product type only influences the process at the corresponding stock point. Therefore, the analysis of a production system with lost sales is in that sense more complicated than the analysis of a production system with backlog.

### Traffic lights

One application of the polling system in Figure 1.4 is the control of a traffic light. This application has a lot of similarities with a multi-item production system. In both systems, there is a single server and multiple queues or product flows and it takes time to switch between two queues or product types. An important difference between the two systems is that in a production system one can make to stock and in that way customers can be served before they arrive. So the production manager has to decide whether or not to produce (more units) to stock, while at an intersection all cars are waiting and cannot be served before arrival.

There exist many studies on the control of a traffic light, see for example the works of Darroch [29], Van den Broek et al. [83] and Haijema and Van der Wal [48]. Darroch and Van den Broek et al. study a fixed cycle control of traffic lights at intersections, which is often used in practice for lightly loaded intersections. For heavily loaded intersections, Haijema and Van der Wal present a two-step approach for the construction of a dynamic control policy that can be obtained for large systems. In the multi-item production system, the possibility of making to stock adds an extra dimension to the problem, but a similar approach as in Haijema and Van der Wal[48] can be used to construct a production strategy for the multi-item production system. This brings us to the contribution of this thesis.

### Contribution

In this thesis, we present the construction of a new production strategy for large production systems in which decisions depend on the complete state of the system. The construction of the new, global lot sizing policy is basically an approach in which a heuristic basis policy is improved with one policy iteration step from Howard's policy iteration algorithm [50]. The idea for this one step improvement approach goes back to Norman [65] and it was used in Wijngaard [92], Bhulai [10], Haijema and Van der Wal [48] and Sassen et al. [73] for production planning, call centers, the control of traffic lights and telecommunication systems, respectively.

The approach is a generic heuristic that starts with a smart basis policy for a complex MDP and then performs a so called improvement step. The choice of the basis policy is important, because for each state a so called relative value has to be found. As was mentioned before, this is impossible if the number of states is too large. So the basis should have a special structure that makes it possible to determine the relative values. Further, the relative values must be easy to obtain if needed. This often results in a very heuristic basis policy, but after one policy iteration, a strategy is constructed in which decisions depend on the complete state of the system. For the production system, the same approach is used, where the basis policy is a fixed cycle policy. As will be shown, the fixed cycle policy allows for a decomposition of the different product flows, which makes it possible to calculate and store the relative values per product type and perform one policy iteration.

Next, we describe how we model the production system and discuss the fixed cycle control and one step improvement approach in more detail.

## 1.2 Model

For both the backlog and the lost sales production system, the number of product types is denoted by $N$ and the products are numbered 1 up to $N$. It is assumed that demand arrives according to (compound) Poisson processes, with an average of $\lambda_i, i = 1, \ldots, N$ per time unit. The system is modeled in discrete time and a one step improvement approach based on a fixed cycle policy is studied. Further, the backlog model is analysed from a queueing point of view by looking at the shortfall levels, while the lost sales model is analysed numerically by looking at the stock levels. Let us discuss each of these elements in somewhat more detail.

### Discrete time

Decisions are taken just after a production or set-up time and therefore, we can embed the process at the decision moments. Because we only look at the system at these decision moments, the system is modeled as if it is in discrete time (see Figure 1.5). Time is divided into slots and because the production and set-up times can be stochastic, the length of each slot may be stochastic. Furthermore, the lengths of the slots can be different, but the lengths of the time slots are assumed to be independent of the demand processes. A disadvantage of looking at the system in discrete time is that idling the machine should also take one time slot, because the length of the time slot may not depend on the demand processes. The advantage is that, because of the assumption on (compound) Poisson demand processes, the system can be modeled as a discrete time Markov process.

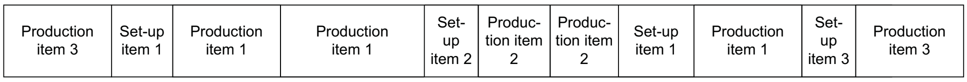


**Figure 1.5:** Time is divided into slots.

If a number is assigned to each slot type, this number tells us what the distribution of the slot length is and it is possible to introduce some notation for the distribution of the demand that arrives during such a slot. Let $a_{i,n}(k)$ denote the probability that demand of type $i$ that arrives during a slot of type $n$ equals $k$.

In the backlog model, a generating function approach will be used to analyse the system. The probability generating function of the arriving demand of type $i$

during a production or set-up time is used, which equals $\sum_{k=0}^{\infty} a_{i,n}(k)z^k$ and will be denoted as $\mathcal{A}_{i,n}(z)$, where $n$ refers to a time slot, production or set-up time.

In order to fulfill the demand, the machine can make to stock. Production takes place per product and the production of one unit of type $i$ requires one production time $T_i^P$. Switching to type $i$ requires a set-up time $T_i^S$. The length of a production or set-up time is possibly stochastic, but independent of the demand process(es) and other production or set-up times.

### State of the system

The stock level of item $i$ is denoted by $I(i)$, which suffices to describe the state at product flow $i$ in the lost sale model. However, in the backlog model customers are backlogged if the stock level equals zero. In that case, the number of units backlogged is denoted by $B(i)$. Another option for the state description is the following. If the maximum stock level, say $S(i)$, is known, one looks at the number of products short to this maximum stock level, denoted by $X(i)$. Obviously, this number is always non-negative and the inventory model is translated into a queueing model by looking at this shortfall level $X(i) = S(i) - I(i) + B(i)$. From a queueing point of view, $X(i)$ can be seen as the number of products that is waiting to be produced. In policies for systems with backlog that apply a base-stock rule for decisions on production, the limiting distribution of $X(i)$ is independent of the value of the base-stock level $S(i)$ (or base-stock levels of other items). Furthermore, if for a specific base-stock policy, the limiting distribution of $X(i)$ is known, a newsvendor type equation can be used to obtain the optimal base-stock level.

### Fixed cycle policy

In a fixed cycle policy often a base-stock rule is used to take decisions on production. Such a fixed cycle strategy is discussed in more detail and analysed in Chapters 2 and 6, and applies the following rules. All product types are produced in a fixed order and for each item, a production period is reserved consisting of a fixed number of production times. During this period, production takes place according to the following rule. If the stock level is below the order-up-to level, one unit of type $i$ is produced. Otherwise, the machine idles during one production time. We also consider this idle time as one time slot. If the production times are stochastic, the idle time is thus also stochastic, with the same distribution as the production times. Therefore, the length of an idle slot is independent from the demand process during that slot.

Because the number of production slots per product type is fixed, one has to number the slots in the fixed cycle and keep track of the number of the current slot. Then, at slot boundaries, an embedded Markov chain is observed. Let $C$ denote the total number of slots in one cycle, $n_m$ the number of the next slot in the cycle after a total number of $m$ slots and $X(i, n_m)$ the shortfall of item $i$, $i = 1, \ldots, N$ just before slot $n_m$.

**Property 1.1.** *The process*

$$\{n_m, X(1, n_m), \dots, X(N, n_m)\}_{m=1}^{\infty}$$

*is a periodic embedded Markov chain with a period of $C$ slots.*

Following the rules for the fixed cycle policy, a production can only start at so-called slot boundaries, i.e. just after a set-up, production or idle time. Obviously, this is suboptimal, but it allows us to analyse the system in discrete time and, more importantly, as a combination of $N$ independent product flows.

**Property 1.2.** *Under the fixed cycle policy, the process at each product flow $i$, $i = 1, \dots, N$ behaves independently from the processes at the other product flows.*

The reason for this is the following. Consider the process at one particular product type $i$. It is then seen that the periodic embedded Markov chain $\{n_m, X(i, n_m)\}_{m=1}^{\infty}$ is not influenced by the processes at the other product flows, because the time that the machine is away consists of a fixed number of set-up and production times and is therefore independent of the processes at the different product flows. Further, the length of the production period of item $i$ is also independent of the shortfall levels of all items and the number of productions in this production period only depends on the shortfall level of item $i$ and not on the shortfall levels of the other items.

### One step improvement

In Chapters 2 and 7, an improvement step of the policy iteration algorithm of Howard [50] is performed that is also used to obtain the optimal policy via an MDP approach. For this one step improvement approach, one needs a smart basis policy. Then, for each possible state, a relative value for this basis policy is calculated. This relative value represents the difference in expected future costs between starting in that state and starting in a certain reference state, under the assumption that in all states the basis policy is followed.

Because for large systems the optimal policy is intractable, the relative value function for the optimal policy is intractable as well. In order to construct a close to optimal production strategy, it is sometimes possible to use a different relative value function. In some problems, this alternative relative value function can be obtained by introducing a heuristic policy with a structure that allows for a tractable relative value function, see for example the works of Sassen et al. [73] on the optimal control of a queueing system and Ott and Krishnan [67] on the optimal routing of a telephone switch. In other problems, it is possible to use the relative value function of a simplified version of the system for which the optimal policy is tractable. The relative value function of the optimal policy for the simplified system can then be used as an approximation for the relative value function for the complex system, as was done in the works of Wijngaard [92] and Bhulai [10] on production planning and multi-skill call centers respectively.

As stated before, the problem in the determination of the optimal policy is that the relative values have to be determined and stored per state. The number of possible states grows exponentially in the number of product types, so this becomes impossible if $N$ is too large. The number of possible states still grows exponentially in the number of product types if a fixed cycle strategy is followed. However, the decomposition property of this strategy allows to determine and store the relative values per product type. For each product type, the number of possible states grows only linearly in the cycle length $C$. Therefore, the total number of (separate) relative values grows linearly in $N \times C$. The relative value for the complete state of the system is just the sum of $N$ separate relative values, which can be calculated at a decision moment.

The one step improvement approach determines the relative urgencies within the fixed cycle policy for each product flow. Based on these relative urgencies, the one step improvement policy calculates the best decision. This decision is executed, the new state is observed and based on the relative urgencies of the heuristic policy, a new decision is calculated, and so on. This policy iteration step can only be performed once, because after this step one has a global lot sizing policy which does not allow for a decomposition of the relative values, so the curse of dimensionality applies again.

Although the analysis for the backlog model differs from the analysis for the lost sales model, there is a large overlap in the notation for the two models. Let us give an overview of this notation.

### Notation

Now that the system is modeled in discrete time, it is possible to express characteristics like demand distributions and slot lengths in terms of the slot type. The type of a slot will be denoted by $n$. This index refers to the type of the slot (a production or set-up slot for a certain item) and therefore also to (the distribution of) the length of the slot. The demand of item $i$ that arrives during a slot of type $n$ is denoted by $D_n(i)$. The distribution of $D_n(i)$ is denoted by $a_{i,n}(k)$, which is the probability that during a slot of type $n$, demand of size $k$ arrives for item $i$. Now that each slot type has an index, the (stochastic) slot lengths can also be denoted by $T_n$ instead of $T_i^P$ and $T_i^S$. The index of a slot type can be any number, as long as each index uniquely refers to a slot type. For example, the production slot of type $i$ could be of type $i$ and a set-up slot of type $i + N$. If a fixed cycle strategy is used, it is more convenient to use the slot number within the fixed cycle as an index for the slot type. In this way, each slot type may have multiple indices, but each index (uniquely) refers to a production or set-up slot. The one step transition costs are also related to the type of the next slot; for each item $i$ we define $c_{i,n}(k)$ as the expected costs during the next slot of type $n$, if the stock level (for lost sales) or shortfall level (for backlog) equals $k$.

In the analysis of the fixed cycle strategy, we will focus on just one of the product types. Therefore, the index of the product type $i$ can be omitted from the notation for the stock, backlog and shortfall level. However, the Markov chain that is observed (see Property 1.1) is embedded at decision moments. Because this Markov chain is periodic, it is necessary to add an index of the slot to the stock, backlog and shortfall level. So instead of looking at $I(i), B(i)$ and $X(i)$ (of item $i$), we look at $I_n, B_n$ and $X_n$ (of slot $n$) or $I_n(i), B_n(i)$ and $X_n(i)$ (of slot $n$ and item $i$). Further, an adjusted fixed cycle policy is studied in the backlog model with a time slot dependent base-stock level $S_n$.

In the fixed cycle policy, the production periods consist of a fixed number of production times. These numbers are denoted by $g_i, i = 1, \ldots, N$. So the total number of slots, denoted by $C$ in a fixed cycle equals $\sum_{i=1}^{N} g_i$ plus the number of set-up slots, which equals $N$ if each item gets one production period per cycle. The relative values for the fixed cycle policy are, by definition, related to the slot type. These values are denoted by $r(n, k_1, \ldots, k_N)$, with $n$ the type of the slot and $k_i$ the stock or shortfall level of type $i, i = 1, \ldots, N$. Because of Property 1.2, $r(n, k_1, \ldots, k_N)$ can be decomposed into $N$ individual relative values $r_i(n, k_i), i = 1, \ldots, N$.

In Chapter 5, a polling model is studied in which $Q_i$ refers to a queue of type $i$. This queueing system is modeled in continuous time and therefore, $C$ will denote the *duration* of a cycle instead of the number of slots within a cycle.

## 1.3 Structure

The structure of this thesis is as follows. In the first part, the fixed cycle and one step improvement policy are discussed and analysed for the lost sales model. In Chapter 2, a literature overview is given and a system with one machine is studied. Then, in Chapter 3, we discuss how to perform the two step approach in a system with two machines. In the second part, we continue with a literature overview for the backlog model in Chapter 4. In Chapter 5, this overview is followed by a study on waiting time distributions of customers in a polling system, where a generating function approach is used to obtain more insights into the effect of the service order at the different queues on the first two moments of the waiting time of the customers. Chapters 6 and 7 analyse the fixed cycle and one step improvement policy for the backlog model, respectively. Chapter 8 summarizes the insights and results obtained for the lost sales and the backlog model, discusses the differences in the analysis and performance of the one step improvement policy in the two models and gives suggestions for future research.

Chapter 2

# Lost sales: One step improvement

The current chapter is based on the work in [22] and [25] and discusses the construction of a one step improvement policy in a production system with lost sales, based on a fixed cycle strategy. First, the fixed cycle strategy is analysed. This strategy reserves a production period for every item $i$, consisting of a fixed number of $g_i$ production slots. Because each production and set-up time represents one time slot, one cycle consists of $C = \sum_{i=1}^{N} g_i + N$ time slots. These slots may have random durations, but the lengths of the slot durations are independent.

The fixed cycle policy allows for a decomposition so that an improvement step can be performed. For each product type, the relative values are determined per slot in the fixed cycle. Then, in the one step improvement policy, at each decision moment, the fixed cycle slot with the minimum relative value is chosen (for the current stock levels). This approach is discussed and analysed in the last three sections of this chapter.

## 2.1 Introduction

In the lost sales model, the state space of the embedded Markov chain (as described in Property 1.1) for each product type is bounded by zero and the maximum stock level. The state of the complete system is described by $\{i, I(1), \ldots, I(N)\}$, with $i$ the item that is currently set-up and $I(j)$ the number of products on stock for item $j$, $j = 1, \ldots, N$.

### Polling

As mentioned in the introduction of this thesis, the production system is translated into a polling model by looking at the shortfall values $X(i) = S(i) - I(i), i = 1, \ldots, N$. Grasman et al. [45] derive the queue length distributions for such a polling system with finite buffers and an exhaustive visit discipline. The exhaustive visit discipline in the queueing model is equivalent to the exhaustive base-stock policy in the production system discussed in Chapter 1. According to this discipline, the server serves a queue until it is empty and then switches to the next queue. Grasman et al. note that the complexity of their analysis grows exponentially in the number of queues and the buffer sizes. Therefore, the queue length distributions for large systems are intractable.

Chung et al. [27] and Lee and Sunjaya [56] also look at a polling system with finite buffers, but consider a random polling order and the buffer size equals one for all queues. Distributional results for the queue lengths and waiting times are obtained and it is shown that their approach also works for a polling system with buffer sizes equal to $S(i)$. However, the number of equations that need to be solved to analyse the system equals $N\prod_{i=1}^{N}(S(i)+1)$, the number of possible states. This number grows exponentially in $N$, so (again) for large values of $N$ and $S(i)$ the analysis becomes too complex.

### Production

For the production system, the optimal strategy can only be found for small systems. The MDP approach is intractable if $N$ is too large, which is illustrated by the following example. Consider a production system with 6 product types. If the maximum stock level for each product type equals 10, the number of possible states equals $6 \times 11^6$, which is more than ten million and therefore already too much for an MDP approach.

Therefore, alternative strategies have been studied for large systems, for example the exhaustive base-stock policy in Krieg and Kuhn [54] and Grasman et al. [46]. Grasman et al. [46] show that an optimal solution for the values of $S(i), i = 1, \dots, N$ is intractable for large systems and provide a heuristic for finding the base-stock levels. Krieg and Kuhn [54] present a method to estimate performance measures with a decomposition based approximation method. Further, Altiok and Shiue [3] analyse the joint behavior of the inventory levels of the different product types, which are produced according to a priority structure. Both the exhaustive base-stock and the priority policy are local lot-sizing policies, because decisions on production, switching or idling only depend on the stock level of the product type currently set-up.

Global lot sizing policies are often more difficult to analyse, because of the multi-dimensionality of the system, particularly if $N$ is large. But for the – local lot sizing – exhaustive base-stock policy the same problem is encountered. Therefore, Krieg and Kuhn [54] approximate the performance measures of this policy by decomposing the system into $N$ subsystems. These subsystems are assumed to be independent, so that the system can be analysed.

But if one follows an exhaustive base-stock policy, the different product flows are not independent. The dependence between the different product flows can be illustrated as follows. If, for example, the production period of one item is long, the other items are likely to have a long production period as well. The reason for this is simple: the expected number of customers that arrive during this long production period is higher than in a production period of average length. It is likely that the service of more customers also takes more time, which comes down to a longer production period. This effect also works the other way around, because a short production period of one item leads – in expectation – to short production periods of the other items. So the processes at the different product flows influence each other, because the length of a production period of one item depends on the lengths of the production periods of all other items.

In a fixed cycle policy, there is no dependence between the different product flows, because each production period has a fixed length and is thus independent of the arrival process (see Property 1.2). Therefore, the analysis of the complete system under this strategy is exact if all product flows are analysed individually. Furthermore, because of this property of independency between the different product flows, the fixed cycle policy can be used as a basis for a one step improvement approach.

In the next section, the one step transition probabilities and costs are given. Then, it is shown how the fixed cycle can be analysed with successive approximations. With this analysis, a good fixed cycle can be found with a local search algorithm presented in Section 2.3. For this fixed cycle, relative values are determined and an improvement step is performed, as presented in Sections 2.4 and 2.5. Results on this new strategy are given in Section 2.6, which is followed by a conclusion in Section 2.7.

## 2.2 Costs and transitions

In the fixed cycle policy, the slots are numbered 1 up to $C$. At each slot boundary, the stock level is observed and costs are incurred based on that stock level and the number of the time slot. Depending on the slot number $n$ and the stock level $I_n(i)$ of item $i$, these one step transition costs equal $c_{i,n}(I_n(i))$, which are the expected costs during the next slot.

### Expected costs

At the start of each slot, the expected costs during that slot are calculated. The expected penalty costs are just the expected number of lost sales times $c_{i,P}$. The holding costs can be incurred in continuous time or in discrete time. Incurring the holding costs in continuous time means that holding costs are paid for each unit during the exact time that it is on stock. Therefore, one then has to keep track of each event during each time slot. If the holding costs are incurred in discrete time, holding costs are paid for every unit on stock for the next slot (regardless whether the stock level decreases or not). So one only needs to look at the system at fixed time instants, as was also done by Fleischmann [41] for the discrete lot-sizing and scheduling problem (DLSP).

The structure of the cost function does not essentially change under either a continuous or a discrete time cost model assumption, because the costs still grow linearly in both the number of products on stock and in the number of lost sales. Because the model we look at is already in discrete time, we prefer to also incur the holding costs in discrete time. However, if the lengths of the production and set-up times (and thus time slots) are very different, it is more natural to divide the slots into smaller slots. Therefore, we choose to incur the costs in the following way.

During each slot, the stock level is observed after each time unit. For the observed stock level, holding costs are paid for the next time unit. If the remaining time until the next decision moment is less than one time unit, holding costs are only paid for the time until that moment. The length of a slot is stochastic, so at the beginning of a slot $n$, the one step expected costs $c_{i,n}(k)$ are $\mathbb{E}\left(c_i(k,T_n)\right)$, with

$$c_i(k,t) = \begin{cases} c_{i,I}tk + c_{i,P}\mathbb{E}(D(i,t)-k)^+, & \text{if } t \leq 1 \\ \sum_{l=0}^{k-1} P(D(i,1)=l)\left(c_{i,P}(k-l) + c_i((k-l)^+, t-1)\right) & \\ +P(D(i,1)\geq k)c_i(0,t-1) + c_{i,I}k + c_{i,P}(\lambda_i - k), & \text{if } t > 1, \end{cases} \tag{2.1}$$

with $D(i,t)$ the demand during a time interval of length $t$.

For deterministic slot lengths, the expected costs are exactly $c_i(k,T_n)$. If the length of the next slot is stochastic, $\mathbb{E}\left(c_i(k,T_n)\right)$ becomes an integral in $T_n$. The total expected penalty costs can be calculated directly, with $c_{i,P}\mathbb{E}\left(D(i,T_n)-k\right)^+$. The expected holding costs equal

$$\sum_{j=0}^{\infty} P(T_n \geq j) p_{i,n}^{(j)}(k,k')k'c_{i,I} + \mathbb{E}\left(T_n - j | j \leq T_n < j+1\right) k'c_{i,I},$$

with $p_{i,n}^{(j)}(k,k')$ the probability that during $j$ time units, the stock level changes from $k$ to $k'$. This summation can be calculated numerically if $T_n$ has an upperbound. If $T_n$ has no upperbound, one has to approximate the expected holding costs.

### Transition probabilities

Let $p_{i,n}(k,k')$ denote the transition probability that the stock level of item $i$ changes from $k$ at slot boundary $n$ to $k'$ at slot boundary $n+1$ and $a_{i,n}(k) = P(D_n(i)=k)$. Note that in the fixed cycle policy, slot boundary $C+1$ must be read as slot boundary 1. Then for production slots for item $i$ it holds that:

$$I_{n+1}(i) = (I_n(i) - D_n(i))^+ + 1_{\{I_n(i)<S(i)\}}.$$

For non productions slots for item $i$, one has

$$I_{n+1}(i) = (I_n(i) - D_n(i))^+.$$

This leads to the following transition probabilities in slot $n$ for product type $i$:

$$p_{i,n}(S(i),k) = a_{i,n}(S(i)-k), \quad 0 < k \leq S(i), \tag{2.2}$$

$$p_{i,n}(S(i),0) = P(D_n(i) \geq S(i)) = 1 - \sum_{j=0}^{S(i)-1} a_{i,n}(j). \tag{2.3}$$

If $n$ is a production slot, then

$$p_{i,n}(k,l) = a_{i,n}(k-l+1),\ 0 < l-1 \leq k < S(i), \tag{2.4}$$

$$p_{i,n}(k,1) = P(D_n(i) \geq k) = 1 - \sum_{j=0}^{k-1} a_{i,n}(j),\ 0 \leq k < S(i). \tag{2.5}$$

For all non production slots,

$$p_{i,n}(k,l) = a_{i,n}(k-l),\ 0 < l \leq k \leq S(i), \tag{2.6}$$

$$p_{i,n}(k,0) = P(D_n(i)) \geq k) = 1 - \sum_{j=0}^{k-1} a_{i,n}(j),\ 0 \leq k \leq S(i). \tag{2.7}$$

### 2.2.1 Successive approximations

In order to compute the expected costs per time unit one may use successive approximations. Define $v_{i,m}(n,k)$ as the expected costs over the next $m$ time slots for item $i$, starting from slot boundary $n$ with stock level $k$. Then

$$\begin{aligned} v_{i,1}(n,k) &= c_{i,n}(k),\quad n = 1,\dots,C, \\ v_{i,m}(n,k) &= c_{i,n}(k) + \sum_{l=0}^{S(i)} p_{i,n}(k,l) v_{i,m-1}(n+1,l),\ n < C,\ m \geq 2, \\ v_{i,m}(C,k) &= c_{i,C}(k) + \sum_{l=0}^{S(i)} p_{i,C}(k,l) v_{i,m-1}(1,l),\ m \geq 2. \end{aligned}$$

For all $n$ and $k$, the expected costs over one cycle $(v_{i,m+C}(n,k) - v_{i,m}(n,k))$ converge to the average costs per cycle. So for every pair $n$ and $k$,

$$\frac{v_{i,m+C}(n,k) - v_{i,m}(n,k)}{\sum_{j=1}^{C} T_j} \to c_i(g,S) \quad (m \to \infty),$$

where $c_i(g,S)$ denote the expected costs per time unit for item $i$, with $g = (g_1,\dots,g_N)$ the lengths of the production periods and $S = (S_1,\dots,S_N)$ the base-stock levels. The total expected costs per time unit equal

$$c_{tot}(g,S) = \sum_{i=1}^{N} c_i(g,S). \tag{2.8}$$

Note that for every item $i$, $c_i(g,S)$ does not depend on $S(j), j \neq i$ and thus also can be written as $c_i(g,S(i))$. The optimal fixed cycle is the fixed cycle that minimizes the total expected costs per time unit $c_{tot}$. The expected costs per time unit depend on the lengths of the production periods $g_1,\dots,g_N$ and the base-stock levels $S(1),\dots,S(N)$.

## 2.3 Finding a near-optimal fixed cycle

We are looking for a fixed cycle that minimizes the expected costs per time unit, i.e. we have to determine two sets of parameters; $g_1, \ldots, g_N$ and $S(1), \ldots, S(N)$. This fixed cycle will be used as a basis for the one step improvement approach. First a local search algorithm is presented in the current section to find a not necessarily optimal, but good fixed cycle.

From Property 1.2, we know that for any combination of production periods $g_1, \ldots, g_N$, the $N$ product flows can be analysed separately. So for a fixed combination of $g_1, \ldots, g_N$, the values of $S(1), \ldots, S(N)$ can be determined per product flow. The periodic production problem for each item $i$ is equivalent to the well known newsvendor problem, with the cost function being convex in the base-stock level, see Khouja [52]. So for each item $i$, $S(i)$ is increased with 1 until the expected costs (for item $i$) per time unit increase. Using these base-stock levels, the minimum expected costs per time unit can be found for any combination of $g_1, \ldots, g_N$. This still leaves the question of how to find the optimal values of $g_1, \ldots, g_N$.

In Haijema and Van der Wal [48] a simple local search algorithm is used to find (near) optimal green times for the traffic lights of the various traffic flows. They start with a cycle of minimum length and one time slot is added (picking the best option among all traffic flows) until for a number of steps no decrease of the average costs per time unit is found. For the production problem, a similar approach is used which works as follows.

Let $g$ denote the vector $(g_1, \ldots, g_N)$ and $c_{tot}(g)$ the expected costs per time unit for a cycle described by $g$ and its corresponding optimal values of $S(1), \ldots, S(N)$. In every iteration of the search algorithm, a number of $N$ fixed cycles is constructed. At the start, $g^{(0)}$, a cycle with just switch-over times ($g^{(0)} = 0$) is constructed, so all demand is lost and the expected costs per time unit equal $c_{tot}(0) = \sum_{i=1}^{N} p_i \lambda_i$.

Secondly, for every item $i$, $c_{tot}(g^{(0)} + e_i)$ is calculated, with $e_i$ a vector with $N-1$ zeroes and $e_i(i) = 1$. Let $i^*$ denote the item that minimizes $c_{tot}(g^{(0)} + e_i)$, then the vector $g^{(1)}$ is defined as $g^{(0)} + e_{i^*}$. The vectors $g^{(2)}$ up to $g^{(N)}$ are determined in a similar way: $g^{(k+1)} = g^{(k)} + e_{i^*_k}$, with $i^*_k = \arg\min_i c_{tot}(g^{(k)} + e_i)$.

If any of the vectors $g^{(1)}, \ldots, g^{(N)}$ gives lower costs than $g^{(0)}$, $g^{(0)}$ is updated with the vector corresponding to the lowest costs. Based on this new value of $g^{(0)}$, the new set of vectors $g^{(1)}, \ldots, g^{(N)}$ are found. This is repeated until no cost reduction is obtained.

## 2.4 One step improvement approach

Now we come to the final step of our construction of a dynamic policy for the multi-item production system. The approach, known as *one-step improvement*, is in fact the policy improvement step in Howard's policy iteration algorithm, see [50]. In order to execute the improvement step, the *relative values* or *bias terms* are needed. If the number of states is very large, these relative values cannot be computed within reasonable time, unless the structure of the stationary strategy is very special. The fixed cycle strategy is a stationary strategy that does have the required special form, since for any given $g$ the 'behavior' of the different products is completely independent, so that calculations can be done one product at a time.

In the improvement step one minimizes the future expected costs, under the assumption that after this decision the original strategy, in our case fixed cycle policy, is followed. This basically means that a decision should indicate which time slot is performed next. This time slot is the best possible one based on the assumption that after this slot one resumes the fixed cycle policy. The relative values are compared to find this slot and represent the relative costs for resuming the fixed cycle policy, starting from a certain time slot. The dynamic policy continues computing such a best slot at the end of every slot. After the one step improvement decision, the fixed cycle strategy just continues with the next time slot in the cycle, while the dynamic policy chooses the best slot in the cycle again, assuming that after this *time jump* the system will be controlled by the fixed cycle rule. In order to compute the next slot we need the relative value for each of the allowed time jumps (not all time jumps are possible as switch-over times are non-zero).

For state $(n, k_1, \ldots, k_N)$ the possible decisions, or slots one can jump to within the cycle, that have to be considered in the improvement step depend on $n$. If $n$ corresponds to the start of a production slot for product $j$ or if $n$ is the start of the switch-over slot from product $j$ to product $j+1$ all production slots for product $j$ and all set-up slots are allowed. The slot to be chosen is the one for which the relative value is minimal.

## 2.5 Relative values

Let us come to the computation of the relative values. As said, in order to compute the relative values, we can consider one product at a time. A complication arises from the fact that the fixed cycle strategy is periodic. For a non-periodic Markov chain, the $m$-period costs $v_m$ asymptotically behave as

$$v_m = mc + r + o(1) \quad (m \to \infty) \ , \tag{2.9}$$

with $m$ the number of time units, $c$ the average costs per time unit and $r$ the relative value vector.

The relative values represent the difference in costs between starting in one slot and starting in another slot, assuming that the fixed cycle is followed. So the relative values depend on the characteristics of the fixed cycle policy, i.e. the base-stock levels and lengths of the production periods. But to keep the notation simple, we do not refer to these characteristics and denote the relative value for slot $n$ and state $(k_1, \ldots, k_N)$ by $r(n, k_1, \ldots, k_N)$.

If the lengths of the time slots are different, one can transform the system so that the processes at the different product flows become aperiodic embedded Markov chains. Without loss of generality, we assume that $T_n \geq 1$ for all $n$. Then in the adjusted system, each slot $n$ is executed with probability $1/T_n$ and the complete state of the system remains the same with probability $1 - 1/T_n$. The one step transition costs are also divided by $T_n$, so that the expected costs before reaching the next slot are still $c_{i,n}(k)$, because it takes on average $T_n$ trials to reach the next slot. This basically is the aperiodicity transformation introduced in Schweitzer [74].

For a periodic Markov chain with slots of unit length and cycle time $C$, one can use as estimate for the relative value vector

$$r^{(m)} = \frac{1}{C} \sum_{n=mC+1}^{(m+1)C} v_n, \tag{2.10}$$

provided $m$ is sufficiently large. Note that in the policy improvement step one does not need the exact value of $r$, any vector $r + \alpha$ with $\alpha$ an arbitrary constant vector will do.

Now, denote for every product type $i$ the state of the system as $(n, k_i)$, with $n$ the slot within the fixed cycle and $k_i$ the number of products in stock. For the fixed cycle strategy, the relative values per state can be approximated by taking $m$ sufficiently large in (2.10):

$$\hat{r}_i(n, k_i) = r_i^{(m)}(n, k_i) = \frac{1}{\sum_{j=1}^{C} T_j} \sum_{l=mC+1}^{(m+1)C} T_{l-mC}(v_{i,l}(n, k_i) - v_{i,j}(n_0, k_0)). \tag{2.11}$$

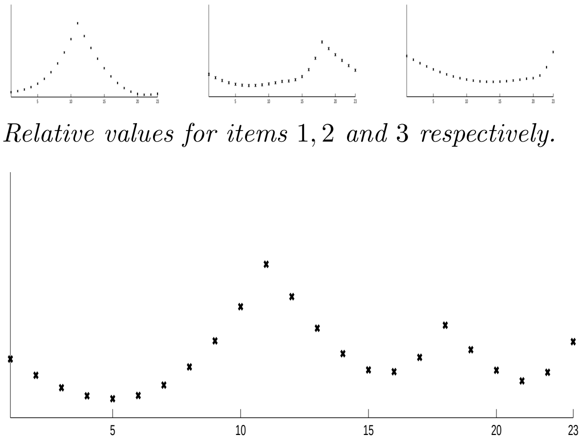
The overall (approximate) relative value $\hat{r}(n, k_1, \ldots, k_N)$ for time slot $n$ and state $(k_1, \ldots, k_N)$ is then taken to be the sum of the relative values for the $N$ products and pairs $(n, k_j)$, $j = 1, \ldots, N$:

$$\hat{r}(n, k_1, \ldots, k_N) = \sum_{i=1}^{N} \hat{r}_i(n, k_i).$$

If the number of states is very large, registering these relative values per state might already be a problem. However, the registration of the relative values per product type requires only a one-dimensional array per stock value. So for $N$ different product types, only $N$ matrices of size $S(i)$ by $C$ are needed.

### 2.5.1 Numerical example

Let us illustrate this one step improvement approach in a numerical example. Consider the following 3-item production system. For every item, the holding costs are equal to 1 and the penalty costs are equal to 100. The production and switch-over times are assumed to be deterministic and of unit length. Furthermore, demand occurs according to Poisson processes with parameters $\lambda_1 = 0.45, \lambda_2 = 0.27$ and $\lambda_3 = 0.18$. The local search algorithm gives us a presumably optimal fixed cycle with $g_1^* = 10$, $g_2^* = 6$, $g_3^* = 4$, so $C = 23$. The optimal base-stock levels for this fixed cycle are $S^*(1) = 10$, $S^*(2) = 8$ and $S^*(3) = 6$. There are 5, 4 and 3 products on stock for respectively product types $1, 2$ and 3. The relative values for this state



*Relative values for items* 1, 2 *and* 3 *respectively.*

*The total relative value function.*

**Figure 2.1:** Three individual relative value functions and the total relative value function for a 3-item production system with stock levels 5, 4 and 3 for items 1, 2 and 3 respectively

of the system are given in Figure 2.1.

If the cycle is in a production slot for item 1, the possible decisions are the first ten (production) slots and the (switch-over) slots 11, 18 and 23. If the stock levels equal 5, 4 and 3 like in Figure 2.1, the global minimum of the total relative value function in slot 5 indicates that the fifth time slot will be executed next in the one step improvement policy. However, if item 2 is currently set-up, it is not allowed to execute the fifth slot. In that case, the next slot to execute according to the dynamic policy is the (production) slot with number 16, a local minimum.

### 2.5.2 Evaluation

For large values of $N$, the number of possible states is very large and the only way to evaluate the new dynamic strategy is by simulation. In the next section, each simulation run has a duration of 25 million slots. For the results presented here, this gives us standard deviations below 1% of the total average costs. A simulation goes as follows. At the start of each slot, the state is observed and the relative values for that state are computed as the sum of the separate relative values. Then the time slot for which the relative costs are minimal is chosen, the expected costs for this slot are added to the total costs and the slot is executed. Then the transition is observed and the next decision is computed. For this decision, the expected costs are added to the total costs, the decision is executed, the new state is observed again and so on.

## 2.6 Results

In order to get some insights in the performance of the one step improvement policy, results are obtained for different parameter settings. There is a large number of parameters that can be changed. The topics that are studied in this section include the number of product types, the load on the system, the holding and penalty costs, the demand distributions and the lengths of the set-up times. We think that the examples shown in this section give a representative view on the performance of the one step improvement policy and provide a good intuition on when this policy outperforms other existing production strategies.

Next, the effect of a suboptimal fixed cycle on the performance of the one step improvement policy is briefly discussed.

### A good fixed cycle

The fixed cycle obtained from the algorithm presented in Section 2.3 is not necessarily optimal. In order to show that this is not very important for the performance of the one step improvement policy, the results in Table 2.1 are given. The results in the table on the left show the performance of both the fixed cycle policy (FC) found with the local search algorithm of Section 2.3 and the one step improvement policy (1SI) based on that fixed cycle policy. The base-stock levels of the fixed cycle policy are decreased and based on this adjusted fixed cycle policy, an improvement step is performed. The results for these two production strategies are shown in the table on the right. The one step improvement step reduces the expected costs by around 17% for the fixed cycle policy with the optimal base-stock levels, while for the fixed cycle policy with the decreased base-stock levels, the costs are reduced

Optimal *fixed cycle* base-stock levels

| $\lambda$ | FC | 1SI |
|---|---|---|
| (0.15,0.15,0.15,0.15) | 15.22 | 12.84 |
| (0.15,0.25,0.1,0.2) | 17.94 | 14.76 |
| (0.1,0.1,0.1,0.2,0.2) | 20.37 | 16.71 |

$c_{i,I} = 1, c_{i,P} = 100, i = 1, \ldots, N$

Decreased base-stock levels

| $\lambda$ | FC | 1SI |
|---|---|---|
| (0.15,0.15,0.15,0.15) | 16.12 | 12.43 |
| (0.15,0.25,0.1,0.2) | 18.97 | 14.52 |
| (0.1,0.1,0.1,0.2,0.2) | 22.01 | 16.89 |

$c_{i,I} = 1, c_{i,P} = 100, i = 1, \ldots, N$

**Table 2.1:** Multi-item production systems with Poisson demand

Optimal *fixed cycle* base-stock levels

| $\lambda$ | g | S |
|---|---|---|
| (0.15,0.15,0.15,0.15) | (3,3,3,3) | (4,4,4,4) |
| (0.15,0.25,0.1,0.2) | (3,6,2,5) | (5,6,3,5) |
| (0.1,0.1,0.1,0.2,0.2) | (2,2,2,5,4) | (3,3,3,5,6) |

$c_{i,I} = 1, c_{i,P} = 100, i = 1, \ldots, N$

Decreased base-stock levels

| $\lambda$ | g | S |
|---|---|---|
| (0.15,0.15,0.15,0.15) | (3,3,3,3) | (3,3,3,3) |
| (0.15,0.25,0.1,0.2) | (3,6,2,5) | (4,5,2,4) |
| (0.1,0.1,0.1,0.2,0.2) | (2,2,2,5,4) | (2,2,2,4,5) |

$c_{i,I} = 1, c_{i,P} = 100, i = 1, \ldots, N$

**Table 2.2:** The values of $g$ and $S$

with approximately 23%. This tells us that the fixed cycle policy is not a good policy.

It is also seen that for two of the examples in Table 2.1, the performance of the one step improvement policy is better for the suboptimal fixed cycle strategies. However, for the example with the 5-item production system, both the performance of the fixed cycle policy and the policy of the one step improvement policy get worse if the base-stock levels are decreased. Apparently, it is important to start with a good fixed cycle, but also the values of the base-stock levels are important, because they determine the maximum stock levels in the one step improvement policy.

With the decreased base-stock levels, one can also search for the optimal lengths of the production periods for these base-stock levels. This is done with the algorithm presented in Section 2.3, but now the base-stock levels are kept fixed. The results are shown in Table 2.3. It is seen that compared to the results in the table on the right in Table 2.1, only the costs for the first two examples are reduced. So one can conclude that decreasing the base-stock levels does not necessarily lead to a better one step improvement policy. Therefore, the remaining results in this section are based on the fixed cycle policy obtained with the algorithm presented in Section 2.3.

Adjusted production periods

| $\lambda$ | g | S | FC | 1SI |
|---|---|---|---|---|
| (0.15,0.15,0.15,0.15) | (2,2,2,2) | (3,3,3,3) | 15.88 | 12.31 |
| (0.15,0.25,0.1,0.2) | (2,4,3,6) | (2,4,4,5) | 18.66 | 14.48 |
| (0.1,0.1,0.1,0.2,0.2) | (2,2,2,4,5) | (2,2,2,4,5) | 22.00 | 16.92 |

$c_{i,I} = 1, c_{i,P} = 100, i = 1, \ldots, N$

**Table 2.3:** Multi-item production systems with Poisson demand

### A comparison

In order to compare the performance of the proposed one-step improvement policy with other policies, simulation studies for 6-item and 10-item production systems are performed.

The results in Tables 2.4 and 2.5 are based on the following parameter settings:

- All production- and set-up times are deterministic and of unit length; $T_n = 1, n = 1, \ldots, C$.
- Demand for item $i$ is Poisson with arrival rate $\lambda_i$, $i = 1, \ldots, N$. For a 6-item production system, $\lambda = (0.25\rho,\ 0.15\rho,\ 0.10\rho,\ 0.25\rho,\ 0.15\rho,\ 0.10\rho)$ and for a 10-item production system, $\lambda_i = 0.1\rho, \forall i$.
- $c_{1,I} = c_{2,I} = \ldots = c_{N,I} = 1$ and in the 6-item production system, $c_{1,P} = c_{2,P} = \ldots = c_{6,P} = 100$. In the 10-item production system, $c_{1,P} = \ldots = c_{4,P} = 100, c_{5,P} = 1000, c_{6,P} = \ldots = c_{9,P} = 100, c_{10,P} = 1000$.

A set-up slot is reachable from every slot in the cycle and a production slot is only reachable from slots just after production slots of the same type or the set-up slot for that type.

The one step improvement policy is compared with the fixed cycle policy, the exhaustive base-stock policy (cf. [54] and [46]), and an adjusted exhaustive base-stock policy. This policy is slightly different from the exhaustive base-stock policy, because it skips the next item if the stock level of the next item is equal to its base-stock level. If none of the items has a shortfall, the machine is set up for the next item. The exhaustive base-stock policy is the most studied production strategy in multi-item production systems. There exist other production strategies, of which the gated base-stock policy is the most well-known policy. Besides the fact that this strategy is harder to analyse than the exhaustive base-stock policy, the exhaustive base-stock policy often outperforms the gated base-stock policy. We observed this not only in the results presented in this section, but in all results that we obtained. It is worth noting that the same observation is made by Federgruen and Katalan in [37] for production systems with backlogged demand. By adjusting the exhaustive base-stock policy, the performance is slightly improved.

Tables 2.4 and 2.5 show the average costs per time unit for the fixed cycle strategy (FC), exhaustive base-stock policy (EXH), adjusted exhaustive base-stock policy (EXH*) and the one step improvement policy (1SI). The results in the tables are ordered according to the offered load $\rho$.

The order-up-to levels $S(1)$ up to $S(N)$ in the (adjusted) exhaustive base-stock policy are determined in the following, heuristic way, which is similar to the procedure to find $g_1, \ldots, g_N$ described in the previous section. A vector with base-stock levels $S^{(0)}$ is defined and set equal to 1. For every item $i$, the average costs are determined with a simulation study for $S + e_i$, i.e. all values of $S$ remain the same, except $S(i)$ which is increased by one. Among these $N$ new vectors with base-stock levels, the one with the lowest expected costs is chosen. This vector is denoted by $S^{(1)}$. Following the same procedure with $S^{(1)}$ as input, $S^{(2)}$ is found, which is used to find $S^{(3)}$ and so on until $S^{(N)}$ is found. $S^{(0)}$ is updated with the best vector among $S^{(1)}, \ldots, S^{(N)}$ if one of them gives lower costs than $S^{(0)}$. Based on this new value of $S^{(0)}$, the new vectors $S^{(1)}$ up to $S^{(N)}$ are found and $S^{(0)}$ can be updated again. These steps are repeated until no cost reduction is obtained.

For every set of base-stock levels, the expected costs per time unit are found with a simulation study. The reason for this is that the performance of the exhaustive base-stock control is numerically intractable if $N$ gets large. The length of the last simulation run is 25 million time slots, so that the calculated average costs are more accurate.

It is seen that the adjusted exhaustive base-stock policy always outperforms the exhaustive base-stock policy. But the one step improvement policy also outperforms the exhaustive base-stock policy, and if $\rho$ is high, it also outperforms the adjusted exhaustive base-stock policy.

Even better results are obtained if the variance of the demand processes is higher and the system has to be more responsive to changes in demand.

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 16.63 | 14.45 | 13.05 | 13.57 |
| 0.6 | 19.26 | 16.79 | 15.31 | 15.79 |
| 0.7 | 22.78 | 19.29 | 18.44 | 18.85 |
| 0.8 | 26.11 | 22.95 | 22.10 | 22.11 |
| 0.9 | 30.59 | 27.34 | 27.28 | 26.55 |

$$\lambda = (0.25\rho, 0.15\rho, 0.10\rho, 0.25\rho, 0.15\rho, 0.10\rho)$$
$$c_{1,I} = c_{2,I} = \ldots = c_{6,I} = 1,$$
$$c_{1,P} = c_{2,P} = \ldots = c_{6,P} = 100.$$

**Table 2.4:** A 6-item production system, with Poisson demand

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 26.64 | 23.61 | 20.33 | 22.18 |
| 0.6 | 31.45 | 26.67 | 24.63 | 26.19 |
| 0.7 | 36.06 | 31.51 | 28.88 | 29.80 |
| 0.8 | 41.60 | 36.53 | 35.16 | 34.68 |
| 0.9 | 48.17 | 42.61 | 42.12 | 41.90 |

$$\lambda = (0.1\rho, 0.1\rho, 0.1\rho, \ldots, 0.1\rho),$$
$$c_{1,I} = c_{2,I} = \ldots = c_{10,I} = 1,$$
$$c_{1,P} = c_{2,P} = c_{3,P} = c_{4,P} = 100, c_{5,P} = 1000,$$
$$c_{6,P} = c_{7,P} = c_{8,P} = c_{9,P} = 100, c_{10,P} = 1000.$$

**Table 2.5:** A 10-item production system, with Poisson demand

### Variance of the demand processes

The results in Tables 2.6 and 2.7 are based on the same parameter settings as in Tables 2.4 and 2.5, only the demand distributions are different. For each type, demand arrives according to the following compound Poisson process. Batches arrive according to a Poisson process with intensity $\frac{\lambda}{2}$ and have size 1 with probability $\frac{2}{3}$ and size 4 with probability $\frac{1}{3}$. Thus the variance of each demand process is increased, while the average number of arrivals per time unit remains the same.

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 28.31 | 26.60 | 25.57 | 25.88 |
| 0.6 | 31.88 | 29.33 | 28.22 | 28.10 |
| 0.7 | 35.63 | 32.28 | 30.81 | 30.70 |
| 0.8 | 39.57 | 35.61 | 34.14 | 34.06 |
| 0.9 | 43.84 | 39.44 | 38.09 | 37.86 |

$$\lambda = (0.25\rho, 0.15\rho, 0.10\rho, 0.25\rho, 0.15\rho, 0.10\rho)$$

**Table 2.6:** A 6-item production system, with compound Poisson demand

For both the 6-item production system and the 10-item production system, the

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 46.12 | 43.46 | 40.57 | 41.44 |
| 0.6 | 51.95 | 49.25 | 45.85 | 47.09 |
| 0.7 | 57.58 | 53.39 | 50.96 | 50.90 |
| 0.8 | 63.58 | 57.70 | 55.10 | 54.00 |
| 0.9 | 70.11 | 63.37 | 60.64 | 58.78 |

$\lambda = (0.1\rho, 0.1\rho, 0.1\rho, \ldots, 0.1\rho)$,

**Table 2.7:** A 10-item production system, with compound Poisson demand

dynamic policy gives lower expected costs than the (adjusted) exhaustive base-stock policy for $\rho \geq 0.7$. In the 6-item production system, the one step improvement policy also outperforms the adjusted exhaustive base-stock policy for $\rho = 0.6$, while this is not the case in the 10-item production system. The reason for this might lie in the fact that the cycles in the 10-item production system are longer, which gives a lower coefficient of variation of the demand that arrives in one cycle.

### Longer set-up times

In many production systems, for example the glass manufacturing system described in Fransoo et al. [42], set-up times take considerable time, which means a loss of capacity. Therefore, the system should not switch too often, especially not to items which have enough products on stock. To see what the effect of longer set-up times is on the performance of the different production strategies, we increase the lengths of the set-up times.

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 18.78 | 17.11 | 15.80 | 16.09 |
| 0.6 | 22.20 | 20.03 | 19.11 | 19.25 |
| 0.7 | 25.98 | 23.77 | 23.27 | 23.04 |
| 0.8 | 30.48 | 28.11 | 27.80 | 27.64 |
| 0.9 | 35.48 | 33.52 | 33.46 | 32.95 |

**Table 2.8:** A 6-item production system, with Poisson demand, $T^S = 2$

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 30.30 | 26.47 | 25.48 | 26.18 |
| 0.6 | 35.53 | 32.73 | 31.23 | 30.96 |
| 0.7 | 41.52 | 37.90 | 37.36 | 37.20 |
| 0.8 | 47.99 | 44.36 | 44.02 | 42.95 |
| 0.9 | 55.58 | 51.65 | 51.57 | 50.46 |

**Table 2.9:** A 10-item production system, with Poisson demand, $T^S = 2$

The results in Tables 2.8 and 2.9 are based on production systems with Poisson distributed demand. It now takes two time units to switch from one product type to another, while the production times are still equal to one time unit. It is expected that the dynamic one step improvement approach anticipates on these longer set-up times and is able to save some capacity in the sense that it does not switch to an item if it is not really necessary. Whereas the adjusted exhaustive base-stock policy only skips an item if the stock level equals the base-stock level, the dynamic policy will also skip an item if the stock level is somewhat less.

The dynamic policy performs best for $\rho \geq 0.7$ in the 6-item production system, while for the 10-item production system this is already the case for $\rho \geq 0.6$. This observation is explained by the fact that in the 10-item production system the machine spends more time on switching during one cycle than in a 6-item production system. Therefore, the actual load in the 10-item production tends to be higher than in the 6-item production system.

For systems with both longer set-up times and compound Poisson distributed demand, the results in Tables 2.10 and 2.11 are obtained. The compound Poisson distributions are the same as the ones used for Tables 2.6 and 2.7.

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 29.49 | 28.07 | 26.88 | 26.86 |
| 0.6 | 33.63 | 31.46 | 29.94 | 30.06 |
| 0.7 | 37.72 | 35.16 | 33.59 | 33.72 |
| 0.8 | 42.13 | 39.02 | 37.87 | 37.78 |
| 0.9 | 47.05 | 43.58 | 42.65 | 42.46 |

**Table 2.10:** A 6-item production system, with compound Poisson demand, $T^S = 2$

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 48.28 | 46.55 | 42.38 | 43.95 |
| 0.6 | 54.60 | 52.00 | 48.91 | 49.08 |
| 0.7 | 60.96 | 57.33 | 54.64 | 54.46 |
| 0.8 | 68.08 | 63.06 | 61.00 | 59.52 |
| 0.9 | 75.48 | 69.70 | 68.27 | 65.58 |

**Table 2.11:** A 10-item production system, with compound Poisson demand, $T^S = 2$

In the 6-item production system, the difference between 1SI and EXH* is small. In the 10-item production system, however, the difference is considerable for $\rho \geq 0.7$.

Summarizing the results so far, we see that particularly for systems with higher loads and more random demand, the one step improvement policy outperforms the exhaustive base-stock policy. Apparently, the fact that the (adjusted) exhaustive base-stock policy does not react to all stock levels becomes a problem if the load on the system is high. If there is a high probability that one or more stock levels quickly decrease, the system should be able to react. Therefore, the performance of the one step improvement policy is very well if the load on the system is high or if the demand is stochastic.

### The production order

If all set-up times are equal, like in all examples in this section, the order of production is not important for the performance of the fixed cycle policy. The reason for this is that the production order does not influence the length of a production or vacation period once the number of production times for each item is set. Because the production period of an item may start in another slot if the production order is changed, the relative value function for this item might be shifted as well. Because the structure of the fixed cycle remains the same, the structure of the relative value function per item also remains the same. However, for different production orders, the *sum* of the individual relative value functions is different.

This effect is illustrated in Figures 2.2, 2.3 and 2.4. Figures 2.2 and 2.3 show the individual relative value functions for two empty systems with the same parameter settings, but a different production order. The first graph shows the relative values for the production order of Table 2.4, while the second graph shows the relative values for the production order of Table 2.12. The parameters in the two systems are the following: Both production and set-up times are of unit length. The demand processes are all Poisson, with $\lambda = (0.125, 0.075, 0.05, 0.125, 0.075, 0.05)$ for system 1 and $\lambda = (0.125, 0.125, 0.075, 0.075, 0.05, 0.05)$ in system 2. The holding costs are all equal to 1, penalty costs are all equal to 100 and the lengths of the production periods in the fixed cycles are $g = (3, 2, 1, 3, 2, 1)$ for the first system, and $g = (3, 3, 2, 2, 1, 1)$ for the second system.

It is seen that 4 of the 6 individual relative value functions are shifted and therefore, the sum of the relative values is different (but not shifted!), which is shown in the third graph. Because of the difference in the relative value function, we also expect a difference in performance of the one step improvement policy.

Tables 2.12 and 2.13 show the performance of the one step improvement policy for a system with the same parameter settings as in Tables 2.4 and 2.6 respectively, but with a production order as in Figures 2.2 and 2.3. The different production order also has an effect on the individual product flows in the (adjusted) exhaustive base-stock policies, which might lead to lower or higher average costs. The following example illustrates one of the effects of a different order of production on the process at product flow 1.

Consider the 6-item production system of Figure 2.4 and number the items such that $\lambda = (0.125, 0.125, 0.075, 0.075, 0.05, 0.05)$. The order of production in the first system is then $1, 3, 5, 2, 4, 6$, while in the second system the order is $1, 2, 3, 4, 5, 6$. Further assume that the stock levels of all items are equal to the base-stock levels of these items. The time period between two successive production periods of item 1 consists of set-up times and production periods of other items. The length of each of the production periods depends on the production order. On average, the production periods of the items with a high demand rate are longer than the other

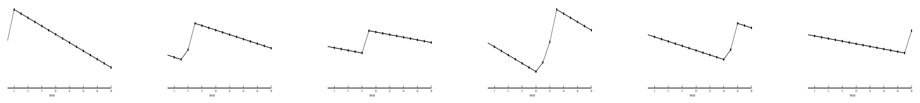


**Figure 2.2:** Individual relative values for system 1

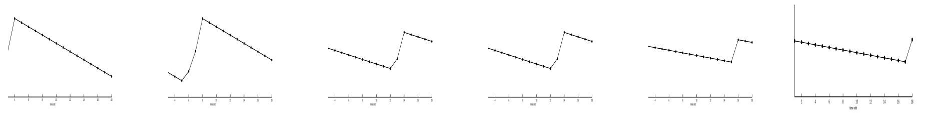


**Figure 2.3:** Individual relative values for system 2

6
8
10
12
14
16
18
time slot
6
8
10
12
14
16
18
time slot

**Figure 2.4:** Relative value functions for 6-item production systems

production periods. If item 2, with $\lambda = 0.125$, is the last item to produce before the machine returns to item 1, it takes more time before the machine is set-up for item 2. Therefore, more demand can arrive during this time. The production period for item 2 will thus, in distribution, take longer than if item 2 is the next item to produce.

So a different order of production has some effect on the different processes at the different product flows. Therefore, results for EXH* are obtained for the different production orders as well. The results with the new production order are shown in the tables on the left, the results from Tables 2.4 and 2.6 are shown in the tables on the right. The standard deviation for the results in Table 2.13 is at most 0.01, so for many examples, the difference in average costs is significant. It is seen that, although the differences are small, for the system with Poisson demand, the performance of 1SI seems best for the first choice of the production order. On

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 16.63 | 14.45 | 13.05 | 13.56 |
| 0.6 | 19.26 | 16.78 | 15.30 | 15.81 |
| 0.7 | 22.78 | 19.29 | 18.44 | 18.90 |
| 0.8 | 26.11 | 22.95 | 22.10 | 22.18 |
| 0.9 | 30.59 | 27.33 | 27.05 | 26.59 |

$\lambda = (0.25\rho, 0.25\rho, 0.15\rho, 0.15\rho, 0.10\rho, 0.10\rho)$

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 16.63 | 14.45 | 13.05 | 13.57 |
| 0.6 | 19.26 | 16.79 | 15.31 | 15.79 |
| 0.7 | 22.78 | 19.29 | 18.44 | 18.85 |
| 0.8 | 26.11 | 22.95 | 22.10 | 22.11 |
| 0.9 | 30.59 | 27.34 | 27.28 | 26.55 |

$\lambda = (0.25\rho, 0.15\rho, 0.10\rho, 0.25\rho, 0.15\rho, 0.10\rho)$

**Table 2.12:** A 6-item production system, with Poisson demand

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 28.31 | 26.61 | 25.56 | 25.85 |
| 0.6 | 31.88 | 29.33 | 28.22 | 28.18 |
| 0.7 | 35.63 | 32.28 | 30.77 | 30.64 |
| 0.8 | 39.57 | 35.60 | 34.02 | 34.10 |
| 0.9 | 43.84 | 39.44 | 38.10 | 37.81 |

$\lambda = (0.25\rho, 0.25\rho, 0.15\rho, 0.15\rho, 0.10\rho, 0.10\rho)$

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 28.31 | 26.60 | 25.57 | 25.88 |
| 0.6 | 31.88 | 29.33 | 28.22 | 28.10 |
| 0.7 | 35.63 | 32.28 | 30.81 | 30.70 |
| 0.8 | 39.57 | 35.61 | 34.14 | 34.06 |
| 0.9 | 43.84 | 39.44 | 38.09 | 37.86 |

$\lambda = (0.25\rho, 0.15\rho, 0.10\rho, 0.25\rho, 0.15\rho, 0.10\rho)$

**Table 2.13:** A 6-item production system, with compound Poisson demand

the other hand, the system with compound Poisson distributed demand gives the impression that it is quite random which of the two production orders is best.

Further, the average costs for the (adjusted) exhaustive base-stock policy are in some cases also significantly different for a different order of production. Unfortunately, it is not possible to state that if EXH* performs best for a specific order of production, the one step improvement policy performs best for that production order too. A counter example is found in Table 2.13 for $\rho = 0.8$, where EXH* outperforms EXH* in Table 2.6, but 1SI of Table 2.13 does not outperform 1SI of Table 2.6. For this counter example, the costs for the adjusted exhaustive base-stock policy are lower than for any of the one step improvement policies in Tables 2.6 and 2.13. But for $\rho = 0.6, 0.7$ and 0.9, the one step improvement policy does outperform the adjusted exhaustive base-stock policy. This is remarkable, but we see no explanation for this observation.

For the system studied in Tables 2.4 and 2.12, results are also obtained for the production orders with $\lambda = (0.25\rho, 0.25\rho, 0.10\rho, 0.10\rho, 0.15\rho, 0.15\rho)$ and $\lambda = (0.25\rho, 0.10\rho, 0.15\rho, 0.25\rho, 0.10\rho, 0.15\rho)$ in Table 2.14. It is seen that, after comparing all four different orders of production, the production order in the table on the right hand side gives the best results for the one step improvement policy for $\rho \leq 0.8$ and the production order in the table on the left hand side is best for $\rho = 0.9$.

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 16.63 | 14.45 | 13.05 | 13.47 |
| 0.6 | 19.26 | 16.78 | 15.30 | 15.76 |
| 0.7 | 22.78 | 19.28 | 18.44 | 18.78 |
| 0.8 | 26.11 | 22.94 | 22.10 | 22.08 |
| 0.9 | 30.59 | 27.34 | 27.05 | 26.72 |

$\lambda = (0.25\rho, 0.10\rho, 0.15\rho, 0.25\rho, 0.10\rho, 0.15\rho)$

| $\rho$ | FC | EXH | EXH* | 1SI |
|---|---|---|---|---|
| 0.5 | 16.63 | 14.45 | 13.05 | 13.50 |
| 0.6 | 19.26 | 16.78 | 15.30 | 15.82 |
| 0.7 | 22.78 | 19.27 | 18.44 | 18.95 |
| 0.8 | 26.11 | 22.94 | 22.10 | 22.17 |
| 0.9 | 30.59 | 27.34 | 27.05 | 26.62 |

$\lambda = (0.25\rho, 0.25\rho, 0.10\rho, 0.10\rho, 0.15\rho, 0.15\rho)$

**Table 2.14:** A 6-item production system, with Poisson demand

Summarizing the results on the different production orders, we can conclude that a different production order can lead to significantly lower or higher costs. However, it is not possible to get a good intuition for the best production order from the results obtained here.

### An extra production period

Another way of changing the production order is by adding an extra production period for items with a high demand rate or high penalty costs. In this way, the vacation periods for these items get shorter and one might need lower safety stocks. On the other hand, the load on the system gets higher, because more time is spent on switching per cycle, which results in higher safety stocks. Production systems with and without an extra production period for the first item are compared in Table 2.15. In this table, the demand rates are all equal, but the distributions of the demand processes are different. First, a Poisson process is considered, then two compound Poisson processes are studied. The first compound Poisson process is as in Tables 2.6, 2.7, 2.10, 2.11 and 2.13. The second compound Poisson process has batch arrivals with rate $\lambda/3$. These batches have size 9 with probability 1/4 and size 1 with probability 3/4. So the second compound Poisson process has a higher variance and thus a higher coefficient of variation than the first compound Poisson process. The holding costs are all equal to 1, the penalty costs are all equal to 100. The fixed cycle, adjusted exhaustive base-stock and one step improvement policy with an extra production period for item 1 are denoted by FC2, EXH*2 and 1SI2 respectively. The order of production is then $1, 2, 3, \mathbf{1}, 4, 5$ instead of $1, 2, 3, 4, 5$.

| Demand process | FC | FC2 | EXH* | EXH*2 | 1SI1 | 1SI2 |
|---|---|---|---|---|---|---|
| Poisson | 18.29 | 18.24 | 14.70 | 14.70 | 14.98 | 14.75 |
| Compound Poisson 1 | 29.57 | 29.07 | 25.64 | 25.62 | 26.13 | 25.78 |
| Compound Poisson 2 | 40.43 | 40.10 | 38.33 | 38.29 | 38.49 | 38.31 |

$\lambda = (0.4, 0.1, 0.05, 0.05, 0.1), h = 1, p = 100$

**Table 2.15:** A 5−item production system with different demand processes

It is seen that, for the three systems studied in Table 2.15, the systems where item 1 gets two production periods per cycle results in lower costs per time unit for both the adjusted exhaustive base-stock policy and the one step improvement policy.

In Tables 2.16 and 2.17, we look at the effect of an extra production period for an item with relatively high penalty costs. The demand rates are now all equal, but the penalty costs for the first item are much higher than for the other items. The production orders $1, 2, 3, 4, 5$ and $1, 2, 3, \mathbf{1}, 4, 5$ are compared for the fixed cycle, adjusted exhaustive base-stock strategy and for the 1SI policy. It is seen that none of the fixed cycles gives lower costs with the extra production period. However, the one step improvement policy performs better with the extra production period for the Poisson demand process and the second compound Poisson demand process. On the other hand, for the first compound Poisson demand process, the production order $1, 2, 3, 4, 5$ gives the lowest costs for the one step improvement policy. These observations show that it is difficult to get a good intuition for the right order of

| Demand process | FC | FC2 | EXH* | EXH*2 | 1SI1 | 1SI2 |
|---|---|---|---|---|---|---|
| Poisson | 16.40 | 16.65 | 12.66 | 12.66 | 13.72 | 13.30 |
| Compound Poisson 1 | 29.68 | 29.75 | 25.30 | 25.28 | 25.22 | 25.36 |
| Compound Poisson 2 | 44.35 | 44.53 | 42.59 | 42.55 | 42.69 | 42.35 |

$\lambda = (0.1, \ldots, 0.1), h = 1, p = (500, 100, 100, 100, 100)$

**Table 2.16:** A 5−item production system with different demand processes

| Demand process | FC | FC2 | EXH* | EXH*2 | 1SI1 | 1SI2 |
|---|---|---|---|---|---|---|
| Poisson | 24.88 | 25.05 | 20.28 | 20.28 | 20.40 | 20.06 |
| Compound Poisson 1 | 39.92 | 39.97 | 33.03 | 33.04 | 32.71 | 32.85 |
| Compound Poisson 2 | 59.20 | 59.20 | 54.99 | 54.97 | 52.47 | 52.33 |

$\lambda = (0.15, \ldots, 0.15), h = 1, p = (500, 100, 100, 100, 100)$

**Table 2.17:** A 5−item production system with different demand processes

production, because an increase in the variance of the demand processes can have both a negative and a positive effect on the performance of the one step improvement policy if an extra production period is added to the fixed cycle.

Besides the parameter settings that are changed for the results in this section, one can also look at the lengths of the production times, the distributions of the production and set-up times and item dependent set-up times. Unfortunately, the simulations that are needed to obtain accurate results can be time consuming. This is no problem if only a few examples need to be studied. But especially the time that is needed to find the optimal base-stock levels for the exhaustive base-stock strategy becomes a problem if one wants to look at a large number of examples. Therefore, the effect of the lengths of the production times and other parameter settings are left for future research.

## 2.7 Conclusion

A dynamic policy for the control of a production system is obtained by performing an improvement step on a fixed cycle strategy. This fixed cycle strategy has a special structure which allows for a decomposition of the relative value function that is used to perform the improvement step. Therefore, the dynamic policy can be obtained for systems with a large number of product types as well, while in these situations the optimal policy is intractable.

The performance of the new strategy is compared with the exhaustive and adjusted exhaustive base-stock policy. The one step improvement policy seems to be very well suited for the dynamic control of more random systems, systems with a high load, large set-up times and different demand rates or penalty costs. Furthermore, the one step improvement strategy performs well if the number of different product types is large, which is exactly the area where we were looking for a good dynamic strategy. It is also seen that the order of production has some influence on the performance of both the exhaustive base-stock strategy and the one step improvement policy. Although the performance of the fixed cycle policy is equal for all production orders in which each item gets one production period and set-up times are all equal, the relative value function for each of the fixed cycles is different which leads to a different performance of the one step improvement policy. It was seen that spreading the items with the highest demand rate equally has a positive effect on the performance of the dynamic policy. Further, ordering the items according to decreasing demand rates or penalty costs also has a positive effect on the cost reduction.

If for one item the demand rate is relatively high, it might be rewarding to insert an extra production period for this item in the fixed cycle. Even though the average costs for the fixed cycle policy do not necessarily decrease with this extra production period, the one step improvement policy can profit from this extra production period, if the demand is very random. Similarly, if for one item the penalty costs are relatively high and the demand is very random, the average costs for the one step improvement policy can decrease with an extra production period for this item.

The next chapter discusses the one step improvement approach based on a fixed cycle policy in a multi-item production system with 2 machines.

## Chapter 3

# Lost sales: 2 machines

For the single machine multi-item production system, the one step improvement approach gave nice results in systems with higher load and more random demand. A more complex, but also very practical model is a similar system with 2 or more machines. The current chapter focusses on this multiple machine multi-item production problem and builds on the work in Chapter 2. As before, the production system can be translated into a queueing system. This queueing system has multiple servers and multiple queues. Because of the complexity of multi-server queueing systems, there are only a few studies on production systems with multiple machines so far and they all concern systems with backlog instead of lost sales.

For systems with backlog, Morris and Wang [64] studied a queueing system with multiple servers that are assumed to visit the queues independently of each other, but according to a fixed cyclic order. If these orders are equal, it turns out that the servers tend to cluster. In order to avoid this clustering effect, they suggest to use two different, 'dispersive' routes for the two servers. Levy et al. [58] propose reversed polling orders to prevent clustering of the servers. Further, they suggest to reverse the directions of two servers when they collide.

Browne and Weiss [20] is one of the few studies in which multiple servers are coupled in a multi-queue system. In their study, the servers visit the queues together and they extend the analysis of a multi-server queueing system to a polling model with $c$ coupled servers. Borst [13] also looks at polling systems with coupled servers and focusses on the distributional results of the waiting time and queue lengths at polling epochs, where each queue gets gated or exhaustive service and service times are exponential or deterministic. Van der Mei and Borst [84] analyse a more general polling model with a power-series algorithm, with which the waiting times and queue lengths of many of these multi-queue models can be analysed and minimized. More recently, de Haan [30] introduced an exhaustive exponential time limited strategy, in which two servers serve each queue during a maximum time or until the queue is empty. This maximum time is stochastic and sampled from an exponential distribution.

For a system with two queues, Osogami et al. [66] look at a system with one server for a beneficiary queue and one server for a donor queue. The server of the donor queue switches to the beneficiary queue if the donor queue is empty and the number of customers in the beneficiary queue is above a certain threshold level. The server of the donor queue switches back as soon as the number of customers in the donor queue exceeds another threshold level. The mean sojourn time of the customers is minimized and optimal values of the two threshold levels are derived. The analysis can be extended to a system with multiple donor queues, but a model with multiple beneficiary queues seems to be too complex.

As the above studies point out, the analysis and optimization of large multi-server polling models is very complex, due to the lack of structure and curse of multi-dimensionality. And just as for the single machine problem, the analysis for a system with two machines is more complex if we consider a system with lost sales. Although the number of states in such a system is smaller (there are no queues), the different base-stock levels influence the performance measures of all other product flows, as was already seen in the previous chapter. Furthermore, a generating function approach, which is also used in Borst [13] and Browne and Weiss [20], does not lead to explicit expressions for a system with lost sales.

The corresponding queueing model of the production system with lost sales is a polling model with two servers and a finite buffer for each queue. For queueing models with buffers, analytic results depend on the buffer sizes of all queues. Marsan et al. [60, 61, 62] look at a multiserver queueing system with finite buffers and present a method to obtain exact numerical values for performance measures like customers' waiting times, but restrict the system to be small enough in the total number of possible states. The same problem arises if one tries to find the optimal production strategy (via an MDP approach) in a production system with a (too) large number of product types. The complexity of the system grows even faster in the number of product types than the complexity of the production system with a single machine. And if the complexity gets too large, the optimal policy becomes intractable.

However, the construction of a dynamic policy with a one step improvement approach could be a way to deal with the multi-dimensionality problem that is also encountered in the backlog systems. Furthermore, the complication with the base-stock levels is then also solved, because of the independence of the product flows when a fixed cycle strategy is used. This leaves the question of how to extend the one step improvement approach so that it can be applied in a system with multiple machines. The current chapter answers this question for two machines.

## 3.1 Model

Just as in Chapter 2, we consider $N$ product types for which demand arrives according to (compound) Poisson processes. All types are made to stock and for each item, the one step transition costs are as in Equation (2.1). To make these product types one uses two identical machines. In the previous chapter, set-up and production times can be of different lengths. If that is also the case in the production system with two machines, decisions are not (always) taken simultaneously. So if on one machine, a set-up or production time is finished and the other machine is still busy with a production or set-up, one needs to know how much time is needed to finish that production or set-up in order to calculate the future expected costs for that machine. So unless all production and set-up times are exponentially distributed, one has to keep track of the time that has passed since the last decision moment for each machine, which makes the analysis more complicated. Thus, in this study it is assumed that all set-up times and all production times of the two machines are deterministic and equal to 1, so that decisions for the two machines are always taken simultaneously. The system is controlled as follows. Every time unit (or decision moment) the state of the system, i.e. the inventories of all product types and the items set-up at the two machines, is inspected and then a combined decision is taken about what to do on the two machines.

As we have already seen in Chapter 2, if the number of product types is somewhat larger, the curse of dimensionality makes it impossible to determine the optimal production and inventory strategy. For the one-machine problem we already developed a one step improvement policy in the previous chapter. For a production system with two machines, a similar approach is now used, which works as follows.

For both machines, a fixed cycle strategy is constructed so that the complete system can be analysed as $N$ independent subsystems. The product types are divided into two disjoint sets, one for each machine. The number of product types in the two sets are denoted by $N_1$ and $N_2$, for machines 1 and 2 respectively. Then, a fixed cycle scheme is found for each of the two sets. Like in the previous chapter, the state of the system with a fixed cycle control is described by the tuple $(n_1, n_2, k_1, \ldots, k_N)$, where $k_i$ is the inventory of type $i$ and $n_1$ and $n_2$ the slots within the cycles. At every decision moment, one may decide to jump to another slot in the fixed cycle. In that way, two separate one step improvement policies are constructed. But for each item, it is allowed to produce it on both machines. So both machines can also produce the product types that are normally (i.e. in the fixed cycle policy) produced on the other machine and therefore, it is possible to start a so called visit period to one of the product types of the other machine. That product type is called the *visited* product type. The fixed cycle of the visiting machine is interrupted, which can be seen as a breakdown or vacation period. For the other machine, it is assumed that the fixed cycle continues and therefore, only the visited product type is influenced by the visit period. If such visit periods are allowed, an extra state variable is needed for each machine to indicate which product type is currently set-up on that machine and how many slots of the visit period are left (zero if there is no visit period).

For the new visit decisions, relative values must be calculated. Fortunately, the decomposition property mentioned in Property 1.2 still holds, so the relative values can be determined per product type. Furthermore, the relative values for the visit decisions can be expressed in terms of the relative values of the fixed cycles. The exact expressions for the relative values are given in Section 3.2.

The constructed production strategy with visit periods will be called a *combined* one step improvement policy instead of the one step improvement policy that still refers to the production strategy discussed in the previous chapter. The two policies are compared in Section 3.4.

Let us now first discuss the costs and transition probabilities.

### 3.1.1 Costs and transition probabilities

The costs and transition probabilities are basically the same as in the previous chapter. However, new assumptions in the production model lead to some slight changes in the notation for the costs and transition probabilities. For example, new transition probabilities need to be introduced, because a visited product type can be produced by two machines simultaneously.

First, the slot index is omitted from the one step transition costs. The reason for that is that it is assumed that all time slots are of unit length. Therefore, the one step transition costs no longer depend on the slot type. So we can define $c^{(i)}(k)$ as the one step transition costs for type $i$ if the current stock level of that type equals $k$. Using the notation of Equation (2.1), we denote $c_i(k) = c_i(k, 1)$ as the expected costs for a slot starting with an inventory of $k$.

Similarly, $D(i)$ is defined as the random variable denoting the demand for type $i$ in a slot (time unit) and $a_i(k) = P(D(i) = k)$. For the transition probabilities $p_{i,n}(k, k')$, Equations (2.2) up to (2.7) still hold for the case that the other machine is not set-up for type $i$. In these equations, $D_n(i)$ must be read as $D(i)$ and $a_{i,n}(k)$ as $a_i(k)$.

However, if the other machine is set-up for type $i$, it is – at least theoretically – possible that both machines work on this type simultaneously. Therefore, new transition probabilities have to be defined. Now, let $p^+_{i,n}(k, k')$ denote the transition probability that $I(i)$ changes from $k$ to $k'$ during slot $n$, given that the next slot on the other machine is reserved for a production of item $i$. Then, if $k \geq S(i) - 1$ or $n$ is not a production slot for item $i$, $p^+_{i,n}(k, k') = p_{i,n'}(k, k')$, with $n'$ a production slot of item $i$. If $n$ is a production slot for item $i$,

$$p^+_{i,n}(k, k') = a_i(k - k' + 2), \quad 0 < k < S(i) - 1\,, \ k' = 3, \ldots, k$$

and

$$p^+_{i,n}(k, 2) = P(D(i) \geq k) = 1 - \sum_{j=0}^{k-1} a_i(j), \quad 0 < k < S(i) - 1.$$

With the one step transition costs and probabilities, the relative values can be defined.

## 3.2 Relative values

### 3.2.1 One machine

The relative values of the fixed cycle policies are already calculated in the previous chapter. With Equation (2.11), one easily obtains the (approximate) relative values $\hat{r}_i(n, k_i)$ for each product type $i$ with stock level $k_i$ for time slot $n$, for instance with successive approximations. The total relative value for slot $n$ for one machine is then equal to $r^{\hat{(m)}}(n, \underline{k}) = \sum_{i=n_m}^{N_m} \hat{r}_i(n, k_i)$. Here, $\underline{k} = (k_1, \ldots, k_N)$, $m$ the number of the machine and $k_i$ the inventory for type $i$.

In the improvement step for one machine, one looks for the best reachable $n'$, that is the reachable slot within the cycle that minimizes $r^{\hat{(m)}}(n', \underline{k})$.

### 3.2.2 Two machines

Recall that at the start, each product type has been allocated to one of the machines. In the combined improvement step, each machine can start a visit period to another product type. For such a decision, a relative value must be calculated. This relative value can be found with the relative values for the fixed cycle policy. In order to limit the number of possible decisions, we do not allow the two machines to visit a product type at the same time. So at most one of the two machines is allowed to work on a product type that normally is not produced on it. Below we show how the decision to help is taken.

### 3.2.3 Notations

We will use an index $P$ to indicate that a product type is produced on its *Preferred* machine (the one it is normally produced on) and $V$ if is it (also) produced on the other machine, the non-preferred or *Visiting* one. Now we can write $r_i^P$ for the ordinary relative values obtained for the fixed cyclic scheme for the machine product type $i$ is normally produced on. The values $r_i^V$ will denote the relative values for a product type that is produced on its non-preferred machine.

In case one of the machines starts a visit period other relative values are needed, for the product type that is visited as well as for the product types that are produced on their preferred machine, but now are interrupted by an intruding product type from the other machine. For the product types that are interrupted, only the (residual) duration of the interrupt matters.

#### Interrupted product types

Let us denote $r_i^P(l, k; n)$ as the relative value for product type $i$ that is produced on its preferred machine, which now is interrupted for the next $l$ time units, given that there are $k$ items of type $i$ left and assuming that at the end of the interrupt the cycle resumes in slot $n$ and the visiting machine returns to its own fixed cycle.

These relative values are computed recursively, as follows

$$r_i^P(l,k;n) = c_i(k) + \sum_{k'} p_{i,0}(k,k')r_i^P(l,k';n-1) - c^{(i)} \ ,$$

with

$$r_i^P(l,k;0) = r_i^P(l,k)$$

and $c^{(i)}$ the average costs per time unit for product type $i$ in the fixed cycle policy.

### The interrupting product

For the product type that is causing the interrupt we have to distinguish between the situation that the first slot is needed for set-up and the case that it can be used for production. Also the slot in which the machine is interrupted is not relevant for this type. What is relevant is the slot the cycle of its preferred machine is in. We will use $n$ to denote this slot and the notations $r_i^V(n,k;l,S)$ and $r_i^V(n,k;l,P)$ with $S$ indicating the set-up slot and $P$ a production slot. Since only the first slot of the interrupt is needed for set-up we get the following recursions:

$$r_i^V(n,k;l,S) = c_i(k) + \sum_{k'} p_{i,n}(k,k')r_i^V(n,l;l-1,P) - c^{(i)} \ ,$$

$$r_i^V(n,k;l,P) = c_i(k) + \sum_{k'} p_{i,n}^+(k,k')r_i^V(n,l+1;l-1,P) - c^{(i)},$$

with

$$r_i^V(n,k;0,P) = r_i^P(n,k).$$

The values of $c^{(i)}$ and $r_i^P(n,k)$ can be found with successive approximations, as described in the previous chapter.

## 3.3 The combined improvement step

Now that the relative values are available, we can calculate the decisions in the combined one step improvement policy. In order to only consider the decisions that are possible, one has to distinguish between the following two situations.

1. In the first one all product types are allocated to their preferred machines. Given the slots the two cycles are in and the inventories for all product types one has to find the best decisions. Then there are two possibilities to consider.

   (a) All product types stay on their preferred machines. The optimal decision pair of time slots is computed per machine, just as in the case of the one-machine situation.

   (b) One of the product types interrupts the cycle of its non-preferred machine. Then there are four decision elements to consider.

      i. Which product type is interrupting,
      ii. How many items will be produced (this number plus 1 for the set-up is the duration of the interrupt),
      iii. Which slot is next for the preferred machine,
      iv. In which slot will the interrupted machine resume its cycle.

2. In the second situation there is an interrupt on one of the machines, i.e., the machine is set-up to produce a product type from the other machine. Then there are two options:

   (a) Terminate the interrupt and then treat the situation as if all product types are on their preferred machines.

   (b) Continue the interrupt for a number of slots. This requires three decisions:

      i. For how many slots to continue,
      ii. In which slot will the interrupted machine resume its cycle, and
      iii. Which slot is next for the preferred machine.

### 3.3.1 Computational complexity

With respect to the complexity one has to distinguish two aspects: 1) how much time is needed to compute a decision, but also, 2) how much time is needed to obtain a fairly good estimate of the performance of the strategy obtained by applying Howard's policy improvement step. For the latter preferably about a million slots need to be simulated. Let us investigate what is needed.

First note that the values $r_i^P(n,k;l)$ can be computed beforehand for every quadruple $i,k,l,n$. With 10 product types, a maximum value for $k$ of 20, $l$ at most 10 and $n$ at most 5 (the number of set-up slots), the number of such quadruples is about 10000. Computing all of them thus costs less than 1 second. The same holds for the values $\hat{r}_i^N(n,k;l,S)$ and $\hat{r}_i^N(n,k;l,P)$. So this computational work can be done in advance.

The next question is, how many decisions are possible in a state. Let us consider the possible situations sketched above.

With respect to 1(b)i, there are $N$ possible decisions, because each product type can interrupt the fixed cycle of its non-preferred machine. For 1(b)ii, there is only one fixed number of production slots in a planned visit period that we consider. In that way, there is only one option for this decision element. Once a visit has started, it is possible to change the length of the visit (at a future decision moment), but for now it is enough to know whether the expected cost reduction of the planned visit period is larger than the expected extra costs for the items that are interrupted. The number of planned production slots must be sufficiently large to compensate the set-up slots that are needed to start and end the visit period. On the other hand, the visit period should not be too long, taking the lost sales costs of the interrupted items into account. Looking at only one fixed number of production slots is probably not always optimal, so one might not start a visit period, while an (expected) cost reduction could be obtained with a visit period if another number of production slots was considered. Fortunately, it is not necessary to use exactly the optimal number of production slots to improve the two separate 1SI policies. Such an improvement can be obtained with any reasonable number of slots. It is also possible to consider several lengths of the visit period.

Concerning 1(b)iii, we only look at the set-up slots in the fixed cycle, because the machine first has to switch back to one of the preferred items before continuing its fixed cycle. This gives us $N_1$ or $N_2$ options (depending on which of the machines is the machine that is interrupted). The number of options for decision element 1(b)iv is the number of reachable slots for the interrupted machine, which is at most the number of slots in one cycle of that machine.

The options for decision element 2(a) are only the set-up slots of the corresponding machine, which comes down to a number of $N_1$ or $N_2$ options. With respect to decision element 2(b)i, we choose the best option among a finite number of values, where the minimum value is at least 1 (the option of zero slots is already treated in 2(a)). While in 1(b)ii, one has to look at a sufficiently large number of production slots during the visit period, here the option of adding just one production slot should always be considered. The reason for this is that no set-up time is inserted for this production slot, so there is no need to compensate the costs for this set-up time.

Concerning 2(b)ii, again only the set-up slots are considered, while for decision element 2(b)iii, every reachable slot is an option to continue the fixed cycle.

With these options for the different decision elements, the number of possible decisions tends to be very large. For example, in a system with 10 different product types, the number of possible decisions might well be in the order of one thousand. Finding the improved decision for a specific state will thus take in the order of a millisecond. A consequence of that is that, depending on the required accuracy, the simulation of the improved strategy might become time consuming. There are a few ways to reduce the number of possible decisions, for example with the following restrictions. If the total shortfall level of one of the machines is higher than the total shortfall level of the other machine or than a certain threshold level, it can not start a visit period. Further, if the shortfall level of one item is below a certain threshold level, it does not interrupt the fixed cycle of its non-preferred machine. It is also possible to restrict the system not to work on one product type with two machines simultaneously. In that case, it is also not possible to start a visit period of length $l$ if it is assumed that within $l$ slots, the same product type will be set-up on the other machine.

## 3.4 Results

The assumed length of a visit period, $l$ must be chosen in such a way that it is not too short and therefore too unattractive to start the visit period. On the other hand, if the number of production slots in the visit period is too high, the calculation of the relative value for the visit period takes too long, because the calculation time grows exponentially in $l$. For the results in this section, we take the value of $l$ equal to the minimum of 4 and $g_i$, the number of slots in a regular production period for item $i$. The value of $l$ is at most 4, so that the calculation time of the relative values is limited. Note that the actual length of the visit period is not necessarily equal to $l$, because it is possible to stop or continue the visit period at any slot boundary. Further, the remaining number of slots of a started visit period is the number of slots between zero and $\min(g_i, 4)$ that gives the minimum relative value. The fixed cycle slot that is executed after the visit period is assumed to be the switch-over slot that gives the minimum relative value.

At the end of the previous section, a number of restrictions is discussed to limit the number of possible decisions. Although these restrictions probably speed up the simulation runs, they are not used for the results in this section.

Further, we want to note that the division of the product types is not necessarily optimal. It is not our goal to look for the optimal sets of product types, but to get some insights into the performance of the C1SI compared to the two separate improvement steps. A set of examples is shown to illustrate the effect of different parameter settings on the performance of the C1SI and 1SI policies. Unfortunately, there was not enough time available to do a more extensive study.

#### Equal loads

In the production systems studied in Tables 3.1 and 3.2, the different items are divided over the two machines in such a way that the load on the machines is the same, 0.8. For these examples, two different demand processes are used. The first one is a Poisson demand process with intensity $\lambda$, the second one is a compound Poisson demand process. In the compound Poisson demand process, batches arrive according to a Poisson process with rate $\lambda/2$ and have size 1 with probability 2/3 and size 4 with probability 1/3. The combined improvement step is denoted by C1SI, the two separate improvement steps are denoted by 1SI. It is seen that with the combined one step improvement policy a significant cost reduction can be obtained compared to the 1SI policy. However, it is also seen from Table 3.1 that the combined improvement step does not necessarily lead to a better result than the two separate improvement steps. A natural question is in which cases the combined improvement policy outperforms the two separate improvement steps. This is an interesting question that is left open for future research.

| demand process | FC | EXH* | 1SI | C1SI |
|---|---|---|---|---|
| Poisson | 54.30 | 43.29 | 43.70 | 40.76 |
| Compound Poisson | 86.92 | 70.16 | 70.42 | 69.21 |

$\lambda = \{0.5, 0.10, 0.10, 0.05, 0.05, 0.4, 0.15, 0.10, 0.10, 0.05\}$,
$h = 1, p = \{300, 300, 300, 300, 300, 100, 100, 100, 100, 100\}$

**Table 3.1:** A 10−item 2 machine production system with different demand processes

| demand process | FC | EXH* | 1SI | C1SI |
|---|---|---|---|---|
| Poisson | 55.90 | 44.06 | 44.60 | 42.56 |
| Compound Poisson | 88.40 | 71.42 | 72.38 | 73.49 |

$\lambda = \{0.5, 0.10, 0.10, 0.05, 0.05, 0.4, 0.15, 0.10, 0.10, 0.05\}$,
$h = 1, p = \{100, 100, 100, 100, 100, 300, 300, 300, 300, 300\}$

**Table 3.2:** A 10−item 2 machine production system with different demand processes

### Different loads

If the load on machine 1 is high, while the load on machine 2 is low, the second machine visits machine 1 quite often. The obtained cost reduction compared to the two 1SI strategies is almost 7.5% for the case with Poisson demand and 3.68% for compound Poisson demand. In both cases, the combined improvement step even outperforms the adjusted exhaustive base-stock policy, while the two separate one step improvement policies perform worse than EXH*.

| demand process | FC | EXH* | 1SI | C1SI |
|---|---|---|---|---|
| Poisson | 37.02 | 29.05 | 31.14 | 28.82 |
| Compound Poisson | 60.88 | 52.28 | 52.48 | 50.55 |

$\lambda = \{0.15, 0.15, 0.15, 0.15, 0.15, 0.1, 0.1, 0.1, 0.1, 0.1\}$,
$h = 1, p = \{100, 100, 100, 100, 100, 100, 100, 100, 100, 100\}$

**Table 3.3:** A 10−item 2 machine production system with different demand processes

### Different costs

The examples in this subsection consider two machines with exactly the same demand distributions, but different penalty costs. So the loads on the two machines are equal, but a lost sale of an item on machine 1 is much more expensive than a lost sale of an item on machine 2. The results in Table 3.4 give the same picture as the results in Tables 3.3, where the combined one step improvement policy outperforms EXH*, while two separate 1SI policies give higher costs than EXH*. In all examples seen so far, the obtained cost reduction with the combined one step improvement policy is higher if the demand processes are more stochastic. Further, it is seen that better results are obtained if the loads on the different machines are different or the costs for the items on one machine are higher than the costs for the items on the other machine. However, it is often possible to construct two fixed cycles in such a way that the expensive items are equally spread over the machines or the loads on the two machines are similar.

| demand process | FC | EXH* | 1SI | C1SI |
|---|---|---|---|---|
| Poisson | 56.24 | 44.59 | 45.49 | 42.83 |
| Compound Poisson | 91.34 | 73.82 | 74.02 | 73.39 |

$\lambda = \{0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15\}$,
$h = 1, p = \{300, 300, 300, 300, 300, 100, 100, 100, 100, 100\}$

**Table 3.4:** A 10−item 2 machine production system with different demand processes

| demand process | FC | EXH* | 1SI | C1SI |
|---|---|---|---|---|
| Poisson | 38.82 | 28.78 | 32.36 | 31.97 |
| Compound Poisson | 72.97 | 58.73 | 61.52 | 60.95 |

$\lambda = \{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\}$,
$h = 1, p = \{500, 500, 500, 500, 500, 100, 100, 100, 100, 100\}$

**Table 3.5:** A 10−item 2 machine production system with different demand processes

## 3.5 Conclusion and further research

A dynamic control strategy is constructed which allows machines to visit product types that are normally only produced on another machine. This control strategy turned out to reduce the expected costs that are obtained with the one step improvement policy discussed in the previous chapter. Especially in the case of asymmetric costs, different demand rates and different loads, the C1SI performs well. A comparison with other strategies is needed to indicate in which cases the C1SI policy outperforms existing strategies like the reversed polling policy of de Levy et al. [58].

The one step improvement approach has now been applied in both a production system with a single machine and a production system with two machines. A logical next step would be to construct a similar dynamic production strategy in a system with three or more machines.

Chapter 4

# Backlog: An overview

The backlog model does not have the advantage that the state space is limited (as in the lost sales model), because there is no bound on the number of backlogged customers. On the other hand, backlogging has the advantage that, as we will show, it allows for a generating function approach to analyse the system. First, the system is translated into a queueing system by looking at the shortfall level of each product type. The shortfall levels are defined as in Chapter 1: $X(i) = S_i - I(i) + B(i), i = 1, \ldots, N$. So in the queueing system, each outstanding order in product flow $i$ is considered as a waiting customer in queue $i$. The values of $X(i)$ are all non-negative, because the stock level $I(i)$ can not exceed the maximum stock level $S_i$. Although the values of $S_i, i = 1, \ldots, N$ play an important role in the expected costs in the production system, they play no role in the queueing processes $X(1), \ldots, X(N)$. Therefore, the analysis of these queueing processes is the same for every set of base-stock levels.

Looking at the system from this queueing point of view, the production system is basically translated into a polling model, which is already discussed briefly in Chapter 1.

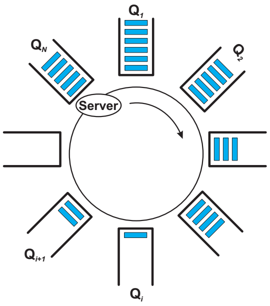


**Figure 4.1:** A polling model

## 4.1 Polling model

Figure 4.1 shows an example of a polling model, in which eight ($N$) queues, denoted by $Q_1, \ldots, Q_N$, are served by a single server. The stochastic arrival processes at the different queues are independent from each other and each customer requires a (stochastic) service time. These service times are also independent from each other, (per queue) identically distributed, and independent from the arrival processes. Switching from one queue to the next may require a (possibly stochastic) set-up time.

A time interval between two consecutive switches is called a visit period for the queue for which the server is then set-up. A vacation time is defined as a period in which the server is away. For every queue, a vacation time consists of the visits to the other queues and the switch-over times, see Figure 4.2. The number of customers served during a visit period of a certain queue depends on the number of customers present at that queue. But if the server spends more time at queue 1, during this time more customers arrive at queues 2 up to $N$, so the length of the vacation depends on the length of the visit period. The key complicating factor in the analysis of a polling system is this dependency between the lengths of the visit periods of the different queues. Furthermore, the lengths of two successive visit periods at the same queue are also dependent.

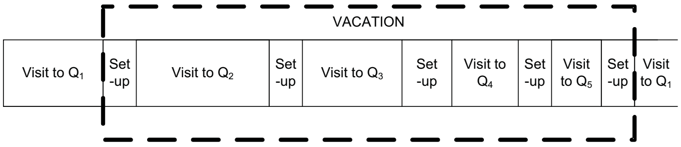


**Figure 4.2:** A vacation period of the first queue.

### Visit disciplines

It was already mentioned in the introduction of this thesis that polling models have been widely studied in the literature. In these polling models, a visit discipline determines which customers are served during a visit period to a certain queue and a service policy determines the order in which those customers are served. The visit and service policies may vary per queue. The most commonly used service policy is First-Come-First-Served and the most well-known visit policies are the exhaustive, gated and $k$-limited policies. These visit disciplines are equivalent to the exhaustive, gated and quantity-limited base-stock policies discussed in Chapter 1 for the production system. Just like in the lost sales model (see Chapter 2), the server serves a queue until it is empty according to the exhaustive visit discipline. According to the gated visit discipline, it serves exactly those customers present upon arrival of the server (i.e. a gate is placed behind the last customer if the server has just been set-up for $Q_i$). According to the $k$-limited visit discipline, the server serves a queue according to an exhaustive visit discipline, but switches earlier if $k$ customers are served. The exhaustive and gated visit disciplines belong to the so-called branching-type disciplines. If all queues are served according to a branching-type visit discipline, the system can be analysed in a quite elegant way (see Resing [71]) with probability generating functions (p.g.f.'s) and Laplace-Stieltjes transforms (LST's).

For discrete random variables, a probability generating function fully determines the distribution of that variable, say $X$, and equals $\mathbb{E}\left(z^X\right)$. A Laplace-Stieltjes transform is the equivalent of a p.g.f. for a continuous random variable, say $Y$, and equals $\mathbb{E}\left(e^{-\omega Y}\right)$. Probability generating functions and Laplace-Stieltjes transforms are often used in the field of queueing theory to determine distributions of queue lengths or waiting times. The key property which makes the exhaustive and gated disciplines branching type disciplines is the following:

**Property 4.1.** *(Branching-type visit disciplines) If the server arrives to $Q_i$ to find $q_i$ customers there, then during the course of the server's visit, each of these $q_i$ customers will effectively be replaced in an i.i.d. manner by a random population having (say) p.g.f. $h_i(z_1, \dots, z_N)$ which can be any $N$-dimensional p.g.f.*

In the exhaustive policy, the random population that replaces a customer present at queue $i$ at a polling instant is in distribution equal to the number of customers that arrives during a busy period of this queue (this is the time that is needed to go from $k$ customers in $Q_i$ to $k-1$ customers in $Q_i$).

In the gated policy, the server serves only those customers that are present upon arrival of the server. Therefore, the number of customers that replaces each of these customers is in distribution equal to the number of customers that arrives during a service time of a customer in $Q_i$ (this is the time that is needed to go from $k$ customers in front of the gate to $k-1$ customers in front of the gate). So for both the exhaustive and the gated policy, the customers present upon arrival of the server are all *independently* replaced by a random population of new customers during the server's visit.

Following the $k$-limited strategy, the server serves a queue until it is empty or $k$ customers are served. This means that if the number of customers at a queue seen upon arrival is greater than $k$, some of these customers will not be effectively replaced by a random population. Therefore, this policy does not belong to the class of branching type disciplines. Van Vuuren and Winands [87] present an iterative algorithm to approximate the queue length distributions for this policy.

Using a generating function approach, it is possible to determine the limiting distribution of the queue length or the waiting time distribution at polling epochs for the gated and exhaustive discipline (see Resing [71]). From these distributions, one can derive explicit expressions for the generating functions of the stationary queue length distributions. In a similar way, the stationary queue length distributions can be derived for the fixed cycle policy, which is done in Chapter 6.

### Base-stock policies

For the production system, the determination of the queue length or shortfall distribution is very useful for the minimization of holding and backlogging costs. Using a newsvendor type result (see for example Porteus [68]), the optimal base-stock level can be obtained from this distribution. In this way, exhaustive and gated base-stock strategies can be constructed for the production system.

Federgruen and Katalan [37, 38, 39] introduce a combination of these two strategies, where products are produced according to either an exhaustive or a gated base-stock policy. Before each switch-over time, a fixed idle time can be inserted. Because they also consider switching costs, the total costs per time unit can be reduced with these extra idle times. They show in [37] that the total average costs only depend on the total idle time inserted in one cycle and using the results from a previous paper on the corresponding polling model [36], they obtain the optimal total idle time with a numerical procedure. The optimal production sequence is constructed in [39].

Wagner and Smits [77] analyse the production system with an iterative method of De Kok [31] for the following periodic base-stock production strategy. For every product type, a fixed review period is set. At the end of every review period, the demand in that review period is placed in a queue. The machine produces the demand that is placed in the queue in a FIFO manner, in which the arrival moments are the review instants. So the machine produces batches for each product type and then switches to the next product type. The size of each batch has a general distribution, because it is exactly the demand that arrives during one review period. Therefore, the length of each production period is independent of the demand processes at the other product flows. The production sequence follows from the review periods. The lengths of the review periods of the different product types may be different, but it is assumed that the review periods are such that the production sequence is cyclic. In Smits et al. [89], the minimization of set-up and holding costs is considered, under the condition that a certain fill-rate is satisfied. A local search algorithm is presented to obtain the optimal review periods.

In Chapter 6, a fixed cycle strategy is studied for the production system. Just as in Chapter 2, there is no dependency between the lengths of the different visit periods if this strategy is followed for a polling model. Therefore, the analysis for this policy can be done per queue, see Property 1.2. For the same reason, the relative value function based on the fixed cycle strategy can also be found per product type and just as in Chapter 2, an improvement step is performed to obtain a dynamic production strategy in Chapter 7.

## 4.2 Queue lengths and waiting times

While in a production system one is interested in the minimization of holding and backlogging costs, in queueing systems often the average queue length or expected waiting time is minimized. Although the objective functions of these problems are different than the cost function we are looking at, the complexity is similar and results on queue lengths or waiting times in polling systems can be translated into results on shortfall levels or lead times in the production problem.

Further, it is good to mention that the waiting time in the queueing system can be seen as the lead time of a product in the production system, because this is exactly the time between the moment an order (or customer) arrives and the moment that the corresponding product is finished. Therefore, the minimization of the average waiting time in a polling system comes down to the minimization of the average lead time in a production system.

There are several studies on the minimization of the queue lengths. For example, the minimization of the total number of customers in a queueing system is a special case of the problem that Baras et al. [9] look at. They consider a discrete time queueing system with two queues, geometric service times, no switch-over times and a preemptive service discipline. For each waiting customer of queue $i$, costs of $c_i$ are incurred per time unit and the arrival processes are independent of the state of the system. Without any assumptions on the arrival processes, they prove that the minimum expected costs are obtained by following a $c\mu$ rule, with $\mu_i, i = 1, 2$ the service rate of queue $i$. Following this rule, the class of customers with the highest value of $c_i\mu_i$ gets priority. The minimization of the total number of customers can be obtained by choosing $c_i$ equal to 1 for all queues. If, for all queues $c_i$ is the same, the queue with the highest service rate gets priority. It is intuitively easy to understand that this rule minimizes the total number of customers in the system, because there are no switch-over times, so in this way, the server always chooses the fastest way to go from $k$ to $k-1$ customers in the system. If the values of $c_i$ are not all equal, the $c\mu$ rule can be seen as reducing the total costs at maximum speed: The value of $c_i\mu_i$ represents the average costs per time unit that is saved by serving queue $i$, if at least one customer is present. An extension of the work in [9] is given in Baras et al. [8] and in the work of Buyukkoc et al. [26] for systems with an arbitrary number of customer classes. Cox and Smith [28] already proved that this rule is optimal for a system with Poisson arrivals, an arbitrary number of customer classes and a general service time distribution.

For queueing systems with switch-over times between different queues, we can refer to the works of Takagi [79], Hofri and Ross [49], Boxma et al. [16], Liu et al. [59], Boxma et al. [18], Koole [53], Lefeber and Rooda [57]. Hofri and Ross made one of the first attempts to optimize a polling system. They consider a system with two queues and conjecture that the policy that minimizes the sum of discounted switch-over costs and queueing costs is exhaustive service in a nonempty queue and of threshold type for switching from an empty queue to another. Distributional results for this policy are obtained by Boxma et al. in [18]. A similar policy is shown to be optimal for a deterministic queueing system by Lefeber and Rooda [57] who study a fluid system with deterministic production and set-up times. Furthermore, the (continuous) demand process is constant. They look at the minimization of the total queueing costs, where each unit of work in queue $i$ brings costs $c_i$ per time unit. They show that the optimal policy is the following: Both queues are served exhaustively, but the server idles at the most important queue if it is empty and the amount of work at the other queue does not exceed a certain threshold level. Given the optimal policy, the state of the system eventually follows a steady state pattern, because everything is deterministic. Lefeber and Rooda [57] are mainly interested in the optimal path towards this pattern if the current state does not belong to the (bounded) set of optimal steady states.

Koole [53] also looks at a system with two queues, where the arrival processes are Poisson and the service times are exponential with rates depending on the queues. He minimizes the sum of switching and queueing costs and studies the limiting behavior of the switching curve. The optimal policy is compared with the threshold policy to show how complex the optimal policy is. The optimal policy turns out to be a mixture between the threshold policy and the $c\mu$ rule. In this strategy, actions depend on both queue lengths and the machine switches to the other queue if the number of customers at that queue exceeds a certain threshold level. This threshold level depends on the number of customers present at the current queue and may be greater than zero if the current queue is empty, even if the server is set-up for the least important queue.

In a system with two queues, one can thus determine these threshold levels, which only depend on the queue length of the other queue. However, in a system with more than two queues such threshold levels depend on the queue lengths of all other queues. Otherwise, it is not clear to which queue the server should switch if two or more queues exceed their threshold levels. But if the threshold levels depend on the lengths of all other queues, the determination of the optimal threshold levels becomes very complex for systems with a large number of queues.

Liu et al. [59] looked at the structure of the optimal policy for a system with an arbitrary number of queues and only partial knowledge of the state of the system is available. They assume that the service policy is non-preemptive and their objective is the minimization of the amount of work or the number of waiting customers. They decompose this problem into three subproblems. The first one is the determination of the optimal action at a nonempty queue (serve, switch, idle) and they show that in that case, under some fairly general conditions, the server should never idle. The second subproblem is to determine the optimal action (switch or idle) if the server is at an empty queue. They show that under certain conditions either idling is always optimal or switching is always optimal. The third subproblem concerns the optimal routing policy: Which queue is next if one decides to switch? They show that for symmetric polling systems, the optimal routing policy belongs to the class of Stochastically Longest Queue (SLQ) policies. If complete information is available, this means that the server always switches to the queue with the highest amount of work.

The minimization of the average number of waiting customers in a queueing system is equivalent to the minimization of the average waiting time of an arbitrary customer in a queueing system in the sense that both objectives lead to the same optimal policy. In general, Little's law can be applied, which gives a linear relation between the average queue length and the mean waiting time and immediately shows why the two minimizations lead to the same optimal policy. The minimization of the average waiting time is also studied by Boxma et al. in [16]. They derive efficient visit frequencies for polling tables, using an approximation method based on the pseudo conservation law for mean waiting times in polling systems given by Boxma et al. in [15].

However, an arbitrary customer waiting in queue $i$ is not only interested in the visit discipline or frequency of the server (gated, exhaustive, $k$-limited for example) at each queue, but also in the service discipline of the server at queue $i$ which determines the position of the customer within a queue. The limiting *distribution* of the waiting time of an arbitrary customer depends on both the visit and the service discipline, so by choosing the right service discipline one can minimize the variance of the waiting time for a given visit discipline. The following chapter discusses the waiting time distribution for different visit and service disciplines. The analysis in this chapter focusses on the gated and globally gated visit disciplines, and uses an approach with Laplace Stieltjes transforms and generating functions to derive distributional results on the sojourn time distributions. Furthermore, mathematical tools are developed to give some better insight into the effect of service order in polling systems.

# Chapter 5

# Backlog: Waiting times for gated queues in polling systems

In this chapter, the lead time distribution of a product type with a gated base-stock policy is analysed. Because the lead time distribution is exactly the same as the waiting-time distribution in the corresponding polling model, it is more convenient to study this distribution directly from a queueing point of view. The same polling model was already studied by Wierman et al. [91], where expressions for the *mean* waiting times for several service and visit disciplines are derived.

So the production system is translated into a polling system with $N$ queues, which are cyclically visited by a single server. For every set of base-stock levels, the corresponding polling system behaves exactly the same. Translating the production system to a polling system, each product flow becomes a queue, but one can still look at this queue as if it is a product flow by setting the base-stock level to zero. Then, each outstanding order is called a customer. Each of these customers requires a service time, which is equal to a production time. The arrival process of each queue is exactly the same as the demand process of the corresponding product flow. The service times are still generally distributed, but the arrival processes are now independent Poisson processes, whereas in the previous chapters we also consider compound Poisson processes. The reason for this assumption is just analytical: The assumption on Poisson arrivals makes the analysis somewhat more elegant, so that the focus of this study lies on the derivation of the waiting-time distribution in general. However, it is also possible to extend the analysis presented here to also find the waiting-time distribution for compound Poisson arrival processes.

When the server visits queue $i$, $i = 1, \dots, N$, it serves a number of customers according to a certain visit discipline. This discipline is assumed to belong to the class of branching-type disciplines, which includes gated and exhaustive service. Although in a production system the order of service at a certain queue or product flow is often First Come First Served (FCFS), one may also consider a Last Come First Served (LCFS) service policy or Random Order of Service (ROS), where all customers (or production orders) present at the current queue (product flow) are equally likely to get served next. Because the service times are stochastic, the order of service can also depend on the service requirements of the present customers. One may for example first serve the customers with the shortest service time. This policy is called a Shortest Job First (SJF) service policy.

In production systems, the machine can only produce one product at a time. In the polling system, this is equivalent to a server that can only serve one customer at a time. However, there also exist polling systems in which a server is able to serve multiple customers at the same time. Examples of such polling systems are the 802.11 (see Lam et al. [55]) and Bluetooth (see Miorandi et al. [63]) protocols, and scheduling policies at routers and I/O systems in web servers. In such applications, often featuring high service time variability, it may be advantageous to give non-FCFS service. For these systems, one can also think of a Processor Sharing (PS) service policy, in which all jobs are served simultaneously.

The special feature of our study is that, within each queue, we do not restrict ourselves to service in order of arrival (FCFS); we are interested in the effect of different service disciplines. After a discussion of the joint distribution of the numbers of customers at each queue at visit epochs of the server to a particular queue, we determine the Laplace-Stieltjes transform of the cycle-time distribution, viz., the time between two successive visits of the server to a certain queue, say queue 1. This yields the transform of the joint distribution of past and residual cycle time, w.r.t. the arrival of a tagged customer at the first queue. Subsequently concentrating on the case of gated service at the first queue, we use that cycle-time result to determine the (Laplace-Stieltjes transform of the) waiting-time distribution at queue 1. Next to locally gated visit disciplines, we also consider the globally gated discipline. Again, we consider various non-FCFS service disciplines at the queues, and we determine the (Laplace- Stieltjes transform of the) waiting-time distribution at an arbitrary queue. This chapter is based on Boxma et al. [19].

## 5.1 Introduction

We consider a polling system of $N$ queues $Q_1, \ldots, Q_N$, cyclically visited by a single server. Customers arrive at these queues according to independent Poisson processes, requiring generally distributed service times. In polling system design several decisions need to be made, for instance one needs to decide on (i) the order of service of the queues, (ii) the visit disciplines, and (iii) the service disciplines. Regarding (i), a fixed cyclic order is usually assumed, but random polling orders and polling tables have also been studied. With regard to (ii), many polling disciplines have been considered. Well-known polling disciplines are the exhaustive discipline, the gated discipline, and the $k$-limited discipline, as described in Chapter 4. Hardly any attention has been given to (iii). It is almost invariably assumed that the order of service within each queue is FCFS (First Come First Served). However, in Wierman et al. [91] several other service disciplines are considered, like PS (Processor Sharing), ROS (Random Order of Service), LCFS (Last Come First

Served), SJF (Shortest Job First), and fixed priorities. Using the recently developed MVA (*Mean Value Analysis*) approach for polling systems of Winands et al. [94], the mean sojourn times at the various queues are obtained, for the case of cyclic polling and either the exhaustive or the gated polling discipline at each queue. It is demonstrated in Wierman et al. [91] that one can quite easily determine the mean sojourn times in this case, and that the effect of the service order may be rather profound, in particular in the case of exhaustive service.

The present chapter builds upon [91]. Our goal is to determine the LST (Laplace-Stieltjes transform) of the sojourn-time *distributions* at the various queues of a cyclic polling system, for several service disciplines. This allows us to study the effect of different service disciplines on the sojourn time. We assume the polling disciplines at the various queues belong to the class of branching-type disciplines (see Resing [71]), which includes gated and exhaustive service but which does not include, e.g., 1-limited service. However, we restrict the determination of the sojourn-time distribution at some queue to the case that the polling discipline at that particular queue is gated. In a future study we intend to tackle the more difficult problem of deriving the sojourn-time distribution at a queue with exhaustive service.

Next to locally gated polling disciplines, we also consider the globally gated discipline, which operates as follows: When the server arrives at $Q_1$, a gate is closed for *all* queues simultaneously. In the next cycle, the server serves exactly those customers who are located before the gate, i.e., those who were already present when the server arrived at $Q_1$. Again, we consider various non-FCFS service disciplines at the queues, and we determine the LST (Laplace-Stieltjes transform) of the sojourn-time distribution at an arbitrary queue.

Our approach is as follows. In the case of a branching-type polling discipline at all the queues, Resing [71] has obtained the joint distribution of the number of customers at each queue at visit epochs of the server to a particular queue. His result is easily seen to remain valid when the service order at a queue is not FCFS. Using this queue-length result, we determine the LST of the cycle-time distribution, viz., the time between two successive visits of the server to, say, $Q_1$. This yields the transform of the joint distribution of past and residual cycle time, w.r.t. the arrival of a tagged customer at $Q_1$. Finally, we use that cycle-time result to determine the (LST of the) sojourn-time distribution at $Q_1$. Manipulation of this transform gives sojourn-time moments, generalizing the mean sojourn-time results recently obtained via Mean Value Analysis in Winands et al. [91].

This chapter is organized as follows. Section 2 contains a model description. In Section 3 we study the cycle time in the cyclic polling system with a branching-type polling discipline at each queue. These results are then used in Section 4, which contains an analysis of the sojourn time distribution in a gated queue, for various service orders like FCFS, LCFS, PS, ROS and SJF. We then show, in Section 5, how our ideas can be applied to polling systems that are served in a globally gated fashion. Finally, Section 6 contains some concluding remarks and mentions topics for further research.

## 5.2 Model Description

A single server visits $N$ queues $Q_1, \dots, Q_N$ in cyclic order. Customers arrive at these queues according to independent Poisson processes $\{N_i(t),\ t \in \mathbb{R}\}$ with arrival rate $\lambda_i$ at $Q_i$, $i = 1, \dots, N$. The service requirements of customers at $Q_i$, to be called type-$i$ customers, are i.i.d. (independent, identically distributed) random variables, with distribution $B_i(\cdot)$ and LST $\beta_i(\cdot)$, $i = 1, \dots, N$; $B_i$ will denote a generic service time at $Q_i$. Since we will be interested in deriving the sojourn-time distributions of customers that arrive to the system during steady-state, it is notationally more convenient to define stationary versions of our processes on the entire real line $\mathbb{R}$. Hence, each arrival process $N_i$ consists of points $\{T_{i,n}\}_{n \in \mathbb{Z}}$, where $\mathbb{Z}$ denotes the set of integers, and $T_0 \leq 0 < T_1$. Associated with each point is its service time $B_{i,n}$; thus, the points $(T_{i,n}, B_{i,n})$ define a marked Poisson process on $\mathbb{R}^2$. The switch-over times of the server from $Q_i$ to $Q_{i+1}$ ($Q_{N+1}$ denoting $Q_1$) have distribution $S_i(\cdot)$ and LST $\sigma_i(\cdot)$, $i = 1, \dots, N$; $S_i$ will denote a generic switch-over time from $Q_i$. The server even switches among queues when all queues are empty. All interarrival times, service times and switch-over times are assumed to be independent.

When the server visits $Q_i$, it serves a number of customers according to a certain *polling discipline.* We first concentrate on polling disciplines that belong to the class of branching-type disciplines, as introduced in Resing [71]. This class is characterized by the fact that each queue satisfies Property 4.1. Important examples of branching-type disciplines are *Exhaustive service* (the server visits a queue until it has emptied the queue) and *Gated service* (during a visit to a queue, the server serves exactly those customers who were present at the beginning of that visit). 1-limited service (the server serves just one customer during a visit, if there is at least one customer present at the beginning of the visit) does *not* belong to the class of branching-type disciplines. Borst [13] gives a slight extension of Property 4.1 that is also held by a globally gated polling system:

**Property 5.1.** *If there are $k_i$ customers present in $Q_i$ at the beginning of a visit to $Q_{\pi(i)}$ with $\pi(i) \in \{1, \dots, N\}$, then during the course of the visit to $Q_i$, each of these $k_i$ customers will effectively be replaced in an i.i.d. manner by a random population having probability generating function $h_i(z_1, z_2, \dots, z_N)$, which may be any $N$-dimensional probability generating function.*

When we begin to discuss globally gated polling disciplines, as introduced in Boxma et al. [17], it will be clear that Property 5.1 is satisfied. Under this discipline the server, in a cycle starting at $Q_1$, only serves the customers that are present at a polling instant at $Q_1$.

Resing [71] has shown that, if Property 4.1 holds at each queue, the joint queue-length process at polling instants of a fixed queue is a so-called multi-type branching process (MTBP) with immigration. The theory of MTBP (see Athreya and Ney [5] or Resing [71]) now leads to an expression for the generating function of the joint queue length process at polling instants.

For a given polling discipline, we still have to specify the *service discipline* during the visit to a queue. As was already mentioned, we are interested in the effect of different service disciplines on the sojourn times of customers.

Define $\rho_i := \lambda_i \mathbb{E} B_i$ the traffic intensity at $Q_i$, and denote by $\rho := \sum_{i=1}^N \rho_i$ the total traffic intensity. We restrict ourselves to the case $\rho < 1$. For the class of polling systems discussed here, this condition guarantees that the vectors of queue lengths at polling epochs and at arbitrary epochs have steady-state distributions.

## 5.3 The Cycle Time in the Branching-Type Polling Model

In this section we determine the LST of the cycle time $C$ for $Q_1$, i.e., the time between two successive visits of the server to $Q_1$. Notice that $C$ now denotes the length of a time interval instead of the number of slots in one cycle, as was done in the previous chapters on the production system with lost sales. The reason for this is that in this chapter, a polling system is modeled in *continuous* time and $C$ is the most natural choice to denote the cycle length.

In Theorem 5.1 we compute the LST of the conditional cycle time, given the numbers of customers present at all buffers in the polling system at the beginning of the cycle. By unconditioning, the cycle time transform is obtained (Corollary 5.1). But first we present some results from Resing [71], which will be used in the sequel.

In Section 5.2 we mentioned the class of branching-type polling disciplines (Resing [71]); see Property 4.1. We assume that each queue in our polling system satisfies this property, with generating function $h_i(z_1, \ldots, z_N)$ at $Q_i$, $i = 1, \ldots, N$. For gated service at $Q_i$,

$$h_i(z_1, \ldots, z_N) = \beta_i(\sum_{j=1}^N \lambda_j (1 - z_j)). \tag{5.1}$$

For exhaustive service at $Q_i$, with $\pi_i(\cdot)$ denoting the LST of the busy period of $M/G/1$ queue $Q_i$ in isolation, i.e., an $M/G/1$ queue with arrival rate $\lambda_i$ and service time distribution $B_i(\cdot)$:

$$h_i(z_1, \ldots, z_N) = \pi_i(\sum_{j \neq i} \lambda_j (1 - z_j)). \tag{5.2}$$

Resing [71] has proven the following. Let $P(z_1, \ldots, z_N)$ denote the GF of the steady-state joint distribution of the numbers of customers $X(1), \ldots, X(N)$ in $Q_1, \ldots, Q_N$ at an arbitrary visit beginning of the server at $Q_1$. Then

$$P(z_1, \ldots, z_N) = \prod_{n=0}^{\infty} g(f_n(z_1, \ldots, z_N)). \tag{5.3}$$

The functions $f_n(z_1, \ldots, z_N)$ are defined inductively by

$$
\begin{aligned}
& f_0(z_1, \ldots, z_N) = (z_1, \ldots, z_N), \\
& f_n(z_1, \ldots, z_N) = \\
& \quad (f^{(1)}(f_{n-1}(z_1, \ldots, z_N)), \ldots, f^{(N)}(f_{n-1}(z_1, \ldots, z_N))),
\end{aligned} \tag{5.4}
$$

where the *off-spring* GFs $f^{(i)}(z_1, \ldots, z_N)$, $i = 1, \ldots, N$, are given by

$$
\begin{aligned}
& f^{(i)}(z_1, \ldots, z_N) = \\
& \quad h_i(z_1, \ldots, z_i, f^{(i+1)}(z_1, \ldots, z_N), \ldots, f^{(N)}(z_1, \ldots, z_N)).
\end{aligned} \tag{5.5}
$$

The *immigration* GF $g(z_1, \ldots, z_N)$ is given by

$$
\begin{aligned}
& g(z_1, \ldots, z_N) = \\
& \prod_{i=1}^{N} \sigma_i(\sum_{k=1}^{i} \lambda_k(1 - z_k) + \sum_{k=i+1}^{N} \lambda_k(1 - f^{(k)}(z_1, \ldots, z_N))).
\end{aligned} \tag{5.6}
$$

Let us now turn to the cycle time. Denoting the visit time (time spent in a queue by the server) of $Q_i$ by $V_i$, $i = 1, \ldots, N$, we have

$$
C = \sum_{k=1}^{N} (V_k + S_k). \tag{5.7}
$$

Let $\theta_i(\omega)$ represent the LST of the time that the server spends at $Q_i$ due to the presence of one customer there. In the case of gated service, $\theta_i(\omega) = \beta_i(\omega)$, the service time LST; in the case of exhaustive service, $\theta_i(\omega) = \pi_i(\omega)$, the busy-period LST. We also need to introduce the following functions: $\psi_i(\omega) = \omega + \lambda_i(1 - \theta_i(\omega))$, $i = 1, \ldots, N$, and $\psi_{i,N}(\omega) = \psi_{i+1}(\psi_{i+2}(...(\psi_N(\omega))))$, $i = 1, \ldots, N$; here $\psi_{N,N}(\omega) = \omega$.

**Theorem 5.1.** *The LST of the cycle time $C$, conditional on the numbers of customers in all queues at the beginning of the cycle, is given by:*

$$
\mathbb{E}(\mathrm{e}^{-\omega C} | X(i) = m_i, 1 \leq i \leq N) = \prod_{i=1}^{N} \sigma_i(\psi_{i,N}(\omega)) \theta_i^{m_i}(\psi_{i,N}(\omega)). \tag{5.8}
$$

**Proof.**
In the formulas below, the condition “$m_1, \ldots, m_k$” denotes $X(1) = m_1, \ldots, X(k) = m_k$.

$$
\begin{aligned}
& \mathbb{E}(\mathrm{e}^{-\omega C} | m_1, \ldots, m_N) = \mathbb{E}(\mathrm{e}^{-\omega \sum_{j=1}^{N} (V_j + S_j)} | m_1, \ldots, m_N) \qquad (5.9) \\
& \quad = \ \sigma_N(\omega) \theta_N^{m_N}(\omega) \mathbb{E}\left( \mathrm{e}^{-(\omega + \lambda_N(1 - \theta_N(\omega))) \sum_{j=1}^{N-1} (V_j + S_j)} | m_1, \ldots, m_{N-1} \right) \\
& \quad = \ \sigma_N(\omega) \theta_N^{m_N}(\omega) \sigma_{N-1}(\psi_N(\omega)) \theta_{N-1}^{m_{N-1}}(\psi_N(\omega)) \\
& \quad \times \ \mathbb{E}\left( \mathrm{e}^{-\psi_{N-1}(\psi_N(\omega)) \sum_{j=1}^{N-2} (V_j + S_j)} | m_1, \ldots, m_{N-2} \right).
\end{aligned}
$$

Repeating the above iteration procedure finally yields the statement of the theorem.

Deconditioning immediately gives the cycle time LST for $Q_1$:

**Corollary 5.1.**

$$\mathbb{E}\left(\mathrm{e}^{-\omega C}\right) = \prod_{i=1}^{N} \sigma_i(\psi_{i,N}(\omega)) \\ \times\ P(\theta_1(\psi_{1,N}(\omega)), \ldots, \theta_i(\psi_{i,N}(\omega)), \ldots, \theta_N(\psi_{N,N}(\omega))). \tag{5.10}$$

A similar type of expression can be given for polling systems that satisfy Property 5.1. Notice that the state of the polling system at the embedded instants when the server begins its visit at $Q_1$ forms a MTBP, either with or without immigration, depending on the setup times. Unlike Resing's property, however, this is not true at any of the other visit epochs in a globally gated polling discipline. Even so, in the globally gated case one can still easily compute the steady-state distribution at any other epoch by knowing the steady-state distribution at $Q_1$.

Our derivation of the LST of $C$ shows that the distribution of $C$ is only dependent on the polling discipline at each queue, and not on the scheduling discipline used within each queue. Indeed, we know from Resing [71] that the steady-state generating function $P$ given in Corollary 5.1 only depends on the polling disciplines, and it is clear that the same is true for the conditional transform found in Theorem 5.1.

We should also point out that, if the service discipline at $Q_1$ is gated, the LST of $C$ has an even simpler form. In particular, while the system is in steady-state, the number of customers $X(1)$ found at $Q_1$ at the moment the server switches to $Q_1$ is equal in distribution to a randomized Poisson random variable (due to $C$ being random) with parameter $\lambda_1 C$. Hence, we have the following corollary.

**Corollary 5.2.** *If the service discipline at $Q_1$ is gated, then for $0 < z < 1$, the generating function of $X(1)$ is as follows:*

$$\mathbb{E}\left(z^{X(1)}\right) = \mathbb{E}\left(e^{-\lambda_1(1-z)C}\right). \tag{5.11}$$

This allows us to relate all of the moments of $C$ to the factorial moments of $X(1)$ (at the beginning of a visit period) in the following way: for each integer $n \geq 1$, $\mathbb{E}\left((X(1))_n\right) = \lambda_1^n \mathbb{E}\left(C^n\right)$, where for $x \in \mathbb{R}$, $(x)_n = x(x-1)\cdots(x-n+1)$.

Moreover, there do exist efficient algorithms that are designed to compute the factorial moments of $X(1)$, at the beginning of a visit period. Resing also discusses a possible method of computing these moments in Section 6 of [71], which involve successively taking derivatives of the following equation (equation (11) in [71]):

$$P(z_1, \ldots, z_n) = g(z_1, \ldots, z_n) P(f_1(z_1, \ldots, z_n)).$$

Notice that this can be used to see why equation (5.3) holds above.

### 5.3.1 The Biased Cycle Length

Throughout this chapter, we will be interested in the distribution of various components of the steady-state cycle time, given that a particular type of customer arrived during such a cycle. Knowing that such an arrival occurred will bias the length of the cycle, and this must be accounted for.

Our cycles will always begin at the moment the server begins to work on jobs present at $Q_1$, due to the fact that the polling discipline at $Q_1$ will always be of a "gated" nature; either $Q_1$ will behave in a gated fashion, or the system will be globally gated with $Q_1$ being the queue that governs the opening and closing of all gates in the system. Assuming that a customer arrives to $Q_1$ during a cycle, let $C^*$, $C^p$ and $C^r$ denote the total biased cycle length, the amount of time between the beginning of the cycle and the arrival of the tagged customer to $Q_1$, and the amount of time between the arrival of such a tagged customer and the end of the cycle, respectively. Clearly $C^* = C^p + C^r$, and the tagged customer will not be served until the next cycle begins. $C^*$ will also be referred to throughout parts of the chapter as the cycle time of the tagged customer. When we look at the globally gated case, we will assume that all gates in the system are synchronized with the gate at $Q_1$, and so this same choice of cycles will be appropriate when we are interested in the sojourn time distribution of customers that arrive at $Q_i$, for $1 \leq i \leq N$.

Our goal is to now relate their distributions to the distribution of $C$, which is the steady-state unbiased cycle-length. It is known that, conditional on $C^*$, the distribution of $C^p$ is uniform on $[0, C^*]$. Furthermore, it is also known in the literature (see, for example, Thörisson [81]) that

$$\mathrm{d}P(C^* \leq x) = \frac{x\mathrm{d}P(C \leq x)}{\mathbb{E}(C)}. \tag{5.12}$$

From this result, it is then immediately clear that

$$\mathbb{E}(C^*) = \frac{\mathbb{E}(C^2)}{\mathbb{E}(C)},$$

$$\mathbb{E}(C^p) = \mathbb{E}(C^r) = \frac{\mathbb{E}(C^2)}{2\mathbb{E}(C)}.$$

Moreover, we can use (5.12) to compute the joint LST of $C^p$ and $C^r$:

$$\int_{t=0}^{\infty}\int_{u=0}^{\infty} e^{-at}e^{-bu}\mathrm{d}P(C^p \leq t; C^r \leq u) \ = \ \frac{\mathbb{E}[e^{-aC}] - \mathbb{E}[e^{-bC}]}{(b-a)\mathbb{E}(C)}. \tag{5.13}$$

This joint LST can be used to show that

$$\mathbb{E}(C^pC^r) \ = \ \frac{\mathbb{E}((C^r)^2)}{2} \tag{5.14}$$

which gives some insight into the correlation between $C^p$ and $C^r$.

The derivation of these last results is known, and also beyond the scope of the chapter so a discussion of their derivation has been omitted. To appease the interested reader, we will mention that these results can be derived through the use of Palm theory, which can be used to capture the biases that are mentioned above. The Palm framework allows us to work with the fact that, under the Palm measure induced by the point process consisting of the times at which a cycle begins, the sequence of cycle lengths formed in the stationary version of this polling system forms a stationary sequence, but does not form an i.i.d. sequence. If this were true, we could instead have made use of well-known results from renewal theory: for instance, the reader may recognize that $\mathbb{E}\left(C^r\right)$ has the same form as the first moment of the stationary residual lifetime from a renewal process. References on Palm theory are numerous: examples of more recent references include Baccelli and Brémaud [7] and Serfozo [75] (both focus on applications in queueing), along with Thörisson [81].

Throughout our analysis, we will also make use of what is known in the literature as the stationary-excess operator $R$ (see, for instance, Abate and Whitt [1]), which is defined in the following way: for a given nonnegative random variable $X$,

$$P(R_X \leq t) = \frac{1}{\mathbb{E}(X)} \int_0^t P(X > s)\mathrm{d}s, \qquad t \geq 0.$$

We will also be applying this operator multiple times to a given random variable, and to denote this we will use the abbreviation $R_{X,n}$, where $R_{X,0} = X$, $R_{X,1} = R_X$, and for any $n \geq 0$, $R_{X,n+1} = R_{R_{X,n}}$.

The reader should note that for cycle times, $R_C$ and $C^r$ will both be used throughout various parts of the chapter, even though they both have the same distribution. The former will typically be used within computations, while the latter will exclusively be used to represent a particular residual cycle time observed by a tagged customer.

Now we are ready to state the following lemma, which will prove to be useful while computing the first and second moments of many of the types of sojourn times considered in this chapter.

**Lemma 5.1.** *For $a, b \geq 0$, $a \neq b$,*

$$\frac{\mathbb{E}(e^{-aC}) - \mathbb{E}(e^{-bC})}{(b-a)\mathbb{E}(C)} = (-a)^{n+1}\frac{\mathbb{E}(R_C^{n+1})}{(n+1)!}\left[\frac{\mathbb{E}(e^{-aR_{C,n+1}}) - \mathbb{E}\left(e^{-bR_{C,n+1}}\right)}{(b-a)\mathbb{E}(R_{C,n+1})}\right]$$
$$+ \sum_{k=0}^{n}(-a)^k\frac{\mathbb{E}\left(R_C^k\right)}{k!}\mathbb{E}\left(e^{-bR_{C,k+1}}\right). \tag{5.15}$$

*Proof.* The LST of $R_C$ is known, and can be found in, for instance, Abate and Whitt [1]:

$$\mathbb{E}(e^{-\omega R_C}) = \frac{1 - \mathbb{E}(e^{-\omega C})}{\omega\mathbb{E}(C)}. \tag{5.16}$$

Equation (5.15), for $n = 0$, then follows from (5.13) and (5.16):

$$\begin{aligned}
\frac{\mathbb{E}(e^{-aC}) - \mathbb{E}(e^{-bC})}{(b-a)\mathbb{E}(C)} &= \frac{1 - \mathbb{E}(e^{-bC})}{(b-a)\mathbb{E}(C)} - \frac{1 - \mathbb{E}(e^{-aC})}{(b-a)\mathbb{E}(C)} \\
&= \frac{b}{b-a}\mathbb{E}(e^{-bR_C}) - \frac{a}{b-a}\mathbb{E}(e^{-aR_C}) \\
&= \mathbb{E}(e^{-bR_C}) - a\mathbb{E}(R_C)\left[\frac{\mathbb{E}(e^{-aR_C}) - \mathbb{E}(e^{-bR_C})}{(b-a)\mathbb{E}(R_C)}\right] \qquad (5.17)
\end{aligned}$$

At this point we begin to see a pattern: if we apply (5.16), but with $C$ and $R_C$ being replaced with $R_C$ and $R_{C,2}$, respectively, to the fraction found in (5.17), and then repeat accordingly, we see that for any $n \geq 1$,

$$\begin{aligned}
\frac{\mathbb{E}\left(e^{-aC}\right) - \mathbb{E}\left(e^{-bC}\right)}{(b-a)\mathbb{E}(C)} &= \sum_{k=0}^{n}(-a)^k\left[\prod_{j=1}^{k}\mathbb{E}\left(R_{C,j}\right)\right]\mathbb{E}\left(e^{-bR_{C,k+1}}\right) \\
&+ (-a)^{n+1}\left[\prod_{j=1}^{n+1}\mathbb{E}\left(R_{C,j}\right)\right]\left[\frac{\mathbb{E}\left(e^{-aR_{C,n+1}}\right) - \mathbb{E}\left(e^{-bR_{C,n+1}}\right)}{(b-a)\mathbb{E}\left(R_{C,n+1}\right)}\right], \qquad (5.18)
\end{aligned}$$

where products of the form $\prod_{j=1}^{0}$ will be understood to equal 1.

The proof will be complete once we compute each of the products found in (5.18); such product computations have been observed before (see Whitt [90] and the references given there, for instance), but for the reader's convenience we will also provide a proof. Notice that

$$\begin{aligned}
&\mathbb{E}\left(e^{-\omega C}\right) = 1 - \omega\mathbb{E}(C)\mathbb{E}\left(e^{-\omega R_C}\right) \\
&= 1 - \omega\mathbb{E}(C) + \omega^2\mathbb{E}(C)\mathbb{E}\left(R_C\right)\mathbb{E}\left(e^{-\omega R_{C,2}}\right) = \dots \\
&= \sum_{k=0}^{n}(-\omega)^k\prod_{m=0}^{k-1}\mathbb{E}\left(R_{C,m}\right) + (-\omega)^{n+1}\left[\prod_{m=0}^{n}\mathbb{E}\left(R_{C,m}\right)\right]\mathbb{E}\left(e^{-\omega R_{C,n+1}}\right)
\end{aligned}$$

for each $n \geq 1$. Therefore

$$\prod_{m=0}^{k-1}\mathbb{E}(R_{C,m}) = \frac{\mathbb{E}\left(C^k\right)}{k!}, \qquad (5.19)$$

so for each $k \geq 1$, we conclude from both (5.19) and (5.12) that

$$\prod_{m=1}^{k}\mathbb{E}(R_{C,m}) = \frac{\mathbb{E}\left(C^{k+1}\right)}{(k+1)!\mathbb{E}(C)} = \frac{\mathbb{E}\left(R_C^k\right)}{k!}.$$

This proves (5.15). □

We would also like to point out to the reader that, in the work of Winands et al. [94], Mean Value Analysis was used to derive a system of equations which, once solved, allow for the computation of both the expected number of customers at a given buffer in steady-state, along with the expected waiting time of a customer at a given buffer. Moreover, the solution to these equations also allows us to numerically compute the first moment of $R_C$.

## 5.4 Sojourn times at a gated queue

In this section we will be interested in the sojourn time distribution of a tagged customer that visits a gated queue, at a time when the system is in steady-state. It should be emphasized here that the polling disciplines used at all other buffers in the system can be of a branching-type; we do not need to assume that all of them also operate under the gated discipline.

### 5.4.1 First Come First Served

We begin by computing the LST of the sojourn time of a tagged customer that visits a queue, whose customers are served in accordance to a FCFS scheduling policy. It is clear that

$$T_{FCFS} = C^r + B_1 + \sum_{T_{1,k} \in (-C_p, 0)} B_{1,k}.$$

Here $T_{FCFS}$ represents the sojourn time of a tagged customer (bringing an amount of work $B_1$ to the system) that arrives to $Q_1$ while the system is in equilibrium. The reader should also notice that we are assuming that the tagged customer arrives at time zero, which is a quite standard assumption in the queueing literature: indeed, it is often implicitly used in many studies found in the literature on queues. Notice that we have suppressed the fact that we are referring to $Q_1$ in our notation for $T_{FCFS}$, and we will continue to do so throughout the rest of this section. The reason why we will follow this practice is because, for gated systems, the gate at $Q_1$ only moves at the moment the server begins working there. This allows us to conclude that the waiting-time distribution has the same form for all other $Q_i$ that operate under a gated scheme; the only difference would involve considering cycles that begin at the moment the server begins working at $Q_i$ instead of $Q_1$.

After conditioning on the past and residual cycle lengths, we see that

$$
\begin{aligned}
\mathbb{E}(e^{-\omega T_{FCFS}}) &= \mathbb{E}\left(e^{-\omega\left(C^r+B_1+\sum_{T_{1,k}\in(-C_r,0)} B_{1,k}\right)}\right) \\
&= \int_0^\infty \int_0^\infty e^{-\omega u} \sum_{n=0}^\infty \beta_1(\omega)^{n+1} \frac{(\lambda_1 t)^n e^{-\lambda_1 t}}{n!} \mathrm{d}P\left(C^p \leq t, C^r \leq u\right) \\
&= \beta_1(\omega) \int_0^\infty \int_0^\infty e^{-\omega u} e^{-\lambda_1(1-\beta_1(\omega))t} \mathrm{d}P\left(C^p \leq t, C^r \leq u\right) \\
&= \beta_1(\omega) \left[\frac{\mathbb{E}\left(e^{-\lambda_1(1-\beta_1(\omega))C}\right) - \mathbb{E}\left(e^{-\omega C}\right)}{\mathbb{E}(C)\left(\omega - \lambda_1(1-\beta_1(\omega))\right)}\right] \qquad (5.20) \\
&= \beta_1(\omega)\mathbb{E}\left(e^{-\omega D_{FCFS}}\right),
\end{aligned}
$$

where $D_{FCFS}$ denotes the delay of the tagged customer. Throughout this chapter, for an arbitrary scheduling discipline $\Gamma$ we will typically let $D_\Gamma$ denote the steady-state sojourn time of a tagged customer *minus* its service time.

The first moment of $T_{FCFS}$ is well-known, and can be found in many places throughout the polling literature (see, for instance, Boxma [14] or Takagi [79]):

$$
\mathbb{E}(T_{FCFS}) = \mathbb{E}(B_1) + \mathbb{E}(C^r)(1+\rho_1).
$$

We will now show how to efficiently use (5.20) to compute both the first and second moment of $T_{FCFS}$. By applying Lemma 5.1 to (5.20), we see that for each $n \geq 1$, when $\omega \downarrow 0$,

$$
\begin{aligned}
&\mathbb{E}(e^{-\omega D_{FCFS}}) = \\
&\quad \sum_{k=0}^n (-1)^k (\lambda_1(1-\beta_1(\omega)))^k \frac{\mathbb{E}\left(R_C^k\right)}{k!} \mathbb{E}\left(e^{-\omega R_{C,k+1}}\right) + \mathcal{O}\left(\omega^{n+1}\right).
\end{aligned}
$$

Due to the fact that

$$
\lambda_1(1-\beta_1(\omega)) = \rho_1\omega - \lambda_1 \frac{\mathbb{E}\left(B_1^2\right)}{2}\omega^2 + \mathcal{O}(\omega^3), \qquad \omega \downarrow 0
$$

we find that the LST of $D_{FCFS}$ can also be expressed in the following way: as $\omega \downarrow 0$,

$$
\begin{aligned}
\mathbb{E}\left(e^{-\omega D_{FCFS}}\right) &= \mathbb{E}(e^{-\omega R_C}) - \lambda_1(1-\beta_1(\omega))\mathbb{E}\left(R_C\right)\mathbb{E}\left(e^{-\omega R_{C,2}}\right) \\
&+ (\lambda_1(1-\beta_1(\omega)))^2 \frac{\mathbb{E}\left(R_C^2\right)}{2}\mathbb{E}\left(e^{-\omega R_{C,3}}\right) + \mathcal{O}\left(\omega^3\right) \\
&= 1 - \mathbb{E}\left(R_C\right)(1+\rho_1)\omega + \lambda_1 \frac{\mathbb{E}\left(B_1^2\right)}{2}\mathbb{E}\left(R_C\right)\omega^2 \\
&+ \frac{\mathbb{E}\left(R_C^2\right)}{2}\left[1+\rho_1+\rho_1^2\right]\omega^2 + \mathcal{O}\left(\omega^3\right).
\end{aligned}
$$

Thus,

$$\mathbb{E}\left(D_{FCFS}^2\right) = \lambda_1 \mathbb{E}\left(B_1^2\right)\mathbb{E}\left(R_C\right) + \mathbb{E}\left(R_C^2\right)\left(1+\rho_1+\rho_1^2\right),$$

which also implies that

$$\begin{aligned}\mathbb{E}\left(T_{FCFS}^2\right) &= \mathbb{E}\left(B_1^2\right) + \mathbb{E}\left(R_C\right)\left(2(1+\rho_1)\mathbb{E}(B_1) + \lambda_1\mathbb{E}\left(B_1^2\right)\right) \\ &+ \mathbb{E}\left(R_C^2\right)\left[1+\rho_1+\rho_1^2\right].\end{aligned}$$

### 5.4.2 Last Come First Served

The LST of the sojourn time $T_{LCFS}$ of a tagged customer under the Last Come First Served (LCFS) discipline has a form that is similar to the LST of $T_{FCFS}$. Under LCFS, all of the workload that arrives to $Q_1$ after the tagged customer, yet during the cycle time of the tagged customer, will be processed before him, and so

$$T_{LCFS} = C^r + B_1 + \sum_{T_{1,k}\in(0,C^r)} B_{1,k}.$$

By performing a sequence of calculations that is similar to what was done in the FCFS case, we see that the LST of $T_{LCFS}$ is just

$$\begin{aligned}\mathbb{E}\left(e^{-\omega T_{LCFS}}\right) &= \mathbb{E}\left(e^{-\omega(C^r+B_1+\sum_{T_{1,k}\in(0,C^r)} B_{1,k})}\right) \\ &= \int_0^\infty e^{-\omega t}\sum_{n=0}^{\infty}\beta_1(\omega)^{n+1}\frac{(\lambda_1 t)^n e^{-\lambda_1 t}}{n!}\mathrm{d}P\left(C^r\le t\right) \\ &= \beta_1(\omega)\mathbb{E}\left(e^{-(\omega+\lambda_1(1-\beta_1(\omega)))C^r}\right) \\ &= \beta_1(\omega)\left[\frac{1-\mathbb{E}\left(e^{-(\omega+\lambda_1(1-\beta_1(\omega)))C}\right)}{\mathbb{E}(C)\left(\omega+\lambda_1(1-\beta_1(\omega))\right)}\right] \\ &= \beta_1(\omega)\mathbb{E}\left(e^{-\omega D_{LCFS}}\right). \qquad (5.21)\end{aligned}$$

Given the form of (5.21), we see that Lemma 5.1 is not useful here, since the delay transform can be explicitly stated in terms of a single $C^r$ transform. Clearly,

$$\omega+\lambda_1(1-\beta_1(\omega)) = (1+\rho_1)\omega - \lambda_1\frac{\mathbb{E}\left(B_1^2\right)\omega^2}{2} + \mathcal{O}\left(\omega^3\right), \qquad \omega\downarrow 0$$

and this simple fact will allow us to rewrite the LST of $D_{LCFS}$ in the following way:

$$\begin{aligned}
\mathbb{E}\left(e^{-\omega D_{LCFS}}\right) &= \sum_{n=1}^{\infty}(-1)^{n-1}\left((1+\rho_1)\omega-\lambda_1\frac{\mathbb{E}\left(B_1^2\right)}{2}\omega^2\right)^{n-1}\frac{\mathbb{E}\left(C^n\right)}{n!\mathbb{E}(C)}\\
&= 1-\left((1+\rho_1)\omega-\frac{\lambda_1\mathbb{E}\left(B_1^2\right)\omega^2}{2}\right)\mathbb{E}\left(R_C\right)\\
&+ \left((1+\rho_1)\omega-\frac{\lambda_1\mathbb{E}\left(B_1^2\right)\omega^2}{2}\right)^2\frac{\mathbb{E}\left(R_C^2\right)}{2}+\mathcal{O}\left(\omega^3\right)\\
&= 1-(1+\rho_1)\mathbb{E}\left(R_C\right)\omega\\
&+ \left[\lambda_1\frac{\mathbb{E}\left(B_1^2\right)\mathbb{E}\left(R_C\right)}{2}+(1+\rho_1)^2\frac{\mathbb{E}\left(R_C^2\right)}{2}\right]\omega^2+\mathcal{O}(\omega^3), \qquad \omega\downarrow 0.
\end{aligned}$$

Hence, the first two moments of this random variable are just

$$\mathbb{E}(D_{LCFS}) = (1+\rho_1)\mathbb{E}\left(R_C\right)$$

and

$$\mathbb{E}\left(D_{LCFS}^2\right) = \lambda_1\mathbb{E}\left(B_1^2\right)\mathbb{E}(R_C)+(1+\rho_1)^2\mathbb{E}\left(R_C^2\right).$$

From this, we can now compute the first and second moments of the sojourn time:

$$\mathbb{E}(T_{LCFS}) = \mathbb{E}(B_1)+\mathbb{E}(R_C)(1+\rho_1) = \mathbb{E}(T_{FCFS}),$$

and

$$\begin{aligned}
\mathbb{E}\left(T_{LCFS}^2\right) &= \mathbb{E}\left(B_1^2\right)+\mathbb{E}(R_C)\left(2(1+\rho_1)\mathbb{E}(B_1)+\lambda_1\mathbb{E}\left(B_1^2\right)\right)\\
&+ \mathbb{E}\left(R_C^2\right)(1+\rho_1)^2\\
&= \mathbb{E}\left(T_{FCFS}^2\right)+\rho_1\mathbb{E}\left(R_C^2\right).
\end{aligned}$$

Thus, we see that the second moment of $T_{LCFS}$ is larger than the one of $T_{FCFS}$, which proves that the sojourn time under LCFS is actually more variable than its FCFS counterpart. This ordering between the second moments was already established on pgs. 283-284 of Wolff [96] (see also Shanthikumar and Sumita [76], whose result is mentioned on pg. 257 of Fuhrmann and Iliadis [44]), but in this case we are able to prove it by giving an explicit calculation. Moreover, this difference between the second moments of $T_{FCFS}$ and $T_{LCFS}$ is not surprising, as this is basically due to the simple fact that (cf. (5.14))

$$\mathbb{E}\left(C^pC^r\right) = \frac{\mathbb{E}\left(\left(C^r\right)^2\right)}{2} < \mathbb{E}\left(\left(C^r\right)^2\right).$$

In other words, large values of $C^r$ intuitively imply that many jobs will enter the system during $C^r$, and similarly for small $C^r$; thus, $T_{LCFS}$ should exhibit more variability than $T_{FCFS}$.

### 5.4.3 Random Order of Service

The next policy that we will analyse in this chapter is known as the Random Order of Service (ROS) policy. Unfortunately, the LST of the sojourn time under this policy isn't as nice as the previous cases, as the reader will see from the derivation below. To compute the LST, let us assign to each customer that arrives at time $T_{i,n}$ a mark $U_{i,n}$, where $U_{i,n}$ is a uniform random variable on $[0, 1]$. We assume that these new marks are independent of all other random elements in the space. Once the server visits $Q_i$, the order at which it serves the customers currently waiting there is determined by the $U_{i,n}$ marks; we will assume that the customer with the smallest $U$ mark is served first, the second-smallest served second, and so on. In the work of Fuhrmann and Iliadis [44], this type of service discipline is known as the Randomly Assigned Priorities (RAP) discipline, but in our system it is clearly equivalent to the ROS discipline. Throughout the rest of this chapter, we will refer to the $U$ marks as ordering marks.

We first compute the LST of $T_{ROS}$, conditional on $x$ being the ordering mark of the tagged customer. Due to classical thinning properties of Poisson processes, we see that

$$T_{ROS}(x) \overset{d}{=} C^r + B_1 + \sum_{T_{1,x,k} \in (-C^p, C^r)} B_{1,k}$$

where $T_{1,x,k}$ correspond to the arrival points of a new Poisson process with rate $\lambda_1 x$. Thus,

$$\begin{aligned}
\mathbb{E}\left(e^{-\omega T_{ROS}(x)}\right) &= \mathbb{E}\left(e^{-\omega\left(C^r + B_1 + \sum_{T_{1,x,k} \in (-C^p, C^r)} B_{1,k}\right)}\right) \\
&= \beta_1(\omega)\mathbb{E}\left(e^{-\lambda_1 x(1-\beta_1(\omega))C^p - (\omega + \lambda_1 x(1-\beta_1(\omega)))C^r}\right) \\
&= \beta_1(\omega)\left[\frac{\mathbb{E}\left(e^{-\lambda_1 x(1-\beta_1(\omega))C}\right) - \mathbb{E}\left(e^{-(\omega+\lambda_1 x(1-\beta_1(\omega)))C}\right)}{\omega\mathbb{E}(C)}\right].
\end{aligned}$$

Now we see that the transform is of the same form as before, and so we can apply Lemma 3.1 to compute the conditional first and second moments. In this case, we see that the conditional first moment is just

$$\mathbb{E}(T_{ROS}(x)) = \mathbb{E}[B_1] + (1 + 2\rho_1 x)\mathbb{E}(R_C)$$

and the second conditional moment is

$$\begin{aligned}\mathbb{E}\left(T^2_{ROS}(x)\right) &= 2\lambda_1\mathbb{E}\left(B_1^2\right)x + (3\rho_1^2x^2 + 3\rho_1 x + 1)\mathbb{E}\left(R_C^2\right) \\ &+ 2(1+2\rho_1 x)\mathbb{E}(B_1)\mathbb{E}(R_C) + \mathbb{E}\left(B_1^2\right).\end{aligned}$$

Finally, the unconditional moments of $T_{ROS}$ can be computed by integrating out $x$ with respect to a uniform density on $[0,1]$. Thus, the first moment is

$$\mathbb{E}(T_{ROS}) = \mathbb{E}(B_1) + \mathbb{E}(R_C)(1+\rho_1).$$

The second moment is just

$$\begin{aligned}\mathbb{E}\left(T^2_{ROS}\right) &= \mathbb{E}\left(B_1^2\right) + \mathbb{E}(R_C)\left(2(1+\rho_1)\mathbb{E}(B_1) + \lambda_1\mathbb{E}\left(B_1^2\right)\right) \\ &+ \frac{\mathbb{E}\left(R_C^2\right)}{2}\left(2 + 3\rho_1 + 2\rho_1^2\right) \\ &= \mathbb{E}\left(T^2_{FCFS}\right) + \frac{\rho_1}{2}\mathbb{E}\left(R_C^2\right).\end{aligned}$$

The term $\rho_1\mathbb{E}(R_C^2)/2$ is explained by the fact that, under ROS, the amount of work in $Q_1$ that is served before the tagged customer partly depends on $C^r$. Under the FCFS discipline, this amount of work only depends on $C^p$. From (5.14) we know that $\mathbb{E}(\rho C^r C^p) = \rho\mathbb{E}\left(R_C^2\right)/2$, while $\mathbb{E}\left(\rho C^r C^r\right) = \rho\mathbb{E}\left(R_C^2\right)$. Taking the product of the amount of work served before the tagged customer and the residual cycle length results in $2\mathbb{E}(\sum_{k=1}^N B_{1,k}C^r)$, where $N$ depends on either $C^p$ (FCFS) or both $C^p$ and $C^r$ (ROS). Hence, for ROS we get $\rho\mathbb{E}\left(R_C^2\right)$, while for FCFS it is just $\rho\mathbb{E}\left(R_C^2\right)/2$. The difference between $\mathbb{E}\left(T^2_{LCFS}\right)$ and $\mathbb{E}\left(T^2_{ROS}\right)$ is exactly the same and can be explained in an analogous manner.

Clearly $\mathbb{E}\left(T^2_{LCFS}\right) > \mathbb{E}\left(T^2_{ROS}\right) > \mathbb{E}\left(T^2_{FCFS}\right)$. Moreover, we have also established that the second moment of $T_{ROS}$ is precisely in between the second moments of $T_{FCFS}$ and $T_{LCFS}$. We should again point out that such an ordering is already known (see Shanthikumar and Sumita [76]) for more general types of queues. The main point is to show that the moments can, in fact, be calculated.

### 5.4.4 The Processor Sharing and Shortest-Job-First Disciplines

The next two policies that we consider in this section are the Shortest Job First (SJF) and the Processor Sharing (PS) policies. Suppose that when the server arrives to $Q_1$, it orders the jobs in increasing order, i.e., let $B_{1,(k)}$ denote the $k^{th}$-smallest job in $Q_1$, where $1 \leq k \leq N_1\left(-C^p, C^r\right)$. Then it is clear that, if $U$ denotes the position of the tagged customer among the ordered (from smallest to largest) list of service times of customers waiting in $Q_1$ at the moment they begin to receive service, then

$$\begin{aligned}T_{PS} &= C^r + \sum_{k=1}^{U}\left(N_1\left(C^p\right) + N_1\left(C^r\right) + 1 - k + 1\right) \\ &\times \left(B_{1,(k)} - B_{1,(k-1)}\right) \qquad (5.22)\end{aligned}$$

and

$$T_{SJF} = C^r + \sum_{k=1}^{U} B_{1,(k)}. \tag{5.23}$$

Here we use the convention that $B_{1,(0)} = 0$ with probability one.

Unfortunately, working with order statistics is often a cumbersome task; consequently, we will not be able to explicitly compute the LST of either $T_{PS}$ or $T_{SJF}$, for an arbitrary service time distribution. The reader may notice, however, that if the services are exponentially distributed, then $T_{PS}$ is equal in distribution to $T_{ROS}$. This follows from the following simple property of exponential random variables (see, for instance, page 19 of Feller [40]):

**Proposition 5.1.** *Let $X_1$, $X_2$, ..., $X_n$ denote a collection of $n$ independent and identically distributed exponential random variables with rate $\alpha$. If $X_{(k)}$ denotes the $k^{th}$-smallest random variable among the population, then the $n$ variables $X_{(k)} - X_{(k-1)}$, $1 \le k \le n$ (set $X_{(0)} = 0$) are independent and $X_{(k)} - X_{(k-1)}$ is exponentially distributed with rate $(n-k+1)\alpha$.*

Even in this case, however, the distribution of $T_{SJF}$ is still difficult to handle. To get around this dilemma, we will need to condition on the service time of the tagged customer. Throughout the rest of this subsection we assume that the service time distributions of all customers in the system are absolutely continuous (i.e. they have a density). If not, two customers at $Q_1$ that are served in the same cycle could possibly bring exactly the same amount of work to the system, and such cases have to be carefully handled. However, this is not difficult, and we leave it to the interested reader to calculate.

### Conditioning on the service time

Suppose that a tagged customer arrives to $Q_1$ with an amount of work $x$. Then the sojourn time of the customer depends on three things: the remaining amount of time it takes for the server to reach $Q_1$, and the amounts of work brought by customers that arrived before, and after, the tagged customer to $Q_1$. For a given scheduling policy $\Gamma$, let $T_\Gamma(x)$ denote the sojourn time of a tagged customer, conditional on the amount of work it brings to the system. Under many policies, this random variable can be written in the following way:

$$T_\Gamma(x) = x + C^r + \sum_{T_{1,k} \in (-C^p, 0)} g_1\left(B_{1,k}, x\right) + \sum_{T_{1,k} \in (0, C^r)} g_2(B_{1,k}, x). \tag{5.24}$$

Here $g_i : [0,\infty) \times [0,\infty) \to \mathbb{R}$, $i = 1, 2$ are functions that capture how the tagged customer's sojourn time is affected by customers that arrive before and after him, respectively. For example, if $\Gamma$ represents the FCFS policy, $g_1(y, x) = y$ and

$g_2(y,x) = 0$, since all customers arriving ahead of the tagged customer will be served first, and no customer arriving afterward will affect the sojourn time. The reader should of course keep in mind that $g_i$ could depend on $x$ as well (such as when analyzing the SJF case), which is why we allow $g_i$ to depend on $x$. This idea of using equation (5.24) to model various service disciplines was used in Winands et al. [91] to compute the mean sojourn times: we will show throughout the rest of this section that it can be used to calculate transforms as well.

Modeling the sojourn times in this manner will allow us to easily compute the LST of $T_\Gamma(x)$. For $\omega \geq 0$, we find that

$$\begin{aligned}\mathbb{E}\left(e^{-\omega T_\Gamma(x)}\right) &= \\ & e^{-\omega x}\mathbb{E}\left(e^{-\omega\left(C^r+\sum_{T_{1,k}\in(-C^p,0)} g_1(B_{1,k},x)+\sum_{T_{1,k}\in(0,C^r)} g_2(B_{1,k},x)\right)}\right) \\ &= e^{-\omega x}\int_0^\infty\int_0^\infty e^{-\omega v}e^{-\lambda_1(1-\phi_1(\omega,x))u} \\ &\times\ e^{-\lambda_1(1-\phi_2(\omega,x))v}\mathrm{d}P\left(C^p\leq u, C^r\leq v\right),\end{aligned}$$

where $\phi_i(\omega,x) = \mathbb{E}(e^{-\omega g_i(B_1,x)})$, for $i = 1,2$, and $B_{\phi,i}$ denotes a random variable with LST $\phi_i$. Therefore,

$$\begin{aligned}\mathbb{E}\left(e^{-\omega T_\Gamma(x)}\right) &= \\ & e^{-\omega x}\frac{\mathbb{E}\left(e^{-\lambda_1(1-\phi_1(\omega,x))C}\right)-\mathbb{E}\left(e^{-(\omega+\lambda_1(1-\phi_2(\omega,x)))C}\right)}{\mathbb{E}(C)\left(\omega+\lambda_1(\phi_1(\omega,x)-\phi_2(\omega,x))\right)}.\end{aligned} \tag{5.25}$$

Showing that the SJF policy fits within this framework is simple: just set $g_1(y,x) = g_2(y,x) = y\mathbf{1}(y\leq x)$. This follows from the fact that all, and only all, jobs present that are of a size smaller than $x$ will be served before the tagged customer.

The PS discipline can also be modeled in this manner, if we choose $g_1(y,x) = g_2(y,x) = \min(y,x)$. It is simple to verify that these functions correctly model the processor-sharing phenomenon.

### Processor Sharing

Now we are ready to analyse the sojourn time of a tagged customer at $Q_1$, which utilizes the processor-sharing rule while it is serving customers waiting in front of the gate.

It should be noted that similar models, for the single-queue case, have been studied in the literature before. Rege and Sengupta [70], for instance, derive various performance measures for what is known as a gated $M/M/1$ queue, which operates as follows: the server provides service to at most $m \geq 1$ customers, in a processor-sharing fashion. Once a group has been served, the server then begins serving the next (up to) $m$ waiting customers, and so on. The works of Avi-Itzhak and

Halfin [6] and Rietman and Resing [72] focus on various extensions of this model. In particular, [6] considers a gated $M/G/1$ queue, and they consider not only the processor-sharing discipline, but other "conservative" scheduling disciplines, which include FCFS, LCFS, and ROS. They also analyse the same type of model in [72], but they go a step further by deriving the joint distribution of both the amount of time a customer spends on both sides of the gate, and the number of customers on both sides of the gate.

We will now begin our calculation of the conditional LST of the sojourn time under PS. From (5.25), we see that

$$\mathbb{E}\left(e^{-\omega T_{PS}(x)}\right) = \\ e^{-\omega x}\frac{\mathbb{E}\left(e^{-\lambda_1(1-\phi(\omega,x))C}\right) - \mathbb{E}\left(e^{-(\omega+\lambda_1(1-\phi(\omega,x)))C}\right)}{\omega\mathbb{E}(C)}, \tag{5.26}$$

where $\phi(\omega, x) = \mathbb{E}\left(e^{-\omega \min(B_k,x)}\right)$. This expression is nice, in that it is given in terms of the LST of the cycle time. To find the unconditional LST of $T_{PS}$, we only need to integrate with respect to the service time distribution, however in many cases this transform will not be tractable.

We will now use this transform to calculate the first and second moments. For first moments, we see that an application of Wald's equality can be used to compute the first moment of $T_{PS}(x)$ by using (5.24), and it can also be found in Winands et al. [91].

In this case,

$$\mathbb{E}(T_{PS}(x)) = x + \mathbb{E}\left(C^r\right)(1 + 2\rho_{1,PS}(x)),$$

where $\rho_{1,PS}(x) = \lambda_1\mathbb{E}[\min(B_1, x)] = \lambda_1\mathbb{E}[B_\phi]$. One can easily check that this result also agrees with the first moment calculation found in Avi-Itzhak and Halfin [6], where they essentially look at the special case of a polling system with zero setup times, and only one buffer.

At first glance, the LST of $T_{PS}(x)$ doesn't look like a nice function to differentiate, but it is still not too difficult to make use of it in order to compute the first and second moment. By applying Lemma 5.1 to (5.26), we find that since $D_{PS}(x) = T_{PS}(x) - x$ (recall the discussion of the use of the $D$-notation below (5.20))

$$\begin{aligned}\mathbb{E}\left(e^{-\omega D_{PS}(x)}\right) &= \sum_{k=0}^{n}(-(1-\phi(\omega)))^k\lambda_1^k\frac{\mathbb{E}\left(R_C^k\right)}{k!} \\ &\times\ \mathbb{E}\left(e^{-(\omega+\lambda_1(1-\phi(\omega)))R_{C,k+1}}\right) + \mathcal{O}\left(\omega^{n+1}\right), \qquad \omega\downarrow 0.\end{aligned}$$

Furthermore, since

$$1-\phi(\omega) = \mathbb{E}(B_\phi)\omega - \frac{\mathbb{E}\left(B_\phi^2\right)}{2}\omega^2 + \mathcal{O}\left(\omega^3\right), \qquad \omega\downarrow 0,$$

and

$$\begin{aligned}\omega + \lambda_1(1-\phi(\omega)) &= (1+\rho_{1,PS}(x))\,\omega \\ &- \lambda_1 \frac{\mathbb{E}\left(B_\phi^2\right)}{2}\omega^2 + \mathcal{O}\left(\omega^3\right), \qquad \omega \downarrow 0,\end{aligned}$$

we have for $\omega \downarrow 0$:

$$\begin{aligned}\mathbb{E}\left(e^{-\omega D_{PS}(x)}\right) &= 1 - \mathbb{E}(R_C)\,(1+2\rho_{1,PS}(x))\,\omega + \lambda_1 \mathbb{E}\left(B_\phi^2\right)\mathbb{E}(R_C)\omega^2 \\ &+ \frac{\mathbb{E}\left(R_C^2\right)}{2}\left(1+3\rho_{1,PS}(x)+3\rho_{1,PS}^2(x)\right)\omega^2 + \mathcal{O}\left(\omega^3\right).\end{aligned}$$

This expression shows that the first and second moments of the conditional delay are just

$$\mathbb{E}(D_{PS}(x)) = \mathbb{E}(R_C)\,(1+2\rho_{1,PS}(x))$$

and

$$\begin{aligned}\mathbb{E}\left(D_{PS}^2(x)\right) &= 2\lambda_1 \mathbb{E}\left(B_\phi^2\right)\mathbb{E}(R_C) \\ &+ \mathbb{E}\left(R_C^2\right)\left(1+3\rho_{1,PS}(x)+3\rho_{1,PS}^2(x)\right).\end{aligned}$$

Before we compute the unconditional moments, let us first consider the finite collection of i.i.d. random variables $\{B_{1,k}\}_{k=1}^n$, where $B_{1,1}$ is equal in distribution to a typical amount of work that is brought by the customer that visits $Q_1$. Then, if we let $B_{1,k:n}$ denote the $k^{th}$ smallest value among this collection of size $n$, we see that

$$\begin{aligned}\int_0^\infty \rho_{1,PS}(x)\mathrm{d}B_{1,1}(x) &= \lambda_1 \int_0^\infty \mathbb{E}\left(\min(B_{1,1},x)\right)\mathrm{d}B_{1,1}(x) \\ &= \lambda_1 \mathbb{E}(B_{1,1:2}),\end{aligned}$$

$$\lambda_1 \int_0^\infty \mathbb{E}\left(\min(B_{1,1},x)^2\right)\mathrm{d}B_{1,1}(x) = \lambda_1 \mathbb{E}\left(B_{1,1:2}^2\right),$$

and

$$\begin{aligned}\int_0^\infty \rho_{1,PS}^2(x)\mathrm{d}B_{1,1}(x) &= \lambda_1^2 \int_0^\infty \mathbb{E}\left(\min(B_{1,1},x)\right)^2 \mathrm{d}B_{1,1}(x) \\ &= \lambda_1^2 \mathbb{E}\left(\min(B_{1,1},B_{1,3})\min(B_{1,2},B_{1,3})\right) \\ &= \lambda_1^2\left[\frac{2\mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right)}{3} + \frac{\mathbb{E}\left(B_{1,1:3}^2\right)}{3}\right].\end{aligned}$$

Thus, we see that the unconditional first moment of the delay is

$$\mathbb{E}(D_{PS}) = \mathbb{E}(R_C)(1 + 2\lambda_1 \mathbb{E}(B_{1,1:2}))$$

and the second moment is just

$$\begin{aligned}&\mathbb{E}\left(D_{PS}^2\right) = 2\lambda_1 \mathbb{E}\left(B_{1,1:2}^2\right)\mathbb{E}(R_C)\\ &+\ \mathbb{E}\left(R_C^2\right)\left(1 + 3\lambda_1\mathbb{E}(B_{1,1:2}) + \lambda_1^2\left[2\mathbb{E}(B_{1,1:3}B_{1,2:3}) + \mathbb{E}\left(B_{1,1:3}^2\right)\right]\right).\end{aligned}$$

After a few more quick calculations, the reader will find that the first and second moments of the sojourn time are as follows:

$$\begin{aligned}\mathbb{E}(T_{PS}) &= \mathbb{E}(B_{1,1}) + \mathbb{E}(R_C)(1 + 2\lambda_1\mathbb{E}(B_{1,1:2})),\\ \mathbb{E}\left(T_{PS}^2\right) &= \mathbb{E}\left(B_{1,1}^2\right) + \mathbb{E}(R_C)\left(2(1+\rho_1)\mathbb{E}(B_{1,1}) + 4\lambda_1\mathbb{E}\left(B_{1,1:2}^2\right)\right)\\ &+\ 2\mathbb{E}(R_C)\lambda_1\mathbb{E}\left(B_{1,1}\right)^2 + \mathbb{E}\left(R_C^2\right)(1 + 3\lambda_1\mathbb{E}(B_{1,1:2}))\\ &+\ \mathbb{E}\left(R_C^2\right)\lambda_1^2\left[2\mathbb{E}(B_{1,1:3}B_{1,2:3}) + \mathbb{E}\left(B_{1,1:3}^2\right)\right].\end{aligned}$$

**Remark** It may be of interest to find all values $x$ where $\mathbb{E}(T_{PS}(x)) \leq \mathbb{E}(T_{FCFS}(x))$, and where $\mathbb{E}(T_{PS}(x)) \geq \mathbb{E}(T_{FCFS}(x))$. If we assume that the distribution of $B_1$ is absolutely continuous (i.e. has a density), an application of the dominated convergence theorem shows that the set of points where $\mathbb{E}(T_{PS}(x)) \geq \mathbb{E}(T_{FCFS}(x))$ is of the form $[x_{PS}, \infty)$, where $x_{PS}$ is the solution to the equation

$$\mathbb{E}(\min(B_1, x)) = \mathbb{E}(B_1)/2.$$

After some simple manipulations, we see that $x_{PS}$ satisfies

$$\int_0^{x_{PS}} \overline{B_1}(t)\mathrm{dt} = \mathbb{E}(\mathrm{B}_1)/2,$$

with $\overline{B_1}(t) = P(B_1 > t)$. This implies that $x_{PS}$ is the median of the residual service time distribution. Notice that if $B_1$ is exponential, then this is just the median of an exponential distribution, and so we can conclude that in this case, half of all customers that arrive to the system will experience a shorter expected sojourn time if the system operates under FCFS, and the other half will experience a shorter expected sojourn time under PS.

The exact difference between the second moments of $T_{PS}$ and $T_{FCFS}$ is just

$$\begin{aligned}&\mathbb{E}\left(T_{PS}^2\right) - \mathbb{E}\left(T_{FCFS}^2\right) = \mathbb{E}(R_C)\lambda_1 4\mathbb{E}\left(B_{1,1:2}^2\right)\\ &+\ \mathbb{E}\left(R_C^2\right)\left(3\lambda_1\mathbb{E}\left(B_{1,1:2}\right) - \rho_1 + 2\lambda_1^2\mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right)\right)\\ &+\ \mathbb{E}\left(R_C^2\right)\left(\lambda_1^2\mathbb{E}\left(B_{1,1:3}^2\right) - \rho_1^2\right).\end{aligned}$$

The term $\mathbb{E}(R_C)\lambda_1 4\mathbb{E}\left(B_{1,1:2}^2\right)$ is explained by the fact that, under the PS service discipline, the effect an arriving customer has on the waiting time of the tagged customer (i.e. an arriving customer adds an additional $\min(B_{1,1}, B)$ to the delay) and the service time of the tagged customer are dependent. Therefore, we get the following term:

$$2\mathbb{E}\left(\sum_{k=1}^{N(-C^p,C^r)} \min\left(B_{1,k}B_1\right) B_1\right) =$$

$$\mathbb{E}\left(R_C\right)\left(2\lambda_1\mathbb{E}\left(B_{1,1:2}^2\right) + 2\lambda_1\mathbb{E}\left(B_1\right)^2\right).$$

Under FCFS, there is no dependence between the waiting time due to an arriving customer and the service time of the tagged customer. Furthermore, only the customers who arrive during $C^p$ cause a delay for the tagged customer. Therefore, we only need to subtract $2\mathbb{E}\left(\sum_{k=1}^{N(-C^p,0]} B_{1,k}B_1\right) = 2\mathbb{E}\left(R_C\right)\lambda_1\mathbb{E}\left(B_1\right)^2$ in the equation above. An extra term of $2\lambda_1\mathbb{E}\left(R_C\right)\mathbb{E}\left(B_{1,1:2}^2\right)$ appears, because an average number of $2\mathbb{E}\left(R_C\right)\lambda_1$ customers arrive during $\left(-C^p, C^r\right)$ who cause a delay for the tagged customer with second moment $\mathbb{E}\left(B_{1,1:2}^2\right)$.

The second term, $\mathbb{E}\left(R_C^2\right)\left(3\lambda_1\mathbb{E}\left(B_{1,1:2}\right) - \rho_1\right)$, is just the difference between $\mathbb{E}\left(C^r\sum_{k=1}^{N(-C^p,C^r)} B_{1,k}^*\right)$ and $\mathbb{E}\left(C^r\sum_{k=1}^{N(-C^p,0)} B_{1,k}\right)$, with $B_{1,k}^*$ representing the amount of work brought by customer $k$ that influences the delay of the tagged customer. Under PS, this amount of time is the minimum of two service times, while under FCFS it is one complete service time. Furthermore, $T_{PS}$ is influenced by customers that arrive during both $C^p$ *and* $C^r$, while under FCFS it is only influenced by customers that arrive during $C^p$, which explains why $\lambda_1\mathbb{E}\left(B_{1,1:2}\right)$ is multiplied by a factor 3 (see (5.14)).

Similarly, the third term $\mathbb{E}\left(R_C^2\right)\left(2\lambda_1^2\mathbb{E}(B_{1,1:3}B_{1,2:3}) + \lambda_1^2\mathbb{E}\left(B_{1,1:3}^2\right) - \rho_1^2\right)$ corresponds to the difference

$$\mathbb{E}\left(\sum_{k=1}^{N(-C^p,C^r)} B_{1,k}^* \sum_{j=1,j\neq k}^{N(-C^p,C^r)} B_{1,j}^*\right) - \mathbb{E}\left(\sum_{k=1}^{N(-C^p,0)} B_{1,k} \sum_{j=1,j\neq k}^{N(-C^p,0)} B_{1,j}\right).$$

Let $B_{1,k}^*B_{1,j}^*$ denote the product of two different (parts of) service times for which the tagged customer has to wait. Now, with probability 1/3, $B_1$ is smaller than both $B_{1,k}^*$ and $B_{1,j}^*$, and in that case $B_{1,k}^*B_{1,j}^* = B_{1,1:3}B_{1,1:3}$. In all other cases, $B_{1,k}^*B_{1,j}^* = B_{1,1:3}B_{1,2:3}$ (either $B_{1,k}^*$ or $B_{1,j}^*$ is the minimum among the three service times). Under the PS discipline, the number of customers that influence the service time of the tagged customer is $N\left(-C^p, C^r\right)$, and its second factorial moment equals $3\lambda_1^2\mathbb{E}\left(R_C^2\right)$. Multiplying both results gives us exactly $\mathbb{E}\left(R_C^2\right)\lambda_1^2\left(2\mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right) + \mathbb{E}\left(B_{1,1:3}^2\right)\right)$. One can check that under FCFS, this term is just $\rho_1^2$.

### Shortest Job First

Now we will present the LST for the sojourn time of a tagged customer that visits $Q_1$ under the Shortest Job First policy. Due to the fact that $g_1 = g_2$ under this policy as well,

$$\mathbb{E}\left(e^{-\omega T_{SJF}(x)}\right) = e^{-\omega x}\frac{\mathbb{E}\left(e^{-\lambda_1(1-\phi(\omega,x))C}\right) - \mathbb{E}\left(e^{-(\omega+\lambda_1(1-\phi(\omega,x)))C}\right)}{\omega\mathbb{E}(C)},$$

but in this case $\phi(\omega, x) = \mathbb{E}\left(e^{-\omega B_1 \mathbf{1}(B_1 \le x)}\right)$. At this point, we can manipulate the transform for this sojourn time in precisely the same manner as was done for the processor-sharing case given above, because we never made explicit use of the form of $\phi$. Therefore, the first and second moment of $D_{SJF}(x)$ are as follows:

$$\mathbb{E}\left(D_{SJF}(x)\right) = (1 + 2\rho_{1,SJF}(x))\mathbb{E}(R_C)$$

and

$$\begin{aligned}\mathbb{E}\left(D_{SJF}(x)^2\right) &= 2\lambda_1\mathbb{E}\left(B_\phi^2\right)\mathbb{E}(R_C)\\ &+ \mathbb{E}\left(R_C^2\right)\left(1 + 3\rho_{1,SJF}(x) + 3\rho_{1,SJF}^2(x)\right),\end{aligned}$$

however in this case $\rho_{1,SJF}(x) = \lambda_1\mathbb{E}(B_{1,1}\mathbf{1}(B_{1,1} \le x))$. The unconditional moments can also be computed, as in the PS case. Indeed,

$$\begin{aligned}\int_0^\infty \mathbb{E}\left(B_{1,1}\mathbf{1}(B_{1,1} \le x)\right)\mathrm{d}B(x) &= \mathbb{E}\left(B_{1,1}\mathbf{1}(B_{1,1} \le B_{1,2})\right)\\ &= \frac{\mathbb{E}\left(B_{1,1:2}\right)}{2},\\ \int_0^\infty \mathbb{E}\left(B_{1,1}^2\mathbf{1}(B_{1,1} \le x)\right)\mathrm{d}B(x) &= \frac{\mathbb{E}\left(B_{1,1:2}^2\right)}{2},\\ \int_0^\infty \mathbb{E}\left(B_{1,1}\mathbf{1}(B_{1,1} \le x)\right)^2\mathrm{d}B(x) &= \frac{\mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right)}{3},\end{aligned}$$

and so by inserting these expressions into our conditional moments, we find that

$$\begin{aligned}\mathbb{E}(T_{SJF}) &= \mathbb{E}(B_{1,1}) + \mathbb{E}(R_C)\left(1 + \lambda_1\mathbb{E}(B_{1,1:2})\right),\\ \mathbb{E}\left(T_{SJF}^2\right) &= \mathbb{E}\left(B_{1,1}^2\right) + \mathbb{E}(R_C)\left(2(1+\rho_1)\mathbb{E}(B_{1,1}) + \lambda_1\mathbb{E}\left(B_{1,1:2}^2\right)\right)\\ &+ \mathbb{E}\left(R_C^2\right)\left(1 + 3\lambda_1\frac{\mathbb{E}\left(B_{1,1:2}\right)}{2} + \lambda_1^2\mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right)\right).\end{aligned}$$

It was shown in Winands et al. [91] that $\mathbb{E}(T_{SJF})$ is the smallest first moment among all first sojourn time moments considered. Furthermore, we also find that $\mathbb{E}\left(T_{SJF}^2\right) \le \mathbb{E}\left(T_{FCFS}^2\right)$ if $\mathbb{E}\left(B_{1,1:2}\right) \le (2/3)\mathbb{E}(B_{1,1:2})$. Such an inequality is satisfied when the service time distribution is DFR (i.e. has a decreasing failure rate); see pg. 1014 of Winands et al. [91].

In particular, for exponential service times it is easy to show that

$$\mathbb{E}\left(T_{SJF}^2\right) < \mathbb{E}\left(T_{FCFS}^2\right) < \mathbb{E}\left(T_{ROS}^2\right) < \mathbb{E}\left(T_{LCFS}^2\right)$$

and $\mathbb{E}\left(T^2_{ROS}\right) = \mathbb{E}\left(T^2_{PS}\right)$.

It is also important to note that we can establish meaningful comparisons between $\mathbb{E}\left(T^2_{SJF}\right)$ and $\mathbb{E}\left(T^2_{FCFS}\right)$, that are similar to those found at the end of Section 5.4.4. We note that under SJF, the number of customers that arrived in $(-C^p, C^r)$ and are served before the tagged customer is just $N_1\left(-C^p, C^r\right)$. Furthermore, each arrival in $(-C^p, C^r)$ influences the amount of time a tagged customer has to wait with the amount $B_{1,k}(< B_1)$, with probability $1/2$. Therefore, one can look at the queue as if customers arrive with rate $\lambda/2$ and bring waiting times (for the tagged customer) distributed as service times that are smaller than $B_1$, the service time of the tagged customer. The terms $\lambda_1 \mathbb{E}\left(B^2_{1,1:2}\right)\mathbb{E}(R_C)$ and $\mathbb{E}\left(R^2_C\right) 3\lambda_1 \mathbb{E}(B_{1,1:2})/2$ are both explained if you combine these two remarks with the result in (5.14). In order to explain the term $\mathbb{E}\left(R^2_C\right)\lambda_1^2 \mathbb{E}\left(B_{1,1:3}B_{1,2:3}\right)$, we consider two customers who are served before the tagged customer and let their service times be $B_{1,j}$ and $B_{1,k}$. It then holds that $\mathbb{E}(B_{1,j}B_{1,k}) = \mathbb{E}(B_{1,1:3}B_{1,2:3})$, because both service times are smaller than the service time of the tagged customer.

## 5.5 A globally gated polling regime

In this section, we compute the LST of the sojourn time $T_{\Gamma,i}$ of an arbitrary type-$i$ customer in a globally gated polling system that serves customers at $Q_i$ according to policy $\Gamma$. In such a polling system, the server serves only the customers who are present at the start of the cycle, i.e. a gate is placed behind every queue just before the server polls the first queue. This polling regime is not within the class of branching type polling disciplines, but it satisfies Property 5.1, which allows us to decompose $T_{\Gamma,i}$ into the sum of four parts which only depend on the total and the residual length ($C^*$ and $C^r$) of the cycle in which a tagged customer arrives. For such a tagged customer in steady-state, these four parts are defined by:

1. the residual cycle length $C^r$,

2. the service times of all customers of type $j = 1, \ldots, i-1$ that arrive during $C^p$ and $C^r$, where the sum of all type $j$ customers arriving in $(-C^p, C^r)$ form the visit time $V_j$ at $Q_j$, $j = 1, \ldots, i-1$,

3. $R_i$, the time interval between the polling epoch of $Q_i$ in the following cycle, and the departure epoch of the tagged customer,

4. the switch-over times $S_1, \ldots, S_{i-1}$.

The LST of the total cycle time is derived in Boxma et al. [17] and satisfies

$$\gamma(\omega) = \mathbb{E}(e^{-\omega C}) = \prod_{i=1}^{\infty} \sigma(\delta^{(i)}(\omega)),$$

with

$$\begin{aligned}
\sigma(\omega) &= \mathbb{E}(e^{-\omega \sum_{i=1}^{N} S_i}) = \prod_{j=1}^{N} \sigma_j(\omega), \\
\delta^{(0)}(\omega) &= \omega, \\
\delta^{(i)}(\omega) &= \delta(\delta^{(i-1)}(\omega)), \\
\delta(\omega) &= \sum_{j=1}^{N} \lambda_j (1 - \beta_j(\omega)).
\end{aligned}$$

In the same chapter, the LST of the waiting time in $Q_i$ with a FCFS service discipline is derived. This result will be discussed in the following section.

### 5.5.1 First Come First Served

In Boxma et al. [17], the LST of the sojourn time in $Q_i$ of a globally gated system with a FCFS service discipline is given:

$$\begin{aligned}
\mathbb{E}(e^{-\omega T_{FCFS,i}}) &= \left( \prod_{j=1}^{i-1} \sigma_j(\omega) \right) \\
\times \quad \frac{1}{\mathbb{E}C} & \frac{\gamma\left(\sum_{j=1}^{i} \lambda_j (1 - \beta_j(\omega))\right) - \gamma\left(\sum_{j=1}^{i-1} \lambda_j (1 - \beta_j(\omega)) + \omega\right)}{\omega - \lambda_i (1 - \beta_i(\omega))}.
\end{aligned}$$

The first and second moment of $T_i$ in FCFS can be derived with Taylor series approximations in the numerator and the denominator. We find

$$\begin{aligned}
\mathbb{E}\left(T_{FCFS,i}\right) &= \mathbb{E}\left(B_i\right) + \sum_{j=1}^{i-1} \mathbb{E}\left(S_j\right) \\
&+ \mathbb{E}\left(R_C\right) \left( 2 \sum_{j=1}^{i-1} \rho_j + \rho_i + 1 \right) \qquad (5.27)
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\left(T^2_{FCFS,i}\right) &= \mathbb{E}\left(B_i^2\right) + \mathbb{E}\left(\left(\sum_{j=1}^{i-1} S_j\right)^2\right) + 2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right) \\
&+ \ \mathbb{E}\left(R_C\right)\left[2\left(\rho_i+1\right)\mathbb{E}\left(B_i\right) + \lambda_i\mathbb{E}\left(B_i^2\right) + 2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right)\right] \\
&+ \ \mathbb{E}\left(R_C\right)\left[4\sum_{j=1}^{i-1}\rho_j\mathbb{E}\left(B_i\right) + \left(4\sum_{j=1}^{i-1}\rho_j + 2\rho_i + 2\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\right] \\
&+ \ \mathbb{E}\left(R_C^2\right)\left(3\left(\sum_{j=1}^{i-1}\rho_j\right)^2 + \rho_i\left(\rho_i+1\right)\right) \\
&+ \ \mathbb{E}\left(R_C^2\right)\left(1 + 3\sum_{j=1}^{i-1}\rho_j\left(\rho_i+1\right)\right),
\end{aligned} \tag{5.28}$$

where $R_C$ is equal in distribution to $C^r$, as pointed out in Section 5.3.1. The result for the second moment of $T_{FCFS}$ can be compared with the results in the following subsections, but these comparisons would be similar to the ones discussed in Section 5.4. It is easily seen that the amount of time a customer has to wait before the server visits its queue does not depend on the service discipline. Therefore, the differences between the second moments are only caused by terms that include the amounts of work brought by customers that arrive to $Q_i$ during the cycle $(-C^p, C^r)$ (including the tagged customer), which was also the case in the system with gated visit disciplines. For a comparison of the second moments of the sojourn times, we thus refer to the previous section.

### 5.5.2 Last Come First Served

In the LCFS policy, $R_i$ consists only of the service times of the customers who arrive during the residual cycle and the service time of the tagged customer. So we get

$$\begin{aligned}
\mathbb{E}\left(e^{-\omega\left(T_{LCFS,i}-\sum_{j=1}^{i-1}S_j\right)}\right) &= \int_{t=0}^{\infty}\int_{u=0}^{\infty}\sum_{k_i=0}^{\infty} e^{-\lambda_i u}\frac{\left(\lambda_i u\right)^{k_i}}{k_i!}e^{-\omega u} \\
&\times \prod_{j=1}^{i-1} e^{-\lambda_j(1-\beta_j(\omega))(t+u)}\mathbb{E}\left(e^{-\omega R_i}|k_i \text{ arrivals in } C^r\right)\mathrm{d}P\left(C^p\le t; C^r \le u\right).
\end{aligned}$$

Clearly, $\mathbb{E}\left(e^{-\omega R_i}|k_i \text{ arrivals in } C^r\right) = \beta_i^{k_i+1}(\omega)$, the LST of the sum of $k_i+1$ service times. So

$$\begin{aligned}
&\mathbb{E}\left(e^{-\omega\left(T_{LCFS,i}-\sum_{j=1}^{i-1}S_j\right)}\right) = \\
&\quad \beta_i(\omega)\int_{t=0}^{\infty}\int_{u=0}^{\infty}\sum_{k_i=0}^{\infty} e^{-\lambda_i u}\frac{\left(\lambda_i u\beta_i(\omega)\right)^{k_i}}{k_i!}e^{-\omega u} \\
&\times\ e^{-\sum_{j=1}^{i-1}\lambda_j(1-\beta_j(\omega))(t+u)}\mathrm{d}P\left(C^p \le t; C^r \le u\right) \\
&=\ \beta_i(\omega)\int_{t=0}^{\infty}\int_{u=0}^{\infty} e^{-\lambda_i(1-\beta_i(\omega))u}e^{-\omega u} \\
&\times\ e^{-\sum_{j=1}^{i-1}\lambda_j(1-\beta_j(\omega))(t+u)}\mathrm{d}P\left(C^p \le t; C^r \le u\right).
\end{aligned}$$

Using (5.13), we get:

$$\mathbb{E}\left(e^{-\omega\left(T_{LCFS,i}-\sum_{j=1}^{i-1}S_j\right)}\right) = \beta_i(\omega)\frac{\gamma\left(X_i(\omega)\right)-\gamma\left(X_{i+1}(\omega)+\omega\right)}{\left(\lambda_i(1-\beta_i(\omega))+\omega\right)\mathbb{E}C}.$$

The first moment of $T_{LCFS,i}$ is exactly the same as in (5.27):

$$\mathbb{E}\left(T_{LCFS,i}\right) = \mathbb{E}(B_i)+\mathbb{E}\left(R_C\right)\left(2\sum_{j=1}^{i-1}\rho_j+\rho_i+1\right)+\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right).$$

However, the second moment is larger than $\mathbb{E}\left(T_{FCFS,i}^2\right)$ and can be found with Taylor expansions or Lemma 5.1:

$$\begin{aligned}
\mathbb{E}\left(T_{LCFS,i}^2\right) &= \mathbb{E}\left(B_i^2\right)+\mathbb{E}\left(\left(\sum_{j=1}^{i-1}S_j\right)^2\right)+2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right) \\
&+\ \mathbb{E}\left(R_C\right)\left[2\left(\rho_i+1\right)\mathbb{E}\left(B_i\right)+\lambda_i\mathbb{E}\left(B_i^2\right)+2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right)\right] \\
&+\ \mathbb{E}\left(R_C\right)\left[4\sum_{j=1}^{i-1}\rho_j\mathbb{E}\left(B_i\right)+\left(4\sum_{j=1}^{i-1}\rho_j+2\rho_i+2\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\right] \\
&+\ \mathbb{E}\left(R_C^2\right)\left[3\left(\sum_{j=1}^{i-1}\rho_j\right)^2+\left(\rho_i+1\right)^2+3\sum_{j=1}^{i-1}\rho_j\left(\rho_i+1\right)\right]. \qquad (5.29)
\end{aligned}$$

Hence $\mathbb{E}\left(T_{LCFS,i}^2\right)=\mathbb{E}\left(T_{FCFS,i}^2\right)+\rho_i\mathbb{E}\left(R_C^2\right)$. This should not come as a surprise, based on what we have previously seen in the gated section.

### 5.5.3 Random Order of Service

For generally distributed service times and a ROS discipline, we derive the LST of the sojourn time of a random customer. The time between the polling epoch of

$Q_i$ and the departure of a tagged type-$i$ customer ($R_i$) depends on the total number of type-$i$ customers that arrived during $C^*$, say $k_i$. Although we previously used random indicators $U_{i,n}$ to determine how many customers are served before the tagged customer, it is also easily seen that this number is uniformly distributed on $\{0, \ldots, k_i - 1\}$. Therefore, $R_i$ is the sum of $l_i$ service times, with $l_i$ randomly chosen from $\{1, \ldots, k_i\}$.

Because the switch-over times are independent of $C^r$, $R_i$ and the service times of all other customers that arrive during $C^*$, we can focus on just these three parts of the sojourn time of a tagged customer, $T_{ROS,i} - \sum_{j=1}^{i-1} S_j$. Because each of these parts only depends on $C^*$ and/or $C^r$, we condition on the residual cycle length $C^r$ and the preceding cycle length $C^p$ ($C^* = C^p + C^r$):

$$\begin{aligned}\mathbb{E}\left(e^{-\omega\left(T_{ROS,i}-\sum_{j=1}^{i-1} S_j\right)}\right) &= \int_{t=0}^{\infty}\int_{u=0}^{\infty}\sum_{k_i=0}^{\infty} e^{-\lambda_i(t+u)}\frac{\left(\lambda_i(t+u)\right)^{k_i}}{k_i!}e^{-\omega u} \\ \times \prod_{j=1}^{i-1} e^{-\lambda_j(1-\beta_j(\omega))(t+u)}&\mathbb{E}\left(e^{-\omega R_i}|k_i \text{ others}\right) \mathrm{d}P\left(C^p \leq t; C^r \leq u\right).\end{aligned}$$

Using the result in (5.13), we get

$$\begin{aligned}&\mathbb{E}\left(e^{-\omega(T_{ROS,i}-\sum_{j=1}^{i-1} S_j)}\right) = \\ &\quad \frac{\beta_i(\omega)}{\mathbb{E}(C)\left(1-\beta_i(\omega)\right)}\frac{1}{\lambda_i}\int_{X_i(\omega)}^{X_{i+1}(\omega)}\frac{\gamma(y)-\gamma(y+\omega)}{\omega}\mathrm{d}y, \qquad (5.30)\end{aligned}$$

with

$$X_i(\omega) = \sum_{j=1}^{i-1}\lambda_j\left(1-\beta_j(\omega)\right).$$

For the first and second moment of $T_{ROS,i}$, we differentiate (5.30) by using a Taylor series development in $\omega$ and find:

$$\begin{aligned}\mathbb{E}\left(T_{ROS,i}\right) &= \mathbb{E}\left(B_i\right) + \frac{\mathbb{E}\left(C^2\right)}{2\mathbb{E}(C)}\left(2\sum_{j=1}^{i-1}\rho_j + \rho_i + 1\right) \\ &\quad + \sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right). \qquad (5.31)\end{aligned}$$

Indeed, the mean sojourn time consists of the mean service time of the tagged customer, the mean residual cycle time, the mean work arriving at $Q_1, \ldots, Q_{i-1}$ during the past and residual cycle time $\left(2 \times \frac{\mathbb{E}(C^2)}{2\mathbb{E}(C)}\right)$, half of the average work arriving at $Q_i$ during the past and residual cycle time and the mean switch over times $\mathbb{E}\left(S_1\right), \ldots, \mathbb{E}\left(S_{i-1}\right)$. Furthermore, the first moment is again exactly the same as in (5.27).

For the second moment, we find

$$\begin{aligned}
\mathbb{E}\left(T_{ROS,i}^2\right) &= \mathbb{E}\left(B_i^2\right) + \mathbb{E}\left(\left(\sum_{j=1}^{i-1} S_j\right)^2\right) + 2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right) \\
+ \ \mathbb{E}\left(R_C\right) &\left(2\left(\rho_i+1\right)\mathbb{E}\left(B_i\right) + \lambda_i\mathbb{E}\left(B_i^2\right) + 2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right)\right) \\
+ \ \mathbb{E}\left(R_C\right) &\left[4\sum_{j=1}^{i-1}\rho_j\mathbb{E}\left(B_i\right) + \left(4\sum_{j=1}^{i-1}\rho_j + 2\rho_i + 2\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\right] \\
+ \ \mathbb{E}\left(R_C^2\right) &\left[3\left(\sum_{j=1}^{i-1}\rho_j\right)^2 + \rho_i^2 + \frac{3}{2}\rho_i + 1 + 3\sum_{j=1}^{i-1}\rho_j\left(\rho_i+1\right)\right].
\end{aligned}$$

Note that the mean sojourn time of a type-$i$ customer can be larger than the mean sojourn time of a type-$(i+1)$ customer, because $\mathbb{E}\left(T_{ROS,i+1}\right) \leq \mathbb{E}\left(T_{ROS,i}\right)$ if

$$\mathbb{E}\left(B_i\right) \geq \mathbb{E}\left(B_{i+1}\right) + \frac{\mathbb{E}\left(C^2\right)}{2\mathbb{E}(C)}\left[\lambda_{i+1}\mathbb{E}\left(B_{i+1}\right) + \lambda_i\mathbb{E}\left(B_i\right)\right] + \mathbb{E}\left(S_i\right).$$

Furthermore, notice that, as is true in the gated case and in Avi-Itzhak and Halfin [6], the second moments are such that $\mathbb{E}\left(T_{FCFS,i}^2\right) < \mathbb{E}\left(T_{ROS,i}^2\right) < \mathbb{E}\left(T_{LCFS,i}^2\right)$. The differences are again as follows:

$$\mathbb{E}\left(T_{LCFS,i}^2\right) - \mathbb{E}\left(T_{ROS,i}^2\right) = \mathbb{E}\left(T_{ROS,i}^2\right) - \mathbb{E}\left(T_{FCFS,i}^2\right) = \frac{\mathbb{E}\left(R_C^2\right)\rho_i}{2}.$$

### 5.5.4 Processor sharing

The derivation of the LST of the sojourn time in the case of the PS service discipline is different from the one in ROS, because the sojourn time now heavily depends on the required service time of the tagged customer. However, for exponentially distributed service times, the analysis is the same as for ROS, because of Proposition 5.1.

Now suppose that the service times are generally distributed. As the reader would guess, it will again be to our advantage to condition on the amount of service brought to $Q_i$ by a tagged customer during steady-state. If such a customer brings an amount of work $x$ to $Q_i$, then its sojourn time minus $x$ is just

$$\begin{aligned} D_{PS,i}(x) &= C^r + \sum_{j=1}^{i-1} (V_j + S_j) + \sum_{T_{i,m} \in (-C^p, 0)} \min(B_{i,m}, x) \\ &+ \sum_{T_{i,n} \in (0, C^r)} \min(B_{i,n}, x). \end{aligned}$$

Again, because the switch-over times are independent of all other quantities present in our representation of $D_{PS,i}(x)$, we will focus on computing the LST of $D_{PS,i}(x) - \sum_{j=1}^{i-1} S_j$. If we let $\phi_i(\omega, x)$ denote the LST of $\min(B_i, x)$, then

$$\begin{aligned} \mathbb{E}\left(e^{-\omega\left(D_{PS,i}(x) - \sum_{j=1}^{i-1} S_j\right)}\right) &= \int_{t=0}^{\infty} \int_{u=0}^{\infty} e^{-\omega u} e^{-\lambda_i(1-\phi_i(\omega,x))(t+u)} \\ \times \prod_{j=1}^{i-1} & e^{-\lambda_j(1-\beta_j(\omega))(t+u)} \mathrm{d}P(C^p \le t, C^r \le u) \\ &= \frac{\gamma(X_i(\omega) + \lambda_i(1 - \phi_i(\omega, x))) - \gamma(X_i(\omega) + \lambda_i(1 - \phi_i(\omega, x)) + \omega)}{\omega \mathbb{E}[C]}, \end{aligned}$$

where the second equality follows from (5.13).

By applying Lemma 5.1, it follows that

$$\begin{aligned} &\mathbb{E}\left(D_{PS,i}(x) - \sum_{j=1}^{i-1} S_j\right) = \\ &\quad \mathbb{E}(R_C)\left[1 + 2\sum_{j=1}^{i-1} \rho_j + 2\lambda_i \mathbb{E}(\min(B_i, x))\right] \end{aligned} \tag{5.32}$$

and

$$\begin{aligned} &\mathbb{E}\left(\left(D_{PS,i}(x) - \sum_{j=1}^{i-1} S_j\right)^2\right) = \\ &\quad \mathbb{E}(R_C)\left[2\sum_{j=1}^{i-1} \lambda_j \mathbb{E}\left(B_j^2\right) + 2\lambda_i \mathbb{E}\left(\min(B_i, x)^2\right)\right] \\ &+ \mathbb{E}\left(R_C^2\right)\left[1 + 3\sum_{j=1}^{i-1} \rho_j + 3\lambda_i \mathbb{E}(\min(B_i, x))\right. \\ &+ \left. 3\left(\sum_{j=1}^{i-1} \rho_j + \lambda_i \mathbb{E}(\min(B_i, x))\right)^2\right]. \end{aligned} \tag{5.33}$$

Finally, after combining the switch-over times and the service time $x$ of the tagged customer with (5.32) and (5.33), and integrating with respect to $dB_1(x)$, we get

$$\begin{aligned}\mathbb{E}\left(T_{PS,i}\right) &= \mathbb{E}\left(B_i\right)+\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\\ &+ \mathbb{E}\left(R_C\right)\left[1+2\sum_{j=1}^{i-1}\rho_j+2\lambda_i\mathbb{E}\left(B_{i,1:2}\right)\right]\end{aligned} \tag{5.34}$$

and

$$\begin{aligned}&\mathbb{E}\left(T^2_{PS,i}\right)=\mathbb{E}\left(B_i^2\right)+\mathbb{E}\left[\left(\sum_{j=1}^{i-1}S_j\right)^2\right]+2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\\ &+\ \mathbb{E}\left(R_C\right)\left[2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right)+2\lambda_i\mathbb{E}\left(B^2_{i,1:2}\right)\right]\\ &+\ 2\mathbb{E}\left(R_C\right)\left[\mathbb{E}\left(B_i\right)+2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1}\rho_j+2\lambda_i\left[\mathbb{E}\left(B_{i,1}\right)^2+\mathbb{E}\left(B^2_{i,1:2}\right)\right]\right]\\ &+\ 2\mathbb{E}\left(R_C\right)\left[1+2\sum_{j=1}^{i-1}\rho_j+2\lambda_i\mathbb{E}\left(B_{i,1:2}\right)\right]\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right)\\ &+\ \mathbb{E}\left(R_C^2\right)\left[1+3\sum_{j=1}^{i-1}\rho_j+3\lambda_i\mathbb{E}\left(B_{i,1:2}\right)+3\left(\sum_{j=1}^{i-1}\rho_j\right)^2\right.\\ &+\ \left.6\sum_{j=1}^{i-1}\rho_j\lambda_i\mathbb{E}\left(B_{i,1:2}\right)+\lambda_i^2\left(2\mathbb{E}(B_{i,1:3}B_{i,2:3})+\mathbb{E}\left(B^2_{i,1:3}\right)\right)\right].\end{aligned} \tag{5.35}$$

### 5.5.5 Shortest Job First

Now we will compute the first and second moments of the sojourn time under the SJF policy. In this case it is clear that, conditional on the service time of the tagged customer being $x$,

$$\begin{aligned}D_{SJF,i}(x) &= C^r+\sum_{j=1}^{i-1}\left(V_j+S_j\right)+\sum_{T_{i,m}\in(-C^p,0)}B_{i,m}\mathbf{1}\left(B_{i,m}\leq x\right)\\ &+ \sum_{T_{i,m}\in(0,C^r)}B_{i,n}\mathbf{1}\left(B_{i,n}\leq x\right).\end{aligned}$$

If we mimic the above derivation of the LST of the conditional delay for the PS case, we see that

$$\begin{aligned}&\mathbb{E}\left(e^{-\omega\left(D_{SJF,i}(x)-\sum_{j=1}^{i-1}S_j\right)}\right) = \\ &\quad \frac{\gamma\left(X_i(\omega)+\lambda_i\left(1-\phi_i(\omega,x)\right)\right)-\gamma\left(X_i(\omega)+\lambda_i\left(1-\phi_i(\omega,x)\right)+\omega\right)}{\omega\mathbb{E}(C)},\end{aligned}$$

where in this case $\phi_i(\omega, x)$ is the LST of $B_i\mathbf{1}\left(B_i \leq x\right)$.

Just as before, we get

$$\begin{aligned}&\mathbb{E}\left(D_{SJF,i}(x)-\sum_{j=1}^{i-1}S_j\right) = \\ &\quad \mathbb{E}\left(R_C\right)\left[1+2\sum_{j=1}^{i-1}\rho_j+2\lambda_i\mathbb{E}\left(B_i\mathbf{1}\left(B_i\leq x\right)\right)\right] \qquad (5.36)\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}\left(\left(D_{SJF,i}(x)-\sum_{j=1}^{i-1}S_j\right)^2\right) = \\ &\quad \mathbb{E}\left(R_C\right)\left[2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right)+2\lambda_i\mathbb{E}\left(B_i^2\mathbf{1}\left(B_i\leq x\right)\right)\right] \\ &\quad + \mathbb{E}\left(R_C^2\right)\left[1+3\sum_{j=1}^{i-1}\rho_j+3\lambda_i\mathbb{E}\left(B_i\mathbf{1}\left(B_i\leq x\right)\right)\right. \\ &\quad + \left.3\left(\sum_{j=1}^{i-1}\rho_j+\lambda_i\mathbb{E}\left(B_i\mathbf{1}\left(B_i\leq x\right)\right)\right)^2\right]. \qquad (5.37)\end{aligned}$$

Therefore, the first and second moments of the sojourn time are as follows:

$$\begin{aligned}\mathbb{E}\left(T_{SJF,i}\right) &= \mathbb{E}\left(B_i\right)+\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right) \\ &\quad + \mathbb{E}\left(R_C\right)\left[1+2\sum_{j=1}^{i-1}\rho_j+\lambda_i\mathbb{E}\left(B_{i,1:2}\right)\right] \qquad (5.38)\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left(T_{SJF,i}^2\right) &= \mathbb{E}\left(B_i^2\right) + \mathbb{E}\left(\left(\sum_{j=1}^{i-1} S_j\right)^2\right) + 2\mathbb{E}\left(B_i\right)\sum_{j=1}^{i-1} E\left(S_j\right) \\ &+ \ \mathbb{E}\left(R_C\right)\left[2\sum_{j=1}^{i-1}\lambda_j\mathbb{E}\left(B_j^2\right) + \lambda_i\mathbb{E}\left(B_{i,1:2}^2\right)\right] \\ &+ \ \mathbb{E}\left(R_C\right)\left[2\mathbb{E}\left(B_i\right) + 4\sum_{j=1}^{i-1}\rho_j\mathbb{E}\left(B_{i,1}\right) + 2\lambda_i\mathbb{E}\left(B_{i,1}\right)^2\right] \\ &+ \ 2\mathbb{E}\left(R_C\right)\left[1 + 2\sum_{j=1}^{i-1}\rho_j + \lambda_i\mathbb{E}(B_{i,1:2})\right]\sum_{j=1}^{i-1}\mathbb{E}\left(S_j\right) \\ &+ \ \mathbb{E}\left(R_C^2\right)\left[1 + 3\sum_{j=1}^{i-1}\rho_j + \frac{3}{2}\lambda_i\mathbb{E}\left(B_{i,1:2}\right) + 3\left(\sum_{j=1}^{i-1}\rho_j\right)^2\right] \\ &+ \ \mathbb{E}\left(R_C^2\right)\left[3\lambda_i\mathbb{E}\left(B_{i,1:2}\right)\sum_{j=1}^{i-1}\rho_j + \lambda_i^2\mathbb{E}\left(B_{i,1:3}B_{i,2:3}\right)\right]. \qquad (5.39)\end{aligned}$$

## 5.6 Conclusion

The production system is translated to a cyclic polling system. For this polling system, we have obtained the (LST of the) sojourn time distribution in a gated queue, for various service orders within that queue. This sojourn time distribution is equal to the corresponding lead time distribution in the production system. The first two moments of the sojourn time also have been obtained, allowing us to study the impact of the service order.

The following ordering results were already obtained by Winands et al. [91], with respect to the average lead time:

$$\mathbb{E}(T_{SJF}) \leq \mathbb{E}(T_{PS}) \leq \mathbb{E}(T_{FCFS}) = \mathbb{E}(T_{LCFS}) = \mathbb{E}(T_{ROS}),$$

if $2\mathbb{E}(B_{1,1:2}) \leq \mathbb{E}(B_1)$ and

$$\mathbb{E}(T_{SJF}) \leq \mathbb{E}(T_{FCFS}) = \mathbb{E}(T_{LCFS}) = \mathbb{E}(T_{ROS}) \leq \mathbb{E}(T_{PS}),$$

if $2\mathbb{E}(B_{1,1:2}) \geq \mathbb{E}(B_1)$.

For the second moment, the ordering was, just like in Shanthikumar and Sumita [76], as follows:

$$\mathbb{E}\left(T_{LCFS}^2\right) > \mathbb{E}\left(T_{ROS}^2\right) > \mathbb{E}\left(T_{FCFS}^2\right).$$

Further,

$$\mathbb{E}\left(T_{SJF}^2\right) \leq \mathbb{E}\left(T_{FCFS}^2\right)$$

if $\mathbb{E}\left(B_{1,1:2}\right) \leq (2/3)\mathbb{E}(B_{1,1:2})$, i.e. if the service time distribution is DFR (i.e. has a decreasing failure rate). For a Processor Sharing service policy, the ordering is a bit more complicated, but for exponential service times it is easily seen that

$$\mathbb{E}\left(T_{PS}^2\right) = \mathbb{E}\left(T_{ROS}^2\right).$$

One could also investigate whether or not there exists a sort of ordering among the distributions of the sojourn times considered here.

The gated polling discipline turns out to be very tractable, thanks to the fact that the sojourn times of the customers who are being served during a visit are not affected by later arrivals which take place in that visit period. We expect exhaustive service to be more complicated. This is a topic for further research. The case of fixed priorities within a queue of a polling system also receives attention in Boon et al. [11] and Boon [12].

The results obtained for the waiting times in a polling model can now easily be interpreted as distributional results for the lead time in the production system as well. The sojourn time in the queueing system represents the lead time (including the production time) in the production system. For the distribution of the lead time excluding the production time, one just looks at the waiting time distribution in the queueing system.

Chapter 6

# Backlog: A fixed cycle

The results for the first two moments of the waiting times of customers in a polling system with a gated or globally gated visit discipline depend on the parameter settings of all queues, which causes no numerical problems. However, in order to find the average costs for these visit disciplines in a production system, one needs the distributional results of the queue lengths. As was seen in the previous chapter, the Laplace-Stieltjes transform of the waiting time in a certain queue contains an infinite product, because of the branching type structure of the (gated and exhaustive) visit disciplines. The same infinite product also appears in the probability generating function of the queue length, see Resing [71]. This infinite product contains an $N$-dimensional immigration function, see Equation (5.6). In order to find the exact value of this immigration function, a recursive procedure is required that causes numerical problems for large values of $N$. The complexity of the calculation lies in the fact that the processes at the different queues depend on each other, which results in a complex recursive procedure to obtain the exact value of the immigration function.

So the multi-dimensionality of the system causes numerical problems for all branching type base-stock control policies. Fortunately, the analysis of the fixed cycle policy does not encounter these numerical problems, because of Property 1.2. This property tells us that the processes at the different product flows behave independently. This chapter presents the analysis of the fixed cycle policy for a production system with backlog. It was already mentioned in Chapters 1 and 2 that under the fixed cycle strategy, the system can be decomposed into $N$ subsystems. Therefore, this chapter considers a cyclic, single-item production system with which we can analyse each of these subsystems. In each subsystem, a periodic embedded Markov chain is observed in the shortfall level at slot boundaries. This embedded shortfall process is analysed using generating functions. Then, the optimal base-stock level is derived from a newsvendor type relation. The model is also extended to one with time slot dependent base-stock levels. This chapter is based on [21] and [23].

## 6.1 Introduction

We look at the fixed cycle strategy and decompose the system into $N$ subsystems. Each subsystem is analysed as a queueing model and an optimal base-stock level is derived. Then, a local search algorithm is presented to find a close to optimal fixed cycle that can serve as a basis for the one step improvement approach in the next chapter. Güllü et al. [47] also study the fixed cycle production scheme and present two heuristic algorithms to find the lengths of the production periods of the different items. Both algorithms are (partly) based on a deterministic model. Further, the optimal decision level is derived for a single period model. In Erkip et al. [35], a similar model is studied, in which all time slots have the same length and demand distributions are equal for each time slot. A matrix analytic method is used to find the optimal decision level for the infinite horizon model. Van den Broek et al. [83] also look at a fixed cycle control scheme for a queueing system, with as application a traffic light. They present several bounds and approximations for the queue lengths in heavy traffic systems that do not require any numerical procedures. They also assume that vehicles which arrive at an empty queue during an active (or green) period, pass through the system without any delay. Darroch [29] studied this traffic light system with a different arrival distribution if the queue is empty and geometric distributed service times. He gives a generating function for the number of waiting cars at slot boundaries and derives some inequalities for the expected queue length and the expected delay per vehicle.

A more general queueing model with vacations is studied in Fuhrmann and Cooper [43]. They give a decomposition result for the distribution of the number of customers present in a queueing model with vacations which holds under certain conditions. These conditions include: Customers arrive according to a Poisson process, the customers are served in an order that is independent of their service times and the number of customers that arrive during a vacation is independent of the number of customers present just before the start of that vacation. The decomposition consists of the number of customers present in a standard $M|G|1$ queue and the number of customers who arrive during a residual vacation. Unfortunately, the fixed cycle model does not satisfy all necessary conditions, because if the base-stock level is reached during a production period, the system idles during one production time. This idling time is also a vacation and therefore, the number of customers that arrive during a vacation is not independent of the number of customers present in the system when the vacation began. Fortunately, the limiting distribution of the shortfall (which can be seen as the queue length distribution of items that need to be produced) can be found in a more direct way, without using the decomposition result of Fuhrmann and Cooper [43].

In this study, the optimal decision levels for a given fixed cycle are derived from a newsvendor type equation in Subsection 6.3.2 and an approximation is given to address numerical problems. Further, a fixed cycle scheme with time slot dependent decision levels is analysed in Section 6.4 and again a newsvendor type equation is given. Section 6.5 presents an algorithm to find a near-optimal fixed cycle scheme.

A summary is given in Section 6.6.

## 6.2 Cyclic production

The fixed cycle policy for a production system with backlog applies the same rules as the fixed cycle policy for a production system with lost sales, which is already discussed in Chapter 2. For ease of reference, we also give a short description of the fixed cycle policy in this chapter.

The fixed cycle policy reserves a production period of fixed length for every item $i$. This production period consists of a number of $g_i$ production times, each with a length $T_i^P$. The order of production is fixed and the decision to produce or not to produce a product is based on the base-stock level $S(i)$. If the stock level of item $i$ equals this level just before the start of a production slot of type $i$, the system idles during the next slot. Every queue can be seen and analysed as a single-item production system with periodic vacations, which is done in the first part of this chapter. Because the analysis focuses on only one of the $N$ product types, the index $i$ is omitted in the notation.

So, we consider a single-item cyclic production system where each cycle starts with $g$ production times and is concluded by a vacation. This vacation period consists of the reserved production periods for the other items and the total time spent on switching. The total expected time spent on switching in one cycle is denoted by $\sigma$. Production and vacation times are possibly random, but independent. Demand arrives according to a (compound) Poisson process. The system is embedded on the instances corresponding to the start of a production time or the start of a vacation. The time intervals in this chain will be called *slots*, where each cycle consists of $g$ production slots and 1 vacation slot. Whether a production slot is used for production or for idling is read from a base-stock level $S$. If at the start of a production slot the stock level is less than $S$, then the slot is used to produce exactly one item. Given the assumptions about the demand process and the production rule we obtain an embedded periodic (cyclic) Markov chain at slot boundaries.

The state of this chain is described by the number of products short to the level $S$ at the beginning of a slot and the slot number within the cycle. Using this formulation, the limiting behavior of the Markov chain is independent of the value of $S$. Linear cost functions are considered for the number of items on stock and the backlog. Then, as we will show, if the distribution of the number of products short to the base-stock level $S$ is known, an expression for the optimal value $S^*$ can be derived from a newsvendor type equation. This stock-out distribution will be determined via a generating function approach.

To this end, first some notation is introduced. The generating function for the demand in a production slot is denoted by $\mathcal{A}_P(z) = \sum_{k=0}^{\infty} a_P(k)z^k$, with $a_P(k)$ the probability that the demand in a production slot is equal to $k$. Similarly, $\mathcal{A}_V(z)$ and $a_V(k)$ are defined for the vacation slot. The length of the vacation slot is denoted by $T^V$. Further, $\lambda$ denotes the mean demand per time unit and $X$ is defined as the number of products short to the base-stock level $S$. In the following subsection an expression is derived for $G_n(z)$, the generating function of $X$ at slot boundary $n$.

## 6.3 The generating function

Define $X_{n,m}$ as the value of $X$ at slot boundary $n$ in cycle $m$ for $n = 1, \ldots, g+1$. Now consider the limiting random variable

$$X_n = \lim_{m \to \infty} X_{n,m}, \quad n = 1, \ldots, g+1.$$

The distribution of $X_n$ is well-defined if the Markov chain $\{X_{n,m}, m = 1, 2, \ldots\}$ is aperiodic and irreducible (which is immediate from the (compound) Poisson demand assumption) and provided the system is stable, i.e. if the number of arrivals per cycle is less than the available number of production slots, so if $\lambda \left( g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)\right) < g$. We assume this is the case and denote the distribution of $X_n$ by

$$p(k, n) = P(X_n = k), \quad k \geq 0, \ n = 1, \ldots, g+1$$

and the generating function of $X_n$ by

$$\mathcal{G}_n(z) = \sum_{k=0}^{\infty} p(k, n) z^k, \quad n = 1, \ldots, g+1.$$

Let now $D_n$ denote the demand that occurs in time slot $n$. Then, one has

$$\begin{aligned} X_1 &= X_{g+1} + D_{g+1}, \\ X_n &= X_{n-1} + D_{n-1} - I_{\{X_{n-1} > 0\}}, \quad n = 2, \ldots, g+1. \end{aligned}$$

From these equations one gets

$$\begin{aligned} \mathcal{G}_1(z) &= \mathcal{G}_{g+1}(z) \mathcal{A}_V(z), \\ \mathcal{G}_n(z) &= \frac{1}{z} \mathcal{A}_P(z) \left[ \mathcal{G}_{n-1}(z) + p(0, n-1)(z-1) \right], \quad n = 2, \ldots, g+1. \end{aligned}$$

As one easily verifies, this leads by iteration to

$$\mathcal{G}_1(z) = \frac{\sum_{m=1}^{g} \mathcal{A}_P^{g+1-m}(z) \mathcal{A}_V(z)(z^m - z^{m-1}) p(0, m)}{z^g - \mathcal{A}_P^g(z) \mathcal{A}_V(z)}, \tag{6.1}$$

$$\mathcal{G}_n(z) = \left( \frac{\mathcal{A}_P(z)}{z} \right)^{n-1} \mathcal{G}_1(z) + \sum_{m=1}^{n-1} p(0, m)(z-1) \left( \frac{\mathcal{A}_P(z)}{z} \right)^{n-m}, \quad n = 2, \ldots, g+1. \tag{6.2}$$

The generating function of $X_1$ is of indeterminate form, but the $g$ boundary probabilities $p(0, n)$, $n = 1, \ldots, g$, can be determined by considering the zeros of the denominator in (6.1) that lie on or within the unit circle. The following Rouché type lemma is taken from Adan et al. [2] and is specialized to our case.

**Lemma 6.1.** *If the effective load* $\rho_{eff} := \frac{\lambda(g\mathbb{E}(T^P)+\mathbb{E}(T^V))}{g} < 1$ *and* $\mathcal{A}_V(0)\mathcal{A}_P^g(0) \neq 0$*, then* $z^g = \mathcal{A}_P^g(z)\mathcal{A}_V(z)$ *has* $g$ *roots on or within the unit circle.*

Denote the $g$ roots of $z^g = \mathcal{A}_P^g(z)\mathcal{A}_V(z)$ in $|z| \leq 1$ by $z_0 = 1, z_1, \dots, z_{g-1}$. Since the function $\mathcal{G}_1(z)$ is finite on and inside the unit circle, the numerator of the right-hand side of (6.1) needs to be zero for each of the $g$ roots, i.e., the numerator should vanish at the exact points where the denominator of the right-hand side of (6.1) vanishes. Lemma 6.1 and (6.1) together lead to $g$ equations in terms of the $g$ boundary probabilities, from which the latter can be determined. The roots can be determined using methods from Janssen and Van Leeuwaarden [51]. It is assumed that these $g$ roots are all different. If roots with multiplicity greater than one occur, the derivatives (up to the number of multiplicity) of the numerator of (6.1) can be set to zero to obtain sufficiently many equations. For the root $z = 1$, l'Hôpital's rule is applied to obtain one equation from $\mathcal{G}_1(1) = 1$.

### 6.3.1 The limiting distribution

In principle, the probabilities $p(k, n)$ can be found by numerically inverting $\mathcal{G}_n(z)$. However, in this case, the probabilities can be derived directly from the $g$ boundary probabilities.

In order to find all limiting probabilities $p(k, n)$ from the probabilities $p(0, n)$, $n = 1, \dots, g$ (obtained via Lemma 6.1), the balance equations are used:

$$\begin{aligned} p(k,1) &= \sum_{j=0}^{k} p(j, g+1) a_v(k-j), && (6.3) \\ p(k,n) &= \sum_{j=1}^{k+1} p(j, n-1) a_p(k+1-j) + p(0, n-1) a_p(k), \\ & \quad n = 2, \dots, g+1. && (6.4) \end{aligned}$$

Using (6.3) with $k = 0$, the probability $p(0, g+1)$ can be obtained from

$$p(0, g+1) = \frac{1}{a_v(0)} p(0, 1).$$

Next let us rewrite equation (6.4) for $n = 2, \dots, g+1$ as

$$\begin{aligned} p(k+1, n-1) = \frac{1}{a_p(0)} \Bigg( & p(k,n) - \sum_{j=1}^{k} p(j, n-1) a_p(k+1-j) \\ & - p(0, n-1) a_p(k) \Bigg). \end{aligned} \quad (6.5)$$

Then, starting with $k = 0$, we first find the probabilities $p(1, n)$, $n = 2, \dots, g+1$. The probability $p(1, 1)$ can then be obtained from equation (6.3). Continuing in this way, one recursively gets the probabilities $p(k, n)$, $k \geq 2$.

### 6.3.2 The optimal base-stock level

In this section the optimal base-stock level will be determined for the case of linear holding and backlogging costs. These costs will be computed from the expected number of products on stock or backlogged at slot boundaries.

In the original, multi-item production model, the vacation period consists of set-up times and production times of other items. So the length of this period is typically much longer than the length of a production slot. The costs are only calculated at slot boundaries, while in the vacation period the stock or backlog level is more variable than during a production slot. Therefore, the vacation period is divided into a number of vacation slots, say $g_V$, such that the slots are small enough to get a good approximation for the expected costs per time unit.

If the length of the vacation period is stochastic, it is not necessarily possible or evident how to divide the vacation period into a number of slots. However, the vacation period consists of a number of production times for other product types and set-up times, so the most natural choice would be to choose the slots corresponding to these production and set-up slots. Further, this structure of the vacation period guarantees us that the division of the vacation period is always possible.

The length of vacation slot $n$ equals $T_n$. Because the demand distribution is assumed to be (compound) Poisson, the stock-out distribution at the new slot boundaries can easily be found, using the stock-out distribution at slot boundary $g$ (which was already found in the previous section).

The expected stock-out at slot boundary $g+n$, $n = 2, \ldots, g_V$ is just $\mathbb{E}(X_{g+1}) + \lambda \sum_{m=g+1}^{g+n} T_m$. The p.g.f. of $X_{g+n}$ equals $\mathcal{G}_g(z) \prod_{m=g+1}^{g+n} \mathcal{A}_{g+n}(z)$, with $\mathcal{A}_{g+n}(z)$ the p.g.f. of the arriving demand in slot $g+n$. Now define the following linear cost function, with weights based on the average slot duration:

$$\begin{aligned} c(S) &= \sum_{n=1}^{g} \frac{\mathbb{E}\left(T^P\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} \left(c_I \mathbb{E}_S(I_n) + c_B \mathbb{E}_S(B_n)\right) \\ &\quad + \sum_{n=g+1}^{g+g_V} \frac{\mathbb{E}\left(T_n\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} \left(c_I \mathbb{E}_S(I_n) + c_B \mathbb{E}_S(B_n)\right), \end{aligned} \tag{6.6}$$

where $I_n$ is the number of items on stock and $B_n$ the backlog at slot boundary $n$. Further, $\mathbb{E}_S(I_n)$ and $\mathbb{E}_S(B_n)$ denote the expected stock and backlog level if base-stock level $S$ is used. Because the cost function is a weighted sum of costs at different time slots, we also look at the corresponding weighted limiting distribution:

$$p(k) = \sum_{n=1}^{g} \frac{\mathbb{E}\left(T^P\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} p(k,n) + \sum_{n=g+1}^{g+g_V} \frac{\mathbb{E}\left(T_n\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} p(k,n) \ .$$

The optimal base-stock level $S^*$ for this ‘newsvendor problem’ (see for example

Porteus [68]) is now readily obtained as:

$$S^* = \min \left\{ S \left| \sum_{k=0}^{S} p(k) > \frac{c_B}{c_I + c_B} \right. \right\}. \tag{6.7}$$

### 6.3.3 A geometric tail approximation

The probabilities $p(k)$ in (6.7) can be found using the recursive method from Subsection 6.3.1. However, we have experienced numerical problems with this procedure for large values of $k$. If the load on the system is high and $S$ gets larger than 20, the recursive method gives numerically unstable results. Therefore we propose to use the following approximation for $p(k)$ if $S^*$ gets large.

Van Eenige [85] and Van Mieghem [86] encounter the same numerical problems and use an approximation from Tijms and Van de Coevering [82] for the tail probabilities that is based on the following asymptotic behavior

$$\lim_{k\to\infty} \frac{p(k)}{p(k+1)} = \gamma,$$

with $\gamma$ the unique root of $z^g - \mathcal{A}_P^g(z)\mathcal{A}_V(z)$ in $(1, \infty)$. This root can easily be computed with bisection.

Let us use the direct computation of $p_k$ up to $K$ and the tail approximation for $k > K$. (The choice of $K$ can be made during the direct computation. If either the geometric behavior seems to have started or one seems to lose the numerical stability, one switches to the geometric tail behavior.)

So, we use

$$P(X = k) \approx \kappa\gamma^{-k}, k = K + 1, \ldots, \tag{6.8}$$

where $\kappa$ is the normalization constant which can be expressed in terms of $P(X \leq K)$:

$$\kappa = \left(1 - \gamma^{-1}\right)\left(1 - P(X \leq K)\right)\gamma^{K+1}.$$

Upon substituting (6.8) into (6.7), and assuming that $S^* > K$, so that the tail approximation is accurate, one gets the following approximative value for $S^*$:

$$\tilde{S} = \left\lceil \frac{-\ln(c_I) + \ln(c_I + c_B) + \ln(\kappa) - \ln(\gamma - 1)}{\ln(\gamma)} \right\rceil, \tag{6.9}$$

with $\lceil x \rceil$ the smallest integer that is greater than or equal to $x$.

In order to see whether (6.9) results in a good approximation, we give some numerical results comparing the approximation with the exact method.

### 6.3.4 Numerical results

For various parameters settings, for which we can determine the exact value of $S^*$ numerically, the results for $S^*$ and $\tilde{S}$ are presented in Tables 6.1, 6.2 and 6.3.

$c_I = 1, c_B = 10, g = 5, T^P = 1, T^V = 5$

| $\rho_{eff}$ | $\mathbb{E}I$ | $\mathbb{E}B$ | $S^*$ | costs | $\tilde{S}$ | costs |
|---|---|---|---|---|---|---|
| 0.50 | 1.26 | 0.12 | 2 | 2.50 | 2.12 | 2.58 |
| 0.60 | 1.92 | 0.11 | 3 | 2.98 | 2.65 | 2.98 |
| 0.70 | 2.48 | 0.14 | 4 | 3.89 | 3.59 | 3.89 |
| 0.80 | 3.65 | 0.21 | 6 | 5.78 | 5.54 | 5.78 |
| 0.90 | 7.32 | 0.44 | 12 | 11.68 | 11.48 | 11.68 |
| 0.95 | 14.74 | 0.89 | 24 | 23.62 | 23.45 | 23.62 |

**Table 6.1:** The values of $\mathbb{E}I$, $\mathbb{E}B$, $S^*$ and $\tilde{S}$. $\lambda = \frac{1}{2}\rho_{eff}$.

$c_I = 1, c_B = 10, g = 10, T^P = 1, T^V = 10$

| $\rho_{eff}$ | $\mathbb{E}I$ | $\mathbb{E}B$ | $S^*$ | costs | $\tilde{S}$ | costs |
|---|---|---|---|---|---|---|
| 0.50 | 1.89 | 0.14 | 3 | 3.30 | 3.69 | 3.37 |
| 0.60 | 2.48 | 0.13 | 4 | 3.81 | 4.07 | 4.00 |
| 0.70 | 2.96 | 0.17 | 5 | 4.62 | 4.85 | 4.62 |
| 0.80 | 4.04 | 0.22 | 7 | 6.27 | 6.63 | 6.27 |
| 0.90 | 7.62 | 0.43 | 13 | 11.91 | 12.40 | 11.91 |
| 0.95 | 15.00 | 0.88 | 25 | 23.71 | 24.29 | 23.71 |

**Table 6.2:** The values of $\mathbb{E}I$, $\mathbb{E}B$, $S^*$ and $\tilde{S}$. $\lambda = \frac{1}{2}\rho_{eff}$.

These results are based on a fixed cycle scheme with deterministic time slots of unit length and a Poisson demand process. The values of $\tilde{S}$ from (6.9) are given without taking the 'ceiling' to show the real difference with the value of $S^*$.

One sees that the approximation $\tilde{S}$ is correct for nearly all parameter settings, except for $\rho_{eff} = 0.5$ in the first table and $\rho_{eff} = 0.5, 0.6$ in the second table. For these systems, the minimum value $K$ for which the approximation in Equation (6.8) is accurate is relatively high, because the systems are lightly loaded. Apparently, the optimal base-stock level is below this value $K$.

For higher values of $\rho_{eff}$, the approximation is equal to $S^*$, which is just what we want, because the numerical problems occur if $\rho_{eff}$ is high. Further, it is observed

$c_I = 1, c_B = 10, g = 3, T^P = 1, T^V = 9$

| $\rho_{eff}$ | $\mathbb{E}I$ | $\mathbb{E}B$ | $S^*$ | costs | $\tilde{S}$ | costs |
|---|---|---|---|---|---|---|
| 0.50 | 1.31 | 0.10 | 2 | 2.29 | 1.86 | 2.29 |
| 0.60 | 2.00 | 0.09 | 3 | 2.85 | 2.42 | 2.85 |
| 0.70 | 2.58 | 0.12 | 4 | 3.79 | 3.37 | 3.79 |
| 0.80 | 3.78 | 0.19 | 6 | 5.71 | 5.33 | 5.71 |
| 0.90 | 7.46 | 0.42 | 12 | 11.66 | 11.29 | 11.66 |
| 0.95 | 14.89 | 0.87 | 24 | 23.57 | 23.26 | 23.57 |

**Table 6.3:** The values of $\mathbb{E}I$, $\mathbb{E}B$, $S^*$ and $\tilde{S}$. $\lambda = \frac{1}{4}\rho_{eff}$.

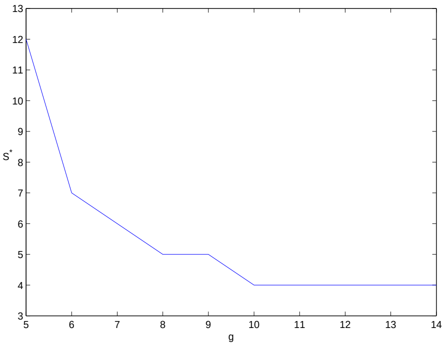


**Figure 6.1:** The optimal decision level $S^*$ decreases as $g$ increases $T^P = 1, T^V = 4, \lambda = 0.5, c_I = 1, c_B = 10$.

that the approximation of $S^*$ is less accurate if $S^*$ is low, because it is based on (6.8), which is only an approximation for the tail probabilities. But for low values of $S^*$, there are no numerical problems, so this is (again) not a problem.

Figure 6.1 gives the effect of $g$ on the value of $S^*$ and shows that $S^*$ increases if $g$ decreases. This is explained by the fact that the effective utilization, $\frac{\lambda T_{FC}}{g}$, increases if $g$ decreases.

It is also seen that $S^*$ decreases very slowly for large values of $g$. The length of the production period may become so large that the base-stock level is just the stock level that one would like to have to cover the vacation period. On the other hand, if $g$ is even larger and the vacation period is (relatively) so small that the production period completely dominates the stock process, $S^*$ converges to the optimal base-stock level for a production system without vacation periods.

Another point is that if $g$ is somewhat longer, one needs the safety stock for the vacation period only at the end of the production period. This suggests that a cost reduction can be obtained with a base-stock level that is lower at the beginning of a production period and increases towards the end of the production period.

## 6.4 Time slot dependent base-stock levels

So, let us now consider the system in which the base-stock levels are time slot dependent. Denote the different base-stock levels by $S_1, \ldots, S_g$, with $S_n$ the base-stock level for time slot $n$, see Figure 6.2. In the sequel, particularly the analysis in Subsection 6.4.1, we will often use the following two assumptions:

**Assumption 6.1.**

$$S_n \ \leq \ S_{n+1}, \ n = 1, \ldots, g.$$

**Assumption 6.2.**

$$S_{n+1} \ \leq \ S_n + 1, \ n = 1, \ldots, g.$$

For most realistic settings, these assumptions will hold, but it might be possible to construct counterexamples for this based on the following intuition.

Assumption 6.1 might be violated in the following situation. If the production times are highly variable and the non-production slots are not, then one might need a higher safety stock at the beginning of the production period than near the end of it.

With respect to Assumption 6.2 the following could occur. At the end of the production period one wants a higher base-stock level because of the coming long vacation period. However, because of the holding costs, one does not want to invest in this higher stock in the production slots before the final one. On the other hand, if less than expected demand arrived during the last cycle and there are still $S_{g-1}+1$ products on stock at the start of the $g-th$ production slot, then one might be willing to produce one more product, which would mean $S_g > S_{g-1} + 1$.

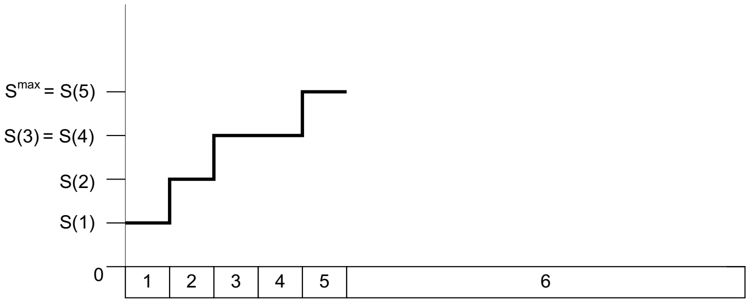


**Figure 6.2:** The base-stock levels of 5 production slots.

In order to find the optimal values of $S_1, \ldots, S_g$, we slightly adapt the model description from Section 6.2. With different base-stock levels (and compound Poisson demand) in every slot the stock level can reach the maximum of $S_1, \ldots, S_g$, so in some slot(s) $n$ the actual stock can be larger than the base-stock level $S_n$.

This is shown in Figure 6.3, where the stock level just before the first production slot in the second cycle is higher than $S_1$.

Therefore, define $S^{\max}$ as $\max\{S_1, \ldots, S_g\}$ and let $\tilde{X_n}$ denote the number of products short compared to $S^{\max}$ at slot boundary $n$.

As before, if $\rho_{eff} < 1$ the limiting distribution of $\tilde{X_n}$ exists and it will be denoted by

$$\tilde{p}(k,n) = \lim_{t\to\infty} P(\tilde{X}_{n,t} = k), \quad n = 1, \ldots, g+1, \quad k \geq 0,$$

with generating function

$$\tilde{\mathcal{G}}_n(z) = \sum_{k=0}^{\infty} \tilde{p}(k,n) z^k, \quad n = 1, \ldots, g+1.$$

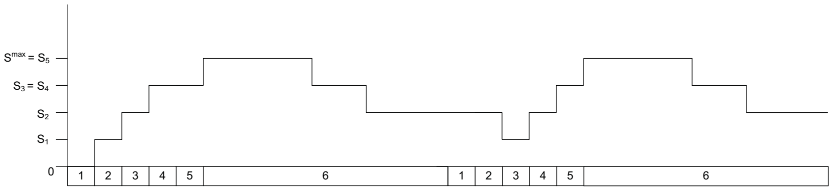


**Figure 6.3:** The inventory level during two cycles.

Define

$$\delta_n := S^{\max} - S_n, \quad n = 1, \ldots, g.$$

In the same way as in Subsection 6.3, we get

$$\tilde{\mathcal{G}}_1(z) = \frac{\sum_{m=1}^{g} \sum_{k=0}^{\delta_m} \tilde{p}(k,m)(z^{k+m} - z^{k+m-1}) \mathcal{A}_P^{g+1-m}(z) \mathcal{A}_V(z)}{z^g - \mathcal{A}_P^g(z) \mathcal{A}_V(z)}, \tag{6.10}$$

$$\tilde{\mathcal{G}}_n(z) = \tilde{\mathcal{G}}_1(z) \left( \frac{\mathcal{A}_P(z)}{z} \right)^{n-1} + \sum_{m=1}^{n-1} \sum_{k=0}^{\delta_m} \tilde{p}(k,m)(z^k - z^{k-1}) \left( \frac{\mathcal{A}_P(z)}{z} \right)^{n-m}, \quad n = 2, \ldots, g+1, \tag{6.11}$$

The expressions for $\tilde{\mathcal{G}}_1(z), \ldots, \tilde{\mathcal{G}}_{g+1}(z)$, however, still contain the unknown boundary probabilities $\tilde{p}(k,n),\ k = 0, \ldots, \delta_n,\ n = 1, \ldots, g$. Lemma 6.1 gives $g$ equations. Since there are more than $g$ unknowns, we will have to construct a larger set of balance equations for these boundary probabilities. A similar problem is discussed in Denteneer et al. [33]. In the next subsection we will follow the approach used there to find these boundary probabilities.

### 6.4.1 The boundary probabilities

The boundary probabilities we are looking for only concern probabilities from the production period. For ease of notation we combine the last production slot and the vacation period into one production slot.

With $a_g^*(k),\ k \geq 0$, denoting the distribution of the total demand in time slots

$g$ and $g+1$ together, the set of balance equations becomes:

$$\begin{aligned}
\tilde{p}(k,n) &= \sum_{m=\delta_{n-1}+1}^{k+1} \tilde{p}(m,n-1)a_p(k+1-m) \\
&+ \sum_{m=0}^{\delta_{n-1}} \tilde{p}(m,n-1)a_p(k-m), \\
&\quad 2 \leq n \leq g, \quad k \geq \delta_{n-1}, && (6.12) \\
\tilde{p}(k,n) &= \sum_{m=0}^{k} \tilde{p}(m,n-1)a_p(k-m), \\
&\quad 2 \leq n \leq g, \quad 0 \leq k < \delta_{n-1}, && (6.13) \\
\tilde{p}(k,1) &= \sum_{m=\delta_g+1}^{k+1} \tilde{p}(m,g)a_g^*(k+1-m) \\
&+ \sum_{m=0}^{\delta_g} \tilde{p}(m,g)a_g^*(k-m), \quad k \geq \delta_g, && (6.14) \\
\tilde{p}(k,1) &= \sum_{m=0}^{k} \tilde{p}(m,g)a_g^*(k-m), \quad 0 \leq k < \delta_g. && (6.15)
\end{aligned}$$

Under the assumption that $S_n \leq S_{n-1}+1$ for all $n$ (see Assumption 6.2), it would be enough to look at the equations described by (6.13), (6.15) and the equations from Lemma 6.1. However, if this assumption does not hold, then for one or more equations described by (6.13) and (6.15) the probability on the left hand side does not appear in the expression for $G_1(z)$. Therefore, below an algorithm is given that results in a set of balance equations which, combined with the equations from Lemma 6.1 gives us the boundary probabilities that appear in (6.1).

The algorithm below uses two sets: a set of unknown probabilities (variables), $U$, and a set of equations, $E$, from which the unknowns have to be obtained. Initially we define $U = \{\tilde{p}(k,n); k = 0,\ldots,\delta_n,\ n = 1,\ldots,g\}$ and we let $E$ contain the $g$ equations from Lemma 6.1 plus the equations described by (6.13) and (6.15). The balance equation with left-hand side $\tilde{p}(k,n)$ will be labeled with $(k,n)$. So at the start $U$ contains the variables $\tilde{p}(k,n), k = 0,\ldots,\delta_n$, and $E$ the equations $(k,n),\ k = 0,\ldots,\delta_{n-1}-1$. Then the number of equations in $E$ (including the ones from Lemma 6.1) and the number of unknowns in $U$ are both equal to $\sum_n \delta_n + g$. However, not all the probabilities that appear in the equations in $E$ are in $U$. For each of these probabilities, an equation will be added to $E$.

We start with the slot *just after* the one with the lowest decision level, thus the largest $\delta_n$. Let $n$ be the current production slot. For each $(k,n) \in E$ for which $\tilde{p}(k,n)$ is not yet in $U$ the variable $\tilde{p}(k,n)$ is added to $U$.

Next, for each of these variables an extra balance equation is added to $E$, namely equation $(k-1,n+1)$ (where $g+1$ is to be read as 1). All probabilities appearing at the right-hand side of this new equation are already in $U$ and at most one extra unknown probability appears at the left-hand side.

Then we move to the next slot, $n+1$. Again each variable that appears in $E$ but is not in $U$ is added to $U$, and in the same way as before, for a new variable $\tilde{p}(k,n)$ equation $(k-1,n+1)$ is added to $E$. Continue until all slots have been considered. In the last step (step $g$), the slot with the highest value of $\delta_n$ is reached. Therefore, the probabilities on the left-hand side of all equations added in the previous step are already in $U$, because $k$ can not exceed $\max\{\delta_1,\ldots,\delta_g\}$. This means that the construction ends with $|E| = |U|$ and the variables in $U$ being the only ones appearing in $E$.

Assuming that all equations from the roots, from l'Hôpital's rule, and obtained via this algorithm are linearly independent, the unknowns in $U$ can be found.

### 6.4.2 Optimal value for $S^{\max}$

Denote the number of products short compared to $S^{\max}$ at 'weighted' random slot boundaries by $\tilde{X}$. The generating function of $\tilde{X}$ is defined as

$$\tilde{\mathcal{G}}(z) = \sum_{k=0}^{\infty} \tilde{p}(k) z^k.$$

So

$$\tilde{\mathcal{G}}(z) = \sum_{n=1}^{g} \frac{\mathbb{E}\left(T^P\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} \mathcal{G}_n(z) + \sum_{n=g+1}^{g+g_V} \frac{\mathbb{E}\left(T_n\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} \mathcal{G}_n(z),$$

with

$$\tilde{p}(k) = \sum_{n=1}^{g} \frac{\mathbb{E}\left(T^P\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} p(k,n) + \sum_{n=g+1}^{g+g_V} \frac{\mathbb{E}\left(T_n\right)}{g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)} p(k,n), \quad k \geq 0.$$

The limiting distributions of $\tilde{X}, \tilde{X}_1, \ldots, \tilde{X}_g$ can be found by inverting $\tilde{\mathcal{G}}(z)$ and $\tilde{\mathcal{G}}_n(z)$, $n = 1, \ldots, g$. The distribution of $\tilde{X}$ depends on $\delta_1, \ldots, \delta_g$, but not on $S^{\max}$. Therefore, a newsvendor type equation can be given for the optimal value of $S^{\max}$. For a given vector $(\delta_1, \ldots, \delta_g)$, the optimal value of $S^{\max}$ is given by

$$S^{\max *} = \min \left\{ S^{\max} \,\middle|\, \sum_{k=0}^{S^{\max}} \tilde{p}(k) > \frac{c_B}{c_I + c_B} \right\}. \tag{6.16}$$

We emphasize that the distribution of $\tilde{X}$ depends on all elements in the vector $(\delta_1, \ldots, \delta_g)$ and thus $S^{\max *}$ does as well. Furthermore, there is no expression for the optimal value of every individual $S_n$, $n = 1, \ldots, g$.

In order to also find the expected costs $c(S_1, \ldots, S_g)$ for a given vector $(S_1, \ldots, S_g)$, one can write

$$\begin{aligned} c(S_1,\dots,S_g) &= c_I \sum_{k=0}^{S^{\max}} \tilde{p}(k)(S^{\max}-k) + c_B \sum_{k=S^{\max}}^{\infty} \tilde{p}(k)(k-S^{\max}) \\ &= (c_I+c_B) \sum_{k=0}^{S^{\max}} \tilde{p}(k)(S^{\max}-k) + c_B(\mathbb{E}\tilde{X} - S^{\max}), \qquad (6.17) \end{aligned}$$

where the weights $\frac{\mathbb{E}(T^P)}{g\mathbb{E}(T^P)+\mathbb{E}(T^V)}$ and $\frac{\mathbb{E}(T_n)}{g\mathbb{E}(T^P)+\mathbb{E}(T^V)}$ are now hidden in $\tilde{p}(k)$ and $\mathbb{E}\tilde{X}$.

The finite sum $\sum_{k=0}^{S^{\max}} \tilde{p}(k)(S^{\max}-k)$ is obtained from the equilibrium probabilities. For $\mathbb{E}\tilde{X}$ we use $\mathbb{E}(\tilde{X}_1),\dots,\mathbb{E}(\tilde{X}_{g+1})$. For the derivation of $\mathbb{E}(\tilde{X}_1)$, we refer to Appendix 6.A. The expressions for $\mathbb{E}(\tilde{X}_n),\ n=2,\dots,g+1$ are then obtained with

$$\tilde{X}_n = \tilde{X}_{n-1} + D_{n-1} - I_{\tilde{X}_{n-1}>\delta_{n-1})}, \quad n=2,\dots,g+1,$$

with $D_{n-1}$ the number of arrivals in time slot $n-1$ and $I_{\tilde{X}_{n-1}>\delta_{n-1}}$ the production indicator of time slot $n-1$. The result is:

$$\begin{aligned} \mathbb{E}(\tilde{X}_1) &= \frac{1}{g-\lambda\left(g\mathbb{E}\left(T^P\right)+\mathbb{E}\left(T^V\right)\right)} \\ &\times \left( \sum_{m=1}^{g}\sum_{k=0}^{\delta_m} \tilde{p}(k,m)\left[\left(g\mathbb{E}\left(T^P\right)+\mathbb{E}\left(T^V\right)\right)\lambda + k + m - 1\right] \right. \\ &- \left. \frac{1}{2}\ \left[g(g-1) - \left(g\mathbb{E}\left(T^P\right)+\mathbb{E}\left(T^V\right)\right)^2\lambda^2\right] \right), \\ \mathbb{E}(\tilde{X}_n) &= \mathbb{E}(\tilde{X}_1) + \lambda\left(g\mathbb{E}\left(T^P\right)+\mathbb{E}\left(T^V\right)\right) - \sum_{m=1}^{n-1}\sum_{k=0}^{\delta_m}\tilde{p}(k,m), \quad n=2,\dots,g+1. \end{aligned}$$

### 6.4.3 Numerical results

Some numerical results are presented to compare the expected costs per time unit for fixed cycles with a constant decision level with the costs for fixed cycles with time slot dependent decision levels. The lengths of the set-up and production slots in this section are again deterministic and of unit length.

Recall that given the vector $(\delta_1,\dots,\delta_g)$ the distribution of $\tilde{X}$ does not depend on $S^{\max}$ and that the corresponding $S^{\max *}$ is given by (6.16).

In order to limit the number of possible vectors $\delta$, Assumptions 6.1 and 6.2 are used. The number of different values for $\delta$ then equals $2^{g-1}$. In the numerical results below, the presented optimal values $S_1,\dots,S_g$ are the optimal ones given these two restrictions.

Tables 6.4 and 6.5 show numerical results for $T^V=5$, while in Tables 6.6 and 6.7, $T^V=25$. In Tables 6.4 and 6.6 we have the results for $c_I=1, c_B=10$, while in

**Table 6.4**

$c_I = 1, c_B = 10, \lambda = \frac{1}{2}\rho_{eff}, g = 5, T^P = 1, T^V = 5$

| $\rho_{eff}$ | $S^*$ | costs | $[S_1, S_2, S_3, S_4, S_5]$ | costs | cost reduction in % |
|---|---|---|---|---|---|
| 0.75 | 5 | 4.642 | [4 4 5 5 5] | 4.608 | 0.716 |
| 0.8 | 6 | 5.782 | [5 5 6 6 6] | 5.762 | 0.334 |
| 0.85 | 8 | 7.731 | [7 7 8 8 8] | 7.716 | 0.195 |
| 0.9 | 12 | 11.682 | [11 11 12 12 12] | 11.672 | 0.089 |
| 0.95 | 24 | 23.630 | [23 23 24 24 24] | 23.625 | 0.023 |

**Table 6.5**

$c_I = 1, c_B = 20, \lambda = \frac{1}{2}\rho_{eff}, g = 5, T^P = 1, T^V = 5$

| $\rho_{eff}$ | $S^*$ | costs | $[S_1, S_2, S_3, S_4, S_5]$ | costs | cost reduction in % |
|---|---|---|---|---|---|
| 0.75 | 6 | 5.807 | [4 5 6 6 6] | 5.699 | 1.869 |
| 0.8 | 8 | 7.262 | [7 7 8 8 8] | 7.239 | 0.324 |
| 0.85 | 10 | 9.794 | [9 9 10 10 10] | 9.769 | 0.248 |
| 0.9 | 15 | 14.749 | [15 15 15 15 15] | 14.749 | 0.000 |
| 0.95 | 30 | 30.000 | [29 30 30 30 30] | 29.963 | 0.110 |

Tables 6.5 and 6.7 $c_B = 20$. For almost every value of $\rho_{eff} < 0.95$, Tables 6.6 and 6.7 show a larger cost reduction than Tables 6.4 and 6.5. Apparently, the length of the vacation period has a positive effect on the attainable cost reduction, while the value of $\rho_{eff}$ and the fraction $\frac{c_I}{c_I+c_B}$ have a negative effect on it. The first observation can be explained by the fact that the average demand that arrives in a production slot decreases if $T^V$ increases. So there is relatively more time to get a high stock level at the end of the production period. Therefore, the decision levels at the beginning of the production period can be lower. The second observation is easy to explain: If the load on the system increases, the system should use its full capacity to get $S^{\max *}$ products on stock. The last observation is explained by the fact that if the backlogging costs are relatively high, one wants to prevent the system to create backlog. Therefore, the decision levels are all close to $S^{\max *}$.

It is seen that the last decision level can be higher than the optimal constant decision level. This is a good example of the effect of multiple decision levels: By increasing the last decision level, one saves possible backlogging costs during the vacation period, but the lower decision levels at the beginning of the production period save the holding costs that are incurred with a constant decision level. The optimal values of $\delta$ are more difficult to find than $S^{\max *}$. The main problem here is that the number of possible vectors is too large. Therefore, time slot dependent base-stock levels will not be discussed in more detail.

**Table 6.6**

$c_I = 1, c_B = 10, \lambda = \frac{1}{6}\rho_{eff}, g = 5, T^P = 1, T^V = 25$

| $\rho_{eff}$ | $S^*$ | costs | $[S_1, S_2, S_3, S_4, S_5]$ | costs | cost reduction in % |
|---|---|---|---|---|---|
| 0.75 | 5 | 4.984 | [3 4 5 5 6] | 4.933 | 1.024 |
| 0.8 | 7 | 6.039 | [4 5 6 7 7] | 6.006 | 0.551 |
| 0.85 | 9 | 7.931 | [6 7 8 8 9] | 7.902 | 0.365 |
| 0.9 | 12 | 11.816 | [10 11 11 12 13] | 11.798 | 0.150 |
| 0.95 | 24 | 23.692 | [22 23 23 24 24] | 23.688 | 0.015 |

**Table 6.7**

$c_I = 1, c_B = 20, \lambda = \frac{1}{6}\rho_{eff}, g = 5, T^P = 1, T^V = 25$

| $\rho_{eff}$ | $S^*$ | costs | $[S_1, S_2, S_3, S_4, S_5]$ | costs | cost reduction in % |
|---|---|---|---|---|---|
| 0.75 | 7 | 6.118 | [4 5 6 6 7] | 5.976 | 2.329 |
| 0.8 | 8 | 7.448 | [6 6 7 7 8] | 7.425 | 0.309 |
| 0.85 | 11 | 9.961 | [9 9 10 11 11] | 9.894 | 0.670 |
| 0.9 | 16 | 14.946 | [14 14 15 16 16] | 14.856 | 0.604 |
| 0.95 | 31 | 30.029 | [29 30 30 31 31] | 29.987 | 0.140 |

## 6.5 The production periods

The (close to) optimal lengths of the production periods can be determined with the local search algorithm presented in Subsection 6.5.2. Although this algorithm can deal with both constant and time slot dependent base-stock levels, we will focus on constant base-stock levels. (As we have seen, the search for an optimal vector of base-stock levels is time consuming and the difference in costs with the constant base-stock level is limited.)

In the remainder of the chapter, a fixed cycle will be described by the vector $g = (g_1, \ldots, g_N)$, a vector with the lengths of the production periods. With $T_i^P$ denoting the average length of a production slot of item $i$, the average duration of the cycle, denoted as $T_{FC}(g)$, satisfies $T_{FC}(g) = \sum_i g_i T_i^P + \sigma$. Such a cycle is stable if for every item the number of production slots in it suffices, i.e, if $\lambda_i T_{FC}(g) < g_i$ for all $i$.

### 6.5.1 The shortest stable fixed cycle

The local search algorithm presented in Subsection 6.5.2 starts with a stable cycle of minimum length.

**Lemma 6.2.** *If $\rho = \sum_{i=1}^N \lambda_i T_i^P < 1$, there exists a unique stable fixed cycle of minimum length.*

A proof of this lemma is found in Appendix 6.B.

This cycle will be referred to as the shortest stable fixed cycle and will be denoted by $g^{min} = (g_1^{\min}, \dots, g_N^{\min})$, with $T_{FC}^{min} = \sum_{i=1}^{N} g_i^{min} T_i^P + \sigma$ the length of this cycle. The next algorithm produces this shortest fixed cycle.

**Algorithm 6.1.**

*Step* 1: *Set $n = 0$ and $g^{(0)} = (1, 1, \dots, 1)$ (or alternatively $g_i^{(0)} = \lambda_i \frac{\sigma}{1-\rho}$ for all $i$, see Appendix 6.C), then $g^{(0)} \leq g^{min}$.*

*Step* 2: *Compute $T_{FC}(g^{(n)}) = \sum_{i=1}^{N} g_i^{(n)} T_i^P + \sigma$. If the system is stable, i.e., $\lambda_i T_{FC}(g^{(n)}) < g_i^{(n)}$ for all $i$, then the minimal fixed cycle has been found: $g^{min} = g^{(n)}$. Otherwise go to Step 3.*

*Step* 3: *Compute: $g_i^{(n+1)} = \left\lfloor \lambda_i \left( \sum_{i=1}^{N} g_i^{(n)} T_i^P + \sigma \right) + 1 \right\rfloor$, with $\lfloor x \rfloor$ the largest integer less than or equal to $x$. Set $n = n + 1$ and go back to Step 2.*

**Lemma 6.3.** *Algorithm 6.1 gives the shortest stable fixed cycle for any polling system with $N$ queues.*

A proof of this lemma is found in Appendix 6.C.

### 6.5.2 A local search algorithm for a good fixed cycle

In order to find a good fixed cycle, we will start with $g^{min}$ and apply a local search algorithm, which works as follows. In each cycle improvement step we lengthen the cycle for one of the product types, the one for which lengthening given the largest reduction in costs. However, there will be two complicating factors. First of all, lengthening the cycle for one type might result in an unstable system for one or more of the other types; this will be solved by lengthening the production period for those product types as well. And second, it is possible that a longer production period for only one product gives an increase in costs, whereas a longer production period for two or more products gives a decrease in costs. This will be taken care of using a special termination criterion; stop only if for a number of improvement steps no improvement has been found. This number is chosen equal to $N$, so that for each product type simultaneously a production slot can be added.

**Algorithm 6.2.**

*Step* 1: *Start with the shortest stable fixed cycle that can be obtained with Algorithm 6.1 and define $g^{(1,0)} = g^{min}$.*

*Step* 2: *In improvement step $n$, starting with the cycle $g^{(n,0)}$ try lengthening the production period of every product type. For item $i$, add a production slot for this type obtaining the cycle $g^{(n,0)} + e_i$, make this cycle stable, denote this stable cycle by $g^{(n,i)}$ and calculate the expected costs $c(g^{(n,i)})$. Determine $i^*$ such that $c(g^{(n,i^*)}) = \min_i c(g^{(n,i)})$ and define $g^{(n+1,0)} = g^{(n,i^*)}$. Set $n := n + 1$.*

*Step* 3: *If in the last $N$ steps no improvement has been found, i.e., if* $c(g^{(n-N,0)}) \leq c(g^{(n-l,0)})$ *for* $l = 0, \dots N-1$, *then terminate.*
*The best cycle found is* $g^{(n-N,0)}$. *Otherwise, return to Step* 2.

## 6.6 Conclusion

A fixed cycle policy is analysed for a multi-item production system. The structure of this policy allows for a decomposition of the system into $N$ independent periodic subsystems, one for each product type. Then an analysis is performed per product type and the optimal base-stock level is found for a given fixed cycle. The analysis is extended to allow for slot dependent base-stock levels. The optimal base-stock levels are obtained from newsvendor type expressions.

A local search algorithm is presented that produces a (close to) optimal fixed cycle policy. This strategy will later be used to construct a good dynamic strategy by means of a single policy improvement step. Because of the decomposition of the system in independent subsystems the size of the problem in terms of product types only plays a minor role.

# Appendix

## 6.A Expectation in the first slot

We obtain the mean value of $\tilde{X}_1$ by taking the first derivative of the generating function $\tilde{\mathcal{G}}_1(z)$. To keep the notation simple, we rewrite $\tilde{\mathcal{G}}_1(z)$ as $\frac{\mathcal{N}(z)}{\mathcal{D}(z)}$, with

$$\mathcal{N}(z) = \sum_{m=1}^{g} \sum_{k=0}^{\delta_m} \tilde{p}(k,m)(z^{k+m} - z^{k+m-1})\mathcal{A}_P^{g+1-m}(z)\mathcal{A}_V(z)$$

and

$$\mathcal{D}(z) = z^g - \mathcal{A}^*(z).$$

Here, $\mathcal{A}^*(z) = \mathcal{A}_P^g(z)\mathcal{A}_V(z)$, the generating function of the total demand during one cycle.

$\tilde{\mathcal{G}}_1'(1)$ can now be rewritten as

$$\frac{\mathcal{N}'(z)\mathcal{D}(z) - \mathcal{D}'(z)\mathcal{N}(z)|_{z=1}}{\mathcal{D}^2(z)|_{z=1}} = \frac{\mathcal{N}'(z) - \mathcal{D}'(z)\tilde{\mathcal{G}}_1(z)|_{z=1}}{\mathcal{D}(z)|_{z=1}}.$$

Since $\tilde{\mathcal{G}}_1(1) = \frac{\mathcal{N}'(1)}{\mathcal{D}'(1)}$ by l'Hôpital and $\mathcal{D}(1) = 0$, we can use l'Hôpital again:

$$\tilde{\mathcal{G}}_1'(1) = \frac{\mathcal{N}''(z) - \mathcal{D}''(z)\tilde{\mathcal{G}}_1(z) - \mathcal{D}'(z)\tilde{\mathcal{G}}_1'(z)|_{z=1}}{\mathcal{D}'(z)|_{z=1}}$$

Using $\tilde{\mathcal{G}}_1(1) = 1$ and rearranging terms gives us:

$$\tilde{\mathcal{G}}_1'(1) = \frac{\mathcal{N}''(1) - \mathcal{D}''(1)}{2\mathcal{D}'(1)},$$

with

$$\begin{aligned}
\mathcal{D}'(1) &= g - \left(\lambda(g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)\right), \\
\mathcal{N}''(1) &= 2\sum_{m=1}^{g} \sum_{k=0}^{\delta_m} \tilde{p}(k,m)\left[\left(g\mathbb{E}\left(T^P\right) + \mathbb{E}\left(T^V\right)\right)\lambda + k + m - 1\right], \\
\mathcal{D}''(1) &= \left[g(g-1) - \mathcal{A}^{*''}(1)\right].
\end{aligned}$$

## 6.B Proof of Lemma 6.2

**If $\rho = \sum_{i=1}^{N} \lambda_i T_i^P < 1$, there exists a unique stable fixed cycle of minimum length.**

*Proof:* The proof is twofold. First it is shown – with an example – that a stable fixed cycle *exists*. Then the uniqueness of the shortest stable fixed cycle is shown by contradiction.
In order to construct a stable fixed cycle, consider the following system of linear equations:

$$g_i = \lambda_i T_{FC},\ i = 1, \ldots, N, \tag{6.18}$$

$$T_{FC} = \sum_{i=1}^{N} g_i T_i^P + \sigma. \tag{6.19}$$

The solution of this system is unique and (by substitution of the first equation into the second) easily seen to be $T_{FC} = \frac{\sigma}{1-\rho}$, $g_i = \lambda_i \frac{\sigma}{1-\rho}$, $i = 1, \ldots, N$. Based on this solution, a stable fixed cycle is now constructed. For the constructed fixed cycle, it should hold that $g_i < \lambda_i T_{FC}$. The total set-up time is equal for all cycles, because the production order is the same. Therefore, we have to increase the lengths of the production periods, so that the fraction of time spent on switching goes down.

Denote the solution of Equations (6.18) and (6.19) by $x = \{x_1, \ldots, x_N\}$ and look at the fixed cycle described by $\hat{g}(K) = (\lceil Kx_1 \rceil, \ldots, \lceil Kx_N \rceil)$, with $K$ an integer and $\hat{T}_{FC}(K) = \sum_{i=1}^{N} \lceil Kx_i \rceil T_i^P + \sigma$ the cycle length. Then, the number of production times for item $i$ satisfies

$$\hat{g}_i(K) = \lceil Kx_i \rceil \geq Kx_i = K\lambda_i \left( \sigma + \sum_{j=1}^{N} x_j T_j^P \right),$$

while the total cycle time is

$$\hat{T}_{FC}(K) = \sigma + \sum_{j=1}^{N} \lceil Kx_j \rceil T_j^P \leq \sigma + \sum_{j=1}^{N} (Kx_j + 1) T_j^P.$$

So the difference between $\hat{g}_i(K)$ and $\lambda_i \hat{T}_{FC}(K)$ is at least

$$\lambda_i \left( (K-1)\sigma - \sum_{j=1}^{N} T_j^P \right).$$

Now let $K$ satisfy $(K-1)\sigma > \sum_{i=1}^{N} T_i^P$. Then $\hat{g}_i(K) - \lambda_i \hat{T}_{FC}(K) > 0$ for all $i$. Hence $\hat{g}(K)$ describes a stable fixed cycle.

In order to prove the uniqueness, assume that there are two different stable fixed cycles of minimum length, described by $g^{(1)}$ and $g^{(2)}$, and as both represent minimal cycles, the lengths of these cycles, say $T_{FC}^{(1)}$ and $T_{FC}^{(2)}$, are equal. Notice that the cycle lengths $T_{FC}^{(1)}$ and $T_{FC}^{(2)}$ do not denote the number of slots in the cycles, but the actual length of the cycles.

Now construct a new cycle by taking the minimum of the two: $g_i^{min} = \min\{g_i^{(1)}, g_i^{(2)}\}$, for $i = 1 \ldots N$. Since the cycles were different, the new cycle must be shorter, i.e., $T_{FC}^{min} < T_{FC}^{(1)} = T_{FC}^{(2)}$. But for all $i$ the number of production slots $g_i^{min}$ for type $i$ is already sufficient in a longer cycle (with duration $T_{FC}^{(1)}$ or $T_{FC}^{(2)}$), so the cycle $g_i^m$ is stable as well. But this is a contradiction, because it was assumed that $g^{(1)}$ and $g^{(2)}$ describe a stable fixed cycle of minimum length. Therefore, the shortest stable fixed cycle is unique.

## 6.C Proof of Lemma 6.3

**Algorithm 6.1 produces the shortest stable fixed cycle for a polling system with $N$ queues.**

*Proof:* The proof is based on induction.

Assume that for $n \geq 0, g^{(n)} \leq g^{min}$, thus $T_{FC}^{(n)} \leq T_{FC}^{min}$. Then we have for all $i$ that

$$g_i^{(n+1)} = \lfloor \lambda_i T_{FC}^{(n)} + 1 \rfloor \leq \lfloor \lambda_i T_{FC}^{min} + 1 \rfloor = g_i^{min} \ ,$$

so $g^{(n+1)} \leq g^{min}$.
As long as $g^{(n)}$ is not stable $g^{(n+1)}$ will have at least one production slot more than $g^{(n)}$ and then $T_{FC}^{n+1} > T_{FC}^{n}$. However, if $g^{(n)}$ is stable then one must have $g^{(n)} = g^{(n+1)} = g^{min}$.

It now remains to prove that $g^{(0)} \leq g^{min}$.
If one chooses $g_i^{(0)} = 1$ for all $i$, then obviously (assuming $\lambda_i > 0$) one has $g_i^{(0)} \leq g_i^{min}$, since for any stable cycle one has $g_i \geq 1$. This completes the proof: starting with $g^{(0)} = (1, \ldots, 1)$ the cycles $g^{(n)}$ monotonically increase until the minimal cycle has been found.

One may speed up the algorithm a little by choosing a different vector for $g^{(0)}$. In order to see this, assume that $T$ is the duration of some stable fixed cycle. Then the corresponding $g_i$ must satisfy $g_i > \lambda_i T$ for all $i$. Thus $\sum_i g_i T_i^P > \sum_i \lambda_i T T_i^P$, or $T - \sigma > \rho T$, so $T > \frac{\sigma}{1-\rho}$. Now define $g_i^{(0*)} = \left\lfloor \lambda_i \frac{\sigma}{1-\rho} \right\rfloor$ for all $i$. One easily sees that $g_i^{(0*)} \leq g_i^{min}$ (since $T_{FC}^{(0*)} < T_{FC}^{min}$), and $g_i^{(1)} > g_i^{(0*)}$ for the $g^{(1)}$ constructed from $g^{(0*)}$ along the lines of the algorithm.

Chapter 7

# Backlog: One step improvement

This chapter is the continuation of the previous chapter on a fixed cycle control in a multi-item production system. This fixed cycle control policy serves as a basis for the one step improvement approach. Using this approach, a dynamic policy is constructed. Numerical results are presented that indicate that the performance is quite good compared to other policies, such as exhaustive base-stock control. This chapter is based on [24].

## 7.1 Introduction

As was mentioned before, the multi-item production system that we consider, has a resemblance with an intersection controlled by traffic lights. At these intersections, one very often sees a fixed cyclic scheme in which (combinations of) traffic flows receive green, not only in a fixed order but also during a fixed time. In case the intersection is lightly loaded, for instance late in the evening, a dynamic control resembling FCFS is found. For heavily loaded intersections, the dynamic control of a traffic light by a two-step approach is studied in Haijema and van der Wal [48]. In the first step an (in some sense) good completely fixed cyclic control scheme is constructed and in the second step a policy is constructed that takes decisions on producing, idling or switching based on an evaluation of the 'relative urgencies' of the different traffic streams in the fixed cycle scheme. This approach is also used in Chapter 2 for the multi-item production system with lost sales. In this chapter, the approach is used for the multi-item production system with backlog.

For the construction of a good fixed cycle policy, we use the results from Chapter 6. This fixed cycle policy is used as a basis for the one step improvement approach. The chapter is structured as follows. First, we start with a good stable fixed cycle policy, found with Algorithm 6.2 from the previous chapter. This fixed cycle policy is not necessarily optimal. But, just as for the lost sales model, we obtain results that show that the performance of the fixed cycle policy is not too important for the performance of the one step improvement policy that is based on that fixed cycle policy.

The fixed cycle scheme is used as a basis for the one step improvement approach in Section 7.3. This approach and its resulting dynamic policy are discussed in Sections 7.3 and 7.4. In Section 7.5 a conclusion and some suggestions for further research are given.

## 7.2 Model and notation

The objective function we consider is the linear cost function $c(\cdot)$ in Equation (6.6) in the previous chapter. The demand process of every item $i$ is (compound) Poisson with parameter (mean) $\lambda_i$. It is assumed that the total load of the system, $\rho := \sum_{i=1}^{N} \lambda_i T_i^P < 1$. Next we compute a good, stable, fixed cycle according to Algorithm 6.2, see Chapter 6, with (constant) order-up-to levels $S(i),\ i = 1, \ldots, N$. With a fixed cycle policy the system becomes a combination of $N$ independent queues and the relative value function is decomposed into $N$ individual relative value functions.

## 7.3 One step improvement approach

In this section we show how the fixed cycle policy can serve as a basis for the one step improvement approach. To execute the improvement step, the relative values (or bias terms) have to be known. The calculation of these values is usually too complex, because of the multi-dimensionality of the system. However, for the fixed cycle policy these relative values can be computed per product type, as the system simplifies to $N$ independent product flows. For each product type one just has a one-dimensional periodic Markov chain. The number of states is still infinite, but this problem can be solved by introducing (large) maximal stock-out levels $M_i,\ i = 1, \ldots, N$. (These maximal stock-out levels $M_i$ are chosen such that this value is hardly ever reached. The relative values for larger shortfall levels can then be approximated, for instance by extrapolation.) Each of the $N$ periodic chains can then be analysed numerically using successive approximations to compute the $n-$period costs for type $i$ given initial state $k_i$, with $k_i$ the shortfall.

In the improvement step the decision one looks for is 'what to do next', but this can be seen as looking for the best slot to continue with within the fixed cycle, assuming that after this slot the fixed cycle strategy is followed again.

Therefore, the relative values are calculated for every time slot within the fixed cycle. For time slot $n$, the relative value for state $(k_1, \ldots, k_N$ is denoted by $r(n, k_1, \ldots, k_N)$ and, as mentioned in Chapter 1, it is just the sum of the $N$ relative values, one for each item:

$$r(n, k_1, \ldots, k_N) = \sum_{i=1}^{N} r_i(n, k_i). \tag{7.1}$$

In order to compute $r_i(n, k_i)$, we need to be aware of the fact that the fixed cycle strategy is periodic. Also note that these relative values $r_i(n, k_i)$ only have to tell the difference in expected costs between starting in one slot and starting in another slot.

### Determination of the relative values

In the calculation of the relative values, it is assumed that arriving demand that finds a shortfall of $M_i$ is lost. The $n-$period costs $v_n$ and relative value vector $r$ for a periodic Markov chain can be found with Equations (2.9) and (2.10) from Section 2.5.

For product type $i$, denote the state of the system as $(n, k_i)$, with $n$ the slot within the fixed cycle and $k_i$ the number of items short compared to $S(i)$. For the periodic fixed cycle strategy, we get the following (approximate) relative value for item $i$ and state $(n, k_i)$:

$$\hat{r}_i(n, k_i) = \frac{1}{\sum_{j=1}^{C} T_j} \sum_{j=mC+1}^{(m+1)C} T_{j-mC} \; v_{i,j}(n, k_i),$$

with $T_l$ the (average) length of slot $l$ and $m$ sufficiently large. The formula is exactly the same as in Equation (2.11), but the value of $k_i$ is now the shortfall value instead of the number of items on stock. The overall relative value $r(n, k_1, \ldots, k_N)$ for time slot $n$ and state $(k_1, \ldots, k_N)$ is approximated by the sum of the approximate relative values for the $N$ products and pairs $(n, k_j)$, $j = 1, \ldots, N$, see Equation (7.1).

### Base-stock levels

The base-stock levels that are used in the fixed cycle policy are optimal for that policy. Although the relative values are calculated with these base-stock levels, one can set new base-stock levels based on the equilibrium distribution of the shortfall levels in the one step improvement policy. If the relative values are linked to the shortfall levels, they remain the same for the new set of base-stock levels. So decisions depend on the shortfall levels of the different items. Therefore, changing the base-stock levels would not change the processes of the shortfall levels. New base-stock levels can be found with the newsvendor type result in Equation (6.7). The limiting shortfall distribution can be estimated with a simulation study after which the newsvendor type result in Equation (6.7) is applied to set the base-stock levels for the one step improvement policy.

### Good fixed cycle

The fixed cycle obtained from Algorithm 6.2 is not necessarily optimal. But just as in Chapter 2, results show that better fixed cycles do not necessarily result in better one step improvement policies.

In Table 7.1, an example is shown for which two fixed cycle strategies result in a different performance of the one step improvement policy. For fixed cycles that are almost unstable, i.e. the load for one or more items is above 0.96, the calculation

Optimal and adjusted production periods

| $\lambda$ | $g$ | $c_{FC}$ | $c_{1SI}$ |
|---|---|---|---|
| $(0.2, 0.2, 0.2, 0.2)$ | $(12, 12, 12, 12)$ | 61.19 | 30.38 |
| $(0.2, 0.2, 0.2, 0.2)$ | $(14, 14, 14, 14)$ | 59.18 | 32.67 |

$c_{i,I} = 1, c_{i,B} = 50, i = 1, \dots, N$

**Table 7.1:** A 4-item production system with Poisson demand

of the relative values becomes time consuming. The reason for this is the following. For the calculation of the relative values, one has to determine the $n-$period costs $v_n$, with $n$ so large that $v_{n+C} - v_n$ equals the average costs per cycle. If the load on the system is high, $v_{n+C} - v_n$ converges very slowly to the average costs per cycle.

Especially for large shortfall levels, the relative values are not accurate if $n$ is not sufficiently large. This problem can be solved by approximating the relative values, for instance by extrapolation. This method is also used for the calculation of the relative values for shortfall levels that exceed the maximum shortfall level. However, it is already seen that the performance of the fixed cycle is not too important for the performance of the one step improvement policy. So to avoid time consuming calculations of relative values, we restrict the load in the fixed cycle policy to be at most 0.96. This only requires a small adjustment in Algorithm 6.1; the condition $\lambda_i T_{FC}(g^{(n)}) < g_i^{(n)}$ for all $i$ becomes $\lambda_i T_{FC}(g^{(n)}) < 0.96 g_i^{(n)}$ for all $i$. Another adjustment must be made in Step 2 of Algorithm 6.2. Here, 'make this cycle stable' is changed into 'lengthen the production periods until the load for every product type is at most 0.96'.

## 7.4 Results

In this section, we give an overview of the results of the fixed cycle policy and the one step improvement approach and compare them with the results for a number of other policies. For a policy $\Gamma$, $c_\Gamma$ will denote the expected costs per time unit. The fixed cycle policy (FC), gated base-stock policy (G), exhaustive base-stock policy (EXH), adjusted exhaustive base-stock policy (EXH*) and one step improvement policy (1SI) are compared, just as in Chapter 2. Because the optimal base-stock levels for the gated base-stock policy are relatively easy to determine in a production system with backlog (compared to the lost sales system), the gated base-stock policy (G) is also analysed. However, in all examples studied here, the gated base-stock policy performs worse than both the exhaustive and the adjusted exhaustive base-stock policy.

In the examples studied in this section, the holding costs per time unit are all equal to 1 and backlogging costs equal 50. Furthermore, all production and set-up times are deterministic and equal to 1. The arrival processes at the different stock points are either Poisson with intensity $\lambda$ or compound Poisson. In the case of a compound Poisson arrival process, batches with customers arrive with intensity

$\lambda/2$ and each batch is of size 4 with probability 1/3 and of size 1 with probability 2/3. The average batch size is then equal to 2, so the average number of arriving customers per time unit equals $\lambda$.

The one step improvement policy is compared with a gated, exhaustive and an adjusted exhaustive base-stock policy. The exhaustive and gated base-stock policies set base-stock levels for all items and produce the different items in a fixed, cyclic order. The server switches if the base-stock level of the item currently set up is reached or if the shortfall seen upon arrival is produced, otherwise it produces another unit of this type. In this way, the server never idles and decisions only depend on the stock level of the item currently set-up. The adjusted exhaustive base-stock policy applies the same rules as the exhaustive base-stock policy, but skips an item if, just before the switch to this item, the stock level of this type is equal to the base-stock level of this type. If all stock levels equal their base-stock levels, the server switches to the next item.

Just as in the lost sales model, the lengths of the production periods are dependent if a gated or an (adjusted) exhaustive base-stock policy is used. But unlike the lost sales model, in the production model with backlog, the expected shortfall at the start of a production period increases – in expectation – linearly in the length of the period in which the machine is away. So the expected length of the production period also grows linearly in the length of the period in which the machine is away. Therefore, the dependence between the lengths of the different production periods is stronger than in the production system with lost sales, where the shortfall levels are at most equal to the base-stock levels.

The gated, exhaustive and adjusted exhaustive base-stock policies are analysed as follows. First, all base-stock levels are set to zero, then a simulation is performed to find the limiting distribution of the shortfall levels. These distributions are not influenced by any of the $N$ base-stock levels, so the newsvendor equation can be used to obtain the optimal base-stock levels. It is seen that in all examples the (adjusted) exhaustive base-stock policy outperforms the gated base-stock policy.

In many realistic settings for multi-item production systems, about 80 percent of the demand is for only 20 percent of the product types. This is also the case in the paper of Winands et al. [95], where 12.5% of the product types is responsible for 67% of the demand. Therefore, the settings for Tables 7.2 and 7.3 are similar: A low percentage (20%) of the product types is responsible for a high percentage $(80 - 90\%)$ of the demand.

| $\lambda$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| $(0.56, 0.03, 0.05, 0.02, 0.04)$ | 65.79 | 21.85 | 17.79 | 15.93 | 15.16 |
| $(0.60, 0.02, 0.04, 0.01, 0.03)$ | 29.69 | 29.48 | 16.08 | 13.90 | 15.36 |
| $(0.63, 0.02, 0.015, 0.025, 0.01)$ | 43.24 | 20.22 | 14.85 | 12.72 | 12.76 |

**Table 7.2:** Poisson demand

Table 7.2 shows results for a 5-item production system with Poisson arrivals,

| $\lambda$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| $(0.56, 0.03, 0.05, 0.02, 0.04)$ | 85.24 | 45.53 | 41.16 | 38.18 | 35.92 |
| $(0.60, 0.02, 0.04, 0.01, 0.03)$ | 84.39 | 45.64 | 38.68 | 36.26 | 33.51 |
| $(0.63, 0.02, 0.015, 0.025, 0.01)$ | 79.26 | 44.92 | 36.53 | 34.51 | 32.07 |

**Table 7.3:** Compound Poisson demand

where the first item is responsible for $80 - 90\%$ of the total demand. The total load on the system is 0.7, which is the sum of the arrival intensities of the demand processes. Table 7.3 shows results for a 5-item production system with compound Poisson arrivals, all other settings are the same as in Table 7.2. It is seen that in 4 out of 6 cases 1SI performs better than EXH*. Particularly if the demand variation is larger (Table 7.3) 1SI clearly outperforms EXH*. For the systems with Poisson demand, it is seen that 1SI always outperforms EXH. However, in the EXH* policy the machine often skips items, because the demand rates of items 2 up to 5 are relatively low. This leads to a large cost reduction, compared to the EXH policy. In 2 out of 3 examples, the EXH* policy outperforms the 1SI policy. The two examples in which EXH* outperforms 1SI are the systems with the lowest demand rates for items 2 up to 5.

We want to note that it is very important to adjust the order-up-to levels when moving from the fixed cycle system to the dynamic system: the base-stock levels are considerably decreased and the difference in costs is huge (around 70 percent).

### Number of product types

Tables 7.4 and 7.5 show results for completely symmetric production systems with Poisson and compound Poisson distributed demand, respectively. The load in the system is equal to 0.7, with $N = 5, 6, 7$ so the average number of arriving customers is $0.7/N$ per time unit for each product flow. Although these settings are not realistic, the results are illustrative for the performance of the one step improvement policy. It is seen that for compound Poisson demand, in all cases 1SI outperforms EXH*. Moreover, the performance of 1SI is better if $N$ is larger.

| $N$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 5 | 47.51 | 25.69 | 24.10 | 23.84 | 28.26 |
| 6 | 58.03 | 29.84 | 28.27 | 27.98 | 32.32 |
| 7 | 70.04 | 33.78 | 32.45 | 32.14 | 35.67 |

**Table 7.4:** Poisson demand, $\lambda = (0.7/N, \dots, 0.7/N)$

| $N$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 5 | 128.85 | 53.62 | 52.02 | 50.00 | 47.73 |
| 6 | 161.36 | 61.66 | 59.34 | 57.08 | 53.88 |
| 7 | 199.78 | 69.02 | 67.00 | 63.37 | 59.75 |

**Table 7.5:** Compound Poisson demand, $\lambda = (0.7/N, \ldots, 0.7/N)$

### Production order

Just as in Chapter 2, the relative value function of the fixed cycle policy depends on the order of production. This effect is illustrated in Figure 7.1 for an empty 3-item production system with Poisson demand processes and demand intensities $\lambda = (0.25, 0.15, 0.10)$ and $\lambda = (0.25, 0.10, 0.15)$. Holding costs are all equal to 1 and backlogging costs are 50 for all items. The production periods of the fixed cycle are equal to $g = (6, 4, 3)$ and the base-stock levels are $S = (6, 5, 4)$. Because the relative value function is different for different production orders, the decisions in the 1SI policy can also be different for different production orders. This leads to a different performance as well.

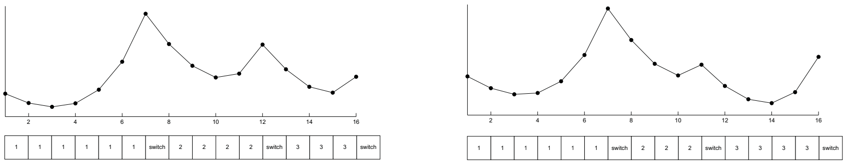


**Figure 7.1:** Relative value functions for two empty 3-item production systems with $\lambda = (0.25, 0.15, 0.10)$ and $\lambda = (0.25, 0.10, 0.15)$.

For 6-item production systems, the effect of the production order is shown in Tables 7.6 up to 7.9. Results are shown for production systems with Poisson demand in Tables 7.6 and 7.7 and for production systems with compound Poisson demand in Tables 7.8 and 7.9. The order of production does have an effect on the performance of the one step improvement policy, but it seems quite random which of the two studied production orders is best. So just as in Chapter 2, it is difficult to give a good intuition for the best order of production.

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 57.01 | 29.30 | 28.23 | 27.20 | 30.74 |
| 0.75 | 76.74 | 34.57 | 33.23 | 32.74 | 32.92 |
| 0.80 | 123.37 | 42.49 | 40.69 | 40.49 | 38.78 |

**Table 7.6:** Poisson demand, $\lambda = \rho(0.25, 0.15, 0.10, 0.25, 0.15, 0.10), b = 50$

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 57.01 | 29.30 | 28.23 | 27.21 | 30.90 |
| 0.75 | 76.74 | 34.58 | 33.25 | 32.73 | 32.96 |
| 0.80 | 123.37 | 42.51 | 40.65 | 40.47 | 38.96 |

**Table 7.7:** Poisson demand, $\lambda = \rho(0.25, 0.25, 0.15, 0.15, 0.10, 0.10), b = 50$

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 159.74 | 61.11 | 59.37 | 55.90 | 52.61 |
| 0.75 | 224.64 | 70.91 | 68.81 | 66.38 | 62.29 |
| 0.80 | 441.63 | 84.86 | 83.28 | 81.43 | 79.61 |

**Table 7.8:** Compound Poisson demand, $\lambda = \rho(0.25, 0.15, 0.10, 0.25, 0.15, 0.10), b = 50$

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 159.74 | 61.12 | 59.38 | 55.89 | 52.54 |
| 0.75 | 224.64 | 70.87 | 68.79 | 66.33 | 62.09 |
| 0.80 | 441.63 | 84.84 | 83.18 | 81.44 | 78.76 |

**Table 7.9:** Compound Poisson demand, $\lambda = \rho(0.25, 0.25, 0.15, 0.15, 0.10, 0.10), b = 50$

### Load, backlogging costs and stochasticity

Although we were not able to get a good intuition for a good production order in the fixed cycle policy, the obtained results in Tables 7.6 up to 7.9 give us a lot of information on the performance of the one step improvement policy. The results show that the one step improvement policy performs better if the load on the system is higher, which was also seen from the results in Tables 7.4 and 7.5.

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 65.83 | 33.50 | 32.03 | 31.12 | 34.27 |
| 0.75 | 88.35 | 39.28 | 37.96 | 37.55 | 36.63 |
| 0.80 | 162.82 | 48.12 | 46.41 | 46.24 | 44.04 |

**Table 7.10:** Poisson demand, $\lambda = \rho(0.25, 0.15, 0.10, 0.25, 0.15, 0.10), b = 100$

| $\rho$ | $c_{FC}$ | $c_G$ | $c_{EXH}$ | $c_{EXH*}$ | $c_{1SI}$ |
|---|---|---|---|---|---|
| 0.70 | 187.15 | 70.93 | 68.99 | 65.66 | 59.73 |
| 0.75 | 266.42 | 82.10 | 80.49 | 77.85 | 70.92 |
| 0.80 | 665.25 | 98.61 | 97.28 | 95.56 | 91.36 |

**Table 7.11:** Compound Poisson demand, $\lambda = \rho(0.25, 0.15, 0.10, 0.25, 0.15, 0.10), b = 100$

Tables 7.10 and 7.11 show results for 6-item production systems with backlogging costs of 100 for all items. All other parameter settings are the same as in Tables 7.6 and 7.8 and it is seen that increasing the backlogging costs has a positive effect on the performance of $1SI$. The fact that the new strategy is able to react to sudden changes in demand also leads to good results, because the one step improvement policy performs better if demand is more stochastic, as in Tables 7.8, 7.9 and 7.11.

## 7.5 Conclusion

A multi-item production system is analysed in which demand is backlogged if it can not be satisfied from stock. For every type, holding and backlogging costs are considered and in order to minimize the total expected costs per time unit, we analysed a one step improvement policy. This one step improvement policy is constructed by starting with a good fixed cycle control and then performing one policy iteration of Howard's policy iteration algorithm [50]. After this policy iteration, the limiting distributions of the shortfall levels are estimated with a simulation study and the base-stock levels are adjusted by applying the newsvendor type result in Equation (6.7). The expected costs are then calculated with the estimated shortfall distributions.

Numerical and simulation results are given to compare the fixed cycle policy, the gated and exhaustive base-stock control policy and the new one step improvement policy. An adjusted exhaustive base-stock control policy is constructed by changing the switching rule in the exhaustive base-stock control policy.

It is shown that the one-step improvement approach leads to a very good dynamic control of the production system, particularly for systems with a large number of product types, systems with a high load, systems with high backlogging costs and systems with stochastic demand. These factors make it difficult to control the production system and apparently, in these systems the one step improvement policy is a good production strategy.

Chapter 8

# Conclusions and further research

## 8.1 Results

### Construction of a dynamic production strategy

In this thesis, a multi-item production system is considered in which $N$ product types share the capacity of a single machine. Two different models are studied, in which we distinguish between backlog and lost sales. First, a production system with lost sales is studied in which the objective is the minimization of the holding and penalty costs. In the production system with backlog, the objective is the minimization of the holding and backlogging costs.

It is seen that the number of possible states in multi-item production systems grows exponentially in the number of product types. Therefore, an MDP approach to obtain the optimal production strategy quickly becomes intractable if the number of product types gets (too) large. Because of this curse of dimensionality, the construction of a dynamic production strategy is often also too complex. However, for both the backlog and the lost sales production system, a dynamic policy is constructed using a one step improvement approach. In this approach, one policy iteration is performed on a fixed cycle policy. The fixed cycle policy is chosen as a basis policy, because it allows for a decomposition of the different product flows. In Chapter 3, the same approach is used for the construction of a dynamic production strategy in a multi-item production system with two machines and lost sales.

### Fixed cycle policy

In a fixed cycle policy, each product flow experiences a periodic production model with production periods and vacation periods. The lengths of these periods are independent of the demand processes. Therefore, each product flow is analysed individually and the total relative value function for the fixed cycle policy is the sum of $N$ individual relative value functions.

In order to find a good fixed cycle, a local search algorithm is presented. The performance of the fixed cycle policy is analysed with successive approximations (for lost sales) or a generating function approach (for backlog). The fact that the resulting fixed cycle is not necessarily optimal is not very important for the performance of the one step improvement policy based on this fixed cycle scheme. We have seen at the beginning of Section 2.6 that better fixed cycles do not always lead to better one step improvement policies. The reason for this probably lies in the fact that the base-stock levels for the fixed cycle policy are too high for the dynamic production strategy. In the dynamic production strategy, the system is able to react to changes in demand and therefore, the stock levels are much more stable than in the fixed cycle policy. So the shortfall distribution changes, which has a direct effect on the optimal base-stock levels. For the backlog model, this direct effect is immediate from the newsvendor result in Equation (6.7).

The shortfall distributions in the fixed cycle policy in the backlog model are independent from the base-stock levels. Because of this property, the newsvendor type equation of Equation (6.7) is used to determine the optimal base-stock levels. In the lost sales model, this equation can not be used, because the base-stock levels determine the shortfall distributions. This is easily seen, after observing that the probability of having no stock is always greater than zero. This is exactly the same as having a shortfall level that equals the base-stock level, which has a probability of zero for policies with a lower base-stock level. However, it is possible to find good base-stock levels with a local search algorithm. Assuming that the cost function is convex in the base-stock levels, these base-stock levels are optimal.

Under the fixed cycle policy, each product flow is analysed as a periodic production model. In the backlog model, an extension of this periodic production model is studied in which the base-stock levels are time slot dependent. It turned out that compared to the production model with a constant base-stock level, a small cost reduction can be obtained by increasing the base-stock level during the production period. It is expected that similar results can be obtained for the lost sales model. However, for the backlog model it is already difficult and time consuming to determine the optimal vector of base-stock levels, because of the large number of possible vectors. The backlog model has the advantage that it can be analysed with a generating function approach and a newsvendor type equation can be derived. This is not the case in the lost sales model, so although introducing time slot dependent base-stock levels could lead to lower expected costs in the fixed cycle policy, the time it takes to find these base-stock levels is very time consuming in both the backlog and the lost sales model.

### One step improvement approach

At each decision moment, the state $(j, k_1, \ldots, k_N)$ is observed, with $j$ the type currently set-up and $k_i$ the stock or shortfall level of type $i, i = 1, \ldots, N$. Depending on the state, one searches for the minimum relative value and executes the corresponding slot. Then, the state is observed again and the next decision is computed. The obtained strategy is analysed by means of simulation. For large systems, the relative value function of the new strategy does not have a special structure that makes it possible to determine the relative values if needed, like the decomposition property of the fixed cycle policy. So for large systems, the new strategy does not allow for another policy improvement.

### Performance

The new production strategy is compared with existing production strategies like gated and exhaustive base-stock control and results for the two different models were similar. For both the lost sales and the backlog model, results showed that the new production strategy performs well, particularly for highly loaded systems, systems with high penalty or backlogging costs and systems with a large number of product types.

It was also seen that the performance of the one step improvement policy is influenced by the order of production in the fixed cycle. It is difficult to get a good intuition for the optimal order of production, but the performance of the one step improvement policy can be significantly different for different production orders.

The base-stock levels that are optimal for the fixed cycle policy are not necessarily optimal for the one step improvement policy as well. It is seen that in some cases the one step improvement policy performs better if the values of the base-stock levels are set somewhat lower than the base-stock levels that are optimal for the fixed cycle policy, see for example the results in Table 2.1 in Section 2.6.

For production systems with lost sales, it is also seen that if one of the items has a relatively high demand rate, an extra production period in the fixed cycle policy for this item has a positive effect on the performance of the one step improvement policy. For production systems with backlog, it is expected that an extra production period for items with a relatively high demand rate also has a positive effect on the performance of the one step improvement policy. So increasing the number of production periods for one or more product types increases the load on the system for the fixed cycle policy, but for asymmetric systems, this can have a positive effect on the performance of the one step improvement policy.

Apparently, it is important to choose a fixed cycle that performs well, but it should also have the characteristics of a good production strategy. The base-stock levels and lengths of the production periods that are optimal for the fixed cycle policy are not equal to the maximum stock levels and visit periods that are expected in the one step improvement policy. Further research is needed to determine how to tune the characteristics of the fixed cycle policy so that the one step improvement policy performs best.

### Two machines

The one step improvement approach is also used to construct a dynamic production strategy in a system with 2 identical machines that can both produce all $N$ items. In such a model, hardly any other dynamic production strategies are known. In Chapter 3, it is assumed that production and set-up times are all deterministic and of unit length, so that decisions are taken simultaneously. It is also possible to relax these assumptions, because the relative values for this model can still be calculated per product type. Although the analysis and notation become more complex, a similar procedure as in Chapter 3 can be used. Decisions are not taken simultaneously and for the calculation of the relative values, one has to calculate the expected costs until the next decision moment for the machine that is still producing or being set-up for one of the product types.

The results for the model in Chapter 3 are promising, but more results are needed in order to draw conclusions on the performance of the strategy compared to other strategies.

### Lead times

It was already mentioned that in the backlog model, a generating function approach is used to analyse the system. Using this 'queueing' approach, distributional results are derived for the shortfall levels. The corresponding queueing system is a polling model with one server and $N$ queues. For such a system, the waiting time distribution for (globally) gated queues is found in Chapter 5. Looking at these results from a production point of view, one observes the lead time distributions for product types that are produced according to a gated base-stock policy. It is easily seen that these distributions are, just as the shortfall distributions, independent of the base-stock levels. The first and second moment of the lead time are found for different service (or production) disciplines, like FCFS, LCFS and ROS. The results for the first moment that were already obtained by Winands et al. [91], tell us that for all service time distributions, the Shortest Job First (SJF) policy gives the minimum expected lead time. A comparison between the second moments tells us that, if the service time distribution has a decreasing failure rate, the variance of the lead time under this policy is also lower than under the FCFS, LCFS and Random Order of Service (ROS) service disciplines.

A comparison between the gated and globally gated visit discipline comes down to a comparison between the cycle time distributions.

## 8.2 Further research

### Combinations of slots

The constructed one step improvement policy is dynamic and therefore, a second improvement step is intractable for large systems. However, it is possible to improve the new production strategy by looking two time slots ahead. This works as follows. Just like in the one step improvement approach, a good fixed cycle is found and the relative values are calculated per product type. Then, for each decision one considers every possible combination for the next two slots, after which it is assumed that the fixed cycle is followed again.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | set-up | 2 | 2 | 2 | 2 | set-up | 3 | 3 | set-up |

**Figure 8.1:** A fixed cycle for a 3-item production system

### Example

Consider a 3-item production system, for which the fixed cycle of Figure 8.1 is found. The possible combinations of time slots that are considered if item 1 is set-up, are the following:

- Two consecutive production slots of item 1,
- One production slot of item 1 and a set-up slot for item 1 (which can be seen as idling),
- One production slot of item 1 and a set-up slot for item 2,
- One production slot of item 1 and a set-up slot for item 3,
- A set-up slot for item 1 and one of the production slots of item 1,
- A set-up slot for item 2 and one of the production slots of item 2,
- A set-up slot for item 3 and one of the production slots of item 3.

For each combination, the costs for the first slot are calculated and, taking the one step transition into account, the relative value for the second slot is added. So, for a production slot of item 1 and followed by the set-up slot for item 3 (i.e. slot 11), one gets the following relative costs for item 1:

$$c_{1,1}(k_1) + \sum_{k_1'=0}^{k_1+1} p_{1,1}(k_1, k_1') r_1(11, k_1'),$$

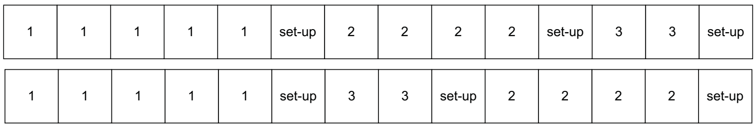
where $k_1$ denotes the current shortfall level of item 1 and slot 11 is the set-up slot for item 3 (see Figure 8.1). Although a decision is calculated for 2 slots simultaneously, first one of the two slots is executed. Then, a new decision is computed, etc.

Some combinations are also considered in the one step improvement policy studied in this thesis. For example, for the combination of one production slot of item 1 and a set-up slot of item 2 one can look at the relative value of slot 5. However, the second and fifth combinations are new, so the number of possible combinations is larger than the number of possible slots in the standard one step improvement policy.

Because the improvement step with the combined slots looks more slots ahead, it is expected that the constructed strategy outperforms the one step improvement policy. It is also possible to look at every combination of the next *three* or more slots. However, the number of combinations grows exponentially in the number of combined slots. So there is a limit to this number, especially if $N$ is large.

### Production order

Using a simple trick, it is possible to change the assumed order of production in the fixed cycle. Because the lengths of the set-up slots do not depend on the preceding product type, the lengths of the production periods remain the same if the production order is changed. Each product type still experiences the same periodic production and because the relative values are stored per product type, one just has to renumber the slots in the cycle if the production order is changed, see Figure 8.2. Therefore, the relative values only need to be calculated and registered once, for one order of production. Then, for each order of production, it is possible to look at the reachable time slots and determine the relative value. The slot with the minimum relative value is executed.



| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | set-up | 2 | 2 | 2 | 2 | set-up | 3 | 3 | set-up |
| 1 | 1 | 1 | 1 | 1 | set-up | 3 | 3 | set-up | 2 | 2 | 2 | 2 | set-up |

**Figure 8.2:** Two fixed cycles for a 3-item production system

If $N$ is large, there is a large number of possible production orders to consider, because the number of possible production orders equals the factorial of $N$. A small number of production orders can be considered that are chosen on the basis of the shortfall levels, demand intensities and holding and penalty costs.

# Bibliography

[1] J. Abate and W. Whitt. An operational calculus for probability distributions via laplace transforms. Adv. Appl. Prob, 28:75–113, 1996.

[2] I. J. B. F. Adan, J. S. H. van Leeuwaarden, and E. M. M. Winands. On the application of Rouché's theorem in queueing theory. *Operations Research Letters*, 34:355–360, 2006.

[3] T. Altiok and G. A. Shiue. Single-stage, multi-product production/inventory systems with lost sales. *Naval Research Logistics*, 42:889–913, 1995.

[4] R. Anupindi and S. Tayur. Managing stochastic multi-product systems: Model, measures and analysis. *Operations Research*, 46(3):98–111, 1998.

[5] K. Athreya and P. Ney. *Branching Processes*. Springer, Berlin, 1972.

[6] B. Avi-Itzhak and S. Halfin. Response times in gated $M/G/1$ queues: the processor-sharing case. *Queueing Systems*, 4:263–279, 1989.

[7] F. Baccelli and P. Brémaud. *Elements of Queueing Theory: Palm-Martingale Calculus and Stochastic Recurrences*. Springer, New York, 2003.

[8] J. S. Baras, D. J. Ma, and A. M. Makowski. Competing queues with geometric service requirements and linear costs; the $\mu$ c-rule is always optimal. Technical Research Report SSR-83-9, Electrical Engineering Department, University of Maryland, College Park, 1983.

[9] J. S. Baras, A. J. Dorsey, and A. M. Makowski. Two competing queues with linear costs and geometric service requirements: The $\mu$ c-rule is often optimal. *Advances in Applied Probability*, 17(1):186–209, 1985.

[10] S. Bhulai. Dynamic routing policies for multi-skill call centers. *Probability in the Engineering and Informational Sciences*, 23:101–119, 2009.

[11] M. A. A. Boon, I. J. B. F. Adan, and O. J. Boxma. A two-queue polling model with two priority levels in the first queue. *Proceedings ValueTools*, 2008.

[12] M. A. A. Boon, I. J. B. F. Adan, and O. J. Boxma. A polling model with multiple priority levels. *Performance Evaluation*, 67(6):468–484, 2010.

[13] S. C. Borst. *Polling systems.* PhD thesis, CWI, Amsterdam, The Netherlands, 1994.

[14] O. J. Boxma. Workloads and waiting times in single-server systems with multiple customer classes. *Queueing Systems*, 5:185–214, 1989.

[15] O. J. Boxma, W. Groenendijk, and J. Weststrate. A pseudoconservation law for service systems with a polling table. *IEEE Transactions on Communications*, 38:1865–1870, 1990.

[16] O. J. Boxma, H. Levy, and J. A. Weststrate. Efficient visit frequencies for polling tables: minimization of waiting cost. *Queueing Systems*, 9(1–2):133–162, 1991.

[17] O. J. Boxma, H. Levy, and U. Yechiali. Cyclic reservation schemes for efficient operation of multiple-queue single-server systems. *Annals of Operations Research*, 35:187–208, 1992.

[18] O. J. Boxma, G. M. Koole, and I. Mitrani. Polling models with threshold switching. *Quantitative Methods in Parallel Systems*, pages 129–140, 1995.

[19] O. J. Boxma, J. Bruin, and B. Fralix. Waiting times in polling systems with various service disciplines. *Performance Evaluation*, 66(11):621–639, 2008.

[20] S. Browne and G. Weiss. Dynamic priority rules when polling with multiple parallel servers. *Oper. Res. Lett.*, 12:129–137, 1992.

[21] J. Bruin. Cyclic multi-item production systems. *Proceedings Analysis of Manufacturing Systems*, 2007.

[22] J. Bruin and J. van der Wal. A dynamic control strategy for multi-item production systems. *Proceedings Stochastic Models of Manufacturing and Service Operations*, 2009.

[23] J. Bruin and J. van der Wal. A cyclic production scheme for multi-item production systems with backlog; part 1. *Submitted to Annals of OR*, 2010.

[24] J. Bruin and J. van der Wal. A dynamic control strategy for multi-item production systems with backlog; part 2. *Submitted to Annals of OR*, 2010.

[25] J. Bruin and J. van der Wal. A dynamic control strategy for multi-item production systems with lost sales. *Submitted to Annals of OR*, 2010.

[26] C. Buyukkoc, P. Varaiya, and J. Walrand. The c$\mu$ rule revisited. *Advances in Applied Probability*, 17(1):237–238, 1985.

[27] H. Chung, C. Un, and W. Jung. Performance analysis of Markovian polling systems with single buffers. *Performance Evaluation*, 19(4):303–315, 1994.

[28] D. R. Cox and W. L. Smith. *Queues.* Chapman and Hall, London, 1961.

[29] J. N. Darroch. On the traffic light queue. *Annals of Mathematical Statistics*, 35:380–388, 1964.

[30] R. de Haan, A. M. Al Hanbali, R. J. Boucherie, and J. C. W. van Ommeren. A transient analysis of polling systems operating under exponential time-limited service disciplines. *Memorandum 1894, Department of Applied Mathematics, University of Twente, Enschede*, (ISSN 1874-4850), 2009.

[31] A. G. de Kok. A moment-iteration method for approximating the waiting-time characteristics of the GI/G/1 queue. *Probability in the Engineering and Informational Sciences*, 3:273–287, 1989.

[32] N. P. Dellaert. Production to order: models and rules for production planning. *Lecture Notes in Economics and Mathematical Systems 333*, 1989.

[33] D. Denteneer, J. S. H. van Leeuwaarden, and I. J. B. F. Adan. The acquisition queue. *Queueing Systems*, 56:229–240, 2007.

[34] I. Eliazar and U. Yechiali. Polling under the randomly timed gated regime. *Stochastic Models*, 14(1-2):79–93, 1998.

[35] N. Erkip, R. Güllü, and A. Kocabiyikoglu. A quasi-birth-and-death model to evaluate fixed cycle time policies for stochastic multi-item production/inventory problem. *Proceedings of MSOM conference, Ann Harbor*, 2000.

[36] A. Federgruen and Z. Katalan. Approximating queue size and waiting time distributions in general polling systems. *Queueing Systems*, 18:353–386, 1994.

[37] A. Federgruen and Z. Katalan. The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times. *Management Science*, 42(6):783–796, 1996.

[38] A. Federgruen and Z. Katalan. Customer waiting-time distributions under base-stock policies in single-facility multi-item production systems. *Naval Research Logistics*, 43:533–548, 1996.

[39] A. Federgruen and Z. Katalan. Determining production schedules under base-stock policies in single facility multi-item production systems. *Operations Research*, 46(6):883–898, 1998.

[40] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 2. Wiley, 1971.

[41] B. Fleischmann. The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, 44(3):337–348, 1990.

[42] J. C. Fransoo, V. Sridharan, and J. W. M. Bertrand. A hierarchical approach for capacity coordination in multiple products single-machine production systems with stationary stochastic demands. *European Journal of Operations Research*, 86:57–72, 1995.

[43] S. W. Fuhrmann and R. B. Cooper. Stochastic decomposition in an $M/G/1$ queue with generalized vacations. *Operations Research*, 33:11–17, 1985.

[44] S. W. Fuhrmann and I. Iliadis. A comparison of three random disciplines. *Queueing Systems*, 18:249–271, 1994.

[45] S. E. Grasman, T. L. Olsen, and J. R. Birge. Finite buffer polling models with routing. *European Journal of Operations Research*, 165(3):794–809, 2005.

[46] S. E. Grasman, T. L. Olsen, and J. R. Birge. Setting basestock levels in multiproduct systems with setups and random yield. *IIE Transactions*, 40(12): 1158–1170, 2008.

[47] R. Güllü, N. Erkip, and S. Hafizogullari. Fixed cycle time policies for the stochastic multi-item production/inventory problem. *Working paper*, 2010.

[48] R. Haijema and J. van der Wal. An MDP decomposition approach for traffic control at isolated signalized intersections. *Probability in the Engineering and Informational Sciences*, 22(4):587–602, 2008.

[49] M. Hofri and K. W. Ross. On the optimal control of two queues with server set-up times and its analysis. *SIAM J. Comput.*, 16:399–420, 1987.

[50] R. Howard. *Dynamic Programming and Markov Processes.* The MIT Press, Cambridge, Mass., 1960.

[51] A. J. E. M. Janssen and J. S. H. van Leeuwaarden. Analytic computation schemes for the discrete-time bulk service queue. *Queueing Systems*, 50:141–163, 2004.

[52] M. Khouja. The single-period (news-vendor) problem: literature review and suggestions for future research. *Omega*, 27(5):537–553, 1999.

[53] G. Koole. Assigning a single server to inhomogeneous queues with switching costs. *Theoretical Computer Science*, 182:203–216, 1997.

[54] G. N. Krieg and H. Kuhn. A decomposition method for multi-product kanban systems with setup times and lost sales. *IIE Transactions*, 34(7):613–625, 2002.

[55] R. Y. W. Lam, V. C. M. Leung, and H. C. B. Chain. Polling-based protocols for packet voice transport over IEEE 802.11 wireless local area networks. *IEEE Wireless Communications*, 13:22–29, 2006.

[56] T. Y. S. Lee and J. Sunjaya. Exact analysis of asymmetric random polling systems with single buffers and correlated Levy input process. *Queueing Systems*, 23(3-4):131–156, 1996.

[57] A. A. J. Lefeber and J. E. Rooda. Controller design for flow networks of switched servers with setup times: the Kumar-Seidman case as an illustrative example. *Asian Journal of Control*, 10(1):55–66, 2008.

[58] H. Levy, G. Mahalal, and M. Sidi. Multi server polling systems: The bang bang policies. In *Proc. Experts on Networks Workshop, UCLA, June 1994*, 1994.

[59] Z. Liu, P. Nain, and D. Towsley. On optimal polling policies. *Queueing Systems*, 11:59–83, 1992.

[60] M. A. Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93–122, 1984.

[61] M. A. Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems*. MIT Press, Cambridge, USA, 1986.

[62] M. A. Marsan, S. Donatelli, and F. Neri. GSPN models of Markovian multiserver multiqueue systems. *Performance Evaluation*, 11(4):227–240, 1990.

[63] D. Miorandi, A. Zanella, and G. Pierobon. Performance evaluation of bluetooth polling schemes: an analytical approach. *Mobile Networks and Applications*, 9: 63–72, 2004.

[64] R. J. T. Morris and Y. T. Wang. Some results for multi-queue systems with multiple cyclic servers. *Performance of Computer-Communication Systems, In: eds. W. Bux and H. Rudin (North-Holland, Amsterdam)*, pages 245–258, 1984.

[65] J. M. Norman. *Heuristic procedures in dynamic programming*. Manchester University Press, Manchester, 1972.

[66] T. Osogami, M. Harchol-Balter, and A. Scheller-Wolf. Analysis of cycle stealing with switching times and thresholds. *Performance Evaluation*, 61(4):347–369, 2005.

[67] T. Ott and K. Krishnan. Separable routing: A scheme for state-dependent routing of circuit switched telephone traffic. *Annals of Operations Research*, 35 (1):43–68, 1992.

[68] E. L. Porteus. *Foundations of Stochastic Inventory*. Stanford University Press, Stanford, 2002.

[69] J. Qiu and R. Loulou. Multiproduct production/inventory control under random demands. *IEEE Transactions on Automatic Control*, 40(2):350–356, 1995.

[70] K. M. Rege and B. Sengupta. A single server queue with gated processor-sharing discipline. *Queueing Systems*, 4:249–261, 1989.

[71] J. A. C. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13:409–426, 1993.

[72] R. Rietman and J. A. C. Resing. An $M/G/1$ queueing model with gated random order of service. *Queueing Systems*, 48:89–102, 2004.

[73] S. A. E. Sassen, H. C. Tijms, and R. D. Nobel. A heuristic rule for routing customers to parallel servers. *Statistica Neerlandica*, 51(1):107–121, 1997.

[74] P. J. Schweitzer. Iterative solution of the functional equations of undiscounted Markov renewal programming. *Journal of Mathematical Analysis and Applications*, 34(4):494–501, 1971.

[75] R. Serfozo. *Introduction to Stochastic Networks.* Springer, New York, 1999.

[76] J. G. Shanthikumar and U. Sumita. Convex ordering of sojourn times in single-server queues: extremal properties of FIFO and LIFO service disciplines. *Journal of Applied Probability*, 24:737–748, 1987.

[77] S. R. Smits, M. Wagner, and A. G. de Kok. Determination of an order-up-to policy in the stochastic economic lot scheduling model. *International Journal of Production Economics*, 90(3):377–389, 2004.

[78] C. R. Sox and J. A. Muckstadt. Optimization-based planning for the stochastic lot scheduling problem. *IIE Transactions*, 29(5):349–357, 1997.

[79] H. Takagi. *Analysis of Polling Systems.* MIT Press, 1986.

[80] T. Takine, Y. Takahashi, and T. Hasegawa. An analysis for interdeparture process of a polling system with single message buffer at each station. *Trans. Inst. Electron. Znf. Commun. Eng.*, 70-B(9):989–998, 1987.

[81] H. Thörisson. *Coupling, Stationarity and Regeneration.* Springer, New York, 2000.

[82] H. C. Tijms and M. C. T. van de Coevering. A simple numerical approach for infinite-state Markov chains. *Probability in the Engineering and Informational Sciences*, 5:85–295, 1991.

[83] M. S. van den Broek, J. S. H. van Leeuwaarden, I. J. B. F. Adan, and O. J. Boxma. Bounds and approximations for the fixed-cycle traffic light queue. *Transportation Science*, 40(4):484–496, 2006.

[84] R. van der Mei and S. C. Borst. Analysis of multiple-server polling systems by means of the power-series algorithm. *Stochastic Models*, 13(2):339–369, 1997.

[85] M. J. A. van Eenige. *Queueing Systems with Periodic Service.* PhD thesis, Technical University Eindhoven, Eindhoven, 1996.

[86] P. van Mieghem. The asymptotic behavior of queueing systems: Large deviations theory and dominant pole approximation. *Queueing Systems*, 23:27–55, 1996.

[87] M. van Vuuren and E. M. M. Winands. Iterative approximation of k-limited polling systems. *Queueing Systems*, 55(3):161–178, 2007.

[88] V. Vishnevskii and O. Semenova. Mathematical methods to study the polling systems. *Automation and Remote Control*, 67:173–220, 2006.

[89] M. Wagner and S. R. Smits. A local search algorithm for the optimization of the stochastic economic lot scheduling problem. *International Journal of Production Economics*, 88:391–402, 2004.

[90] W. Whitt. The renewal-process stationary-excess operator. *Journal of Applied Probability*, 22:156–167, 1985.

[91] A. Wierman, E. M. M. Winands, and O. J. Boxma. Scheduling in polling systems. *Performance Evaluation*, 64(9–12):1009–1028, 2007.

[92] J. Wijngaard. Decomposition for dynamic programming in production and inventory control. *Engineering and Process Economics*, 4:385–388, 1979.

[93] E. M. M. Winands, I. J. B. F. Adan, and G. J. van Houtum. The stochastic economic lot scheduling problem: A survey. *BETA WP*, (133), 2005.

[94] E. M. M. Winands, I. J. B. F. Adan, and G. J. van Houtum. Mean value analysis for polling systems. *Queueing Systems*, 54:45–54, 2006.

[95] E. M. M. Winands, A. G. de Kok, and C. Timpe. Case study of a batch-production/inventory system. *Report BASF*, 2007.

[96] R. W. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice Hall, London, 1989.

# Index of Symbols

# Summary

This thesis focusses on the analysis and construction of control policies in multi-item production systems. In such systems, multiple items can be made to stock, but they have to share the finite capacity of a single machine. This machine can only produce one unit at a time and if it is set-up for one item, a switch-over or set-up time is needed to start the production of another item. Customers arrive to the system according to (compound) Poisson processes and if they see no stock upon arrival, they are either considered as a lost sale or backlogged. In this thesis, we look at production systems with backlog and production systems with lost sales. In production systems with lost sales, all arriving customers are considered lost if no stock is available and penalty costs are paid per lost customer. In production systems with backlog, arriving customers form a queue if they see no stock and backlogging costs are paid for every backlogged customer per time unit.

These production systems find many applications in industry, for instance glass and paper production or bulk production of beers, see Anupindi and Tayur [4]. The objective for the production manager is to minimize the sum of the holding and penalty or backlogging costs. At each decision moment, the manager has to decide whether to switch to another product type, to produce another unit of the type that is set-up or to idle the machine. In order to minimize the total costs, a balance must be found between a fast switching scheme that is able to react to sudden changes in demand and a production plan with a little loss of capacity. Unfortunately, a fast switching scheme results in a loss of capacity, because switching from one product type to another requires a switch-over or set-up time.

In the optimal production strategy, decisions depend on the complete state of the system. Because the processes at the different product flows depend on these decisions, the processes also depend on the complete state of the system. This means that the processes at the different product flows are not independent, which makes the analysis and construction of the optimal production strategy very complex. In fact, the complexity of the determination of this policy grows exponentially in the number of product types and if this number is too large, the optimal policy becomes intractable. Production strategies in which decisions depend on the complete system are defined as global lot sizing policies and are often difficult to construct or analyse, because of the dependence between the different product flows.

However, in this thesis the construction of a global lot sizing policy is presented which also works for production systems with a large number of product types. The

key factor that makes the construction possible is the fact that it is based on a fixed cycle policy. In Chapter 2, the fixed cycle policy is analysed for production systems with lost sales and in Chapter 6, the fixed cycle policy is analysed for production systems with backlog. The fixed cycle policy can be analysed per product flow and this decomposition property allows for the determination of the so called relative values. If it is assumed that one continues with a fixed cycle control, the relative values per product type represent the relative expected future costs for each decision. Based on these relative values, an improvement step (see Norman [65]) is performed which results in a 'one step improvement' policy. This policy is constructed and analysed in Chapters 2 and 7 for production systems with lost sales and production systems with backlog, respectively.

This global lot sizing policy turns out to perform well compared to other, heuristic production strategies, especially in systems with a high load and demand processes with a high variability. A similar approach as for the production system with a single machine is performed in a system with two machines and lost sales in Chapter 3. Results show that in some cases the constructed strategy works well, although in some systems two separate one step improvement policies perform better.

Examples of more heuristic production strategies are gated and exhaustive base-stock policies. In these 'local lot sizing' policies, decisions depend only on the stock level of the product type that is set-up. But even in these policies, the processes at the different product flows are dependent. This makes the analysis difficult, but for production systems with backlog a translation can be made to a queueing system by looking at the number of products short to the base-stock level. So the machine becomes a server and each product flow becomes a queue. In these queueing systems, also known as polling systems, gated and exhaustive base-stock policies become gated and exhaustive visit disciplines. For polling systems, an exact analysis of the queue length or waiting time distribution is often possible via generating functions or Laplace-Stieltjes transforms. In Chapter 5, the determination of the sojourn time distribution of customers in a polling system with a (globally) gated visit discipline is presented, which comes down to the determination of the lead time distribution in the corresponding production system.

# Samenvatting

In dit proefschrift wordt de analyse en constructie van productie strategieën besproken in multi-item productie systemen. In zulke systemen kunnen verschillende items worden geproduceerd en voor ieder item kan voorraad gehouden worden. Er is echter maar één machine die één product tegelijk kan maken, en bovendien kost het tijd om van het ene product type naar het andere product type om te schakelen. Klanten arriveren bij het systeem volgens (compound) Poisson processen en als zij geen voorraad aantreffen, worden zij of als verloren beschouwd of zij vormen een rij. Er worden in dit proefschrift twee modellen beschouwd, namelijk productiesystemen met lost sales (verloren klanten) en productiesystemen met backlog. In productiesystemen met lost sales worden alle klanten die geen voorraad treffen als verloren beschouwd en per verloren klant worden boetekosten betaald. In productiesystemen met backlog sluiten alle klanten die geen voorraad treffen aan in een rij en per wachtende klant worden wachtkosten per tijdseenheid betaald.

In de praktijk komen deze productie systemen veel voor, bijvoorbeeld in de glas- en papierindustrie of bij de productie van bier (zie Anupindi en Tayur [4]). Het doel van de manager van deze systemen is het minimaliseren van de voorraad- en boete- of wachtkosten. Er moet hierbij steeds gekozen worden voor een nieuwe productie, een omschakeling of het tijdelijk stilzetten van de machine. In een strategie waarin veel omgeschakeld wordt, gaat capaciteit verloren omdat het omschakelen een set-up of switch-over tijd vereist. Deze capaciteit is nodig om genoeg producten te kunnen maken. Om de kosten te minimaliseren moet daarom gezocht worden naar een balans tussen een strategie waarin veel omgeschakeld wordt zodat gereageerd kan worden op plotselinge veranderingen in de vraag, en een productieplan waarin weinig capaciteit verloren gaat.

In de optimale productie strategie zijn beslissingen afhankelijk van de volledige toestand van het systeem. De processen bij de verschillende voorraadpunten hangen weer af van deze beslissingen, dus deze processen zijn ook weer afhankelijk van elkaar. Dit maakt de analyse en in het bijzonder het vinden van de optimale strategie zeer complex. De complexiteit van het vinden van de optimale strategie groeit zelfs exponentieel in het aantal verschillende product types. Dit betekent dat het aantal product types al snel te groot wordt om de optimale strategie te kunnen bepalen. Door deze complexiteit is het ook lastig om een strategie waarin beslissingen van de volledige toestand van het systeem afhangen, te bedenken of te analyseren voor productie systemen met een groot aantal product types.

In dit proefschrift laten we zien dat met behulp van een éénstapsverbeteringstechniek van Norman (zie [65]), een productie strategie geconstrueerd kan worden voor grote systemen die kijkt naar de volledige toestand van het systeem. Bij deze éénstapsverbeteringstechniek worden relatieve kosten bepaald per toestand en beslissing, in de veronderstelling dat na deze beslissing een bepaalde basis strategie wordt gevolgd. De keuze van de basis strategie is hier belangrijk, omdat in principe het berekenen van de relatieve kosten per toestand niet mogelijk is voor (te) grote systemen. Maar in een strategie met een vaste cyclus (fixed cycle strategie) kunnen alle product types apart geanalyseerd worden, omdat de processen van de product types onderling onafhankelijk zijn bij deze strategie. Hierdoor kunnen ook de relatieve waarden per product type berekend en opgeslagen worden. Dit wordt voor productiesystemen met lost sales laten zien in hoofdstuk 2 en voor productiesystemen met backlog in hoofdstuk 6. Op basis van de relatieve kosten wordt een nieuwe productie strategie bepaald, die 'one step improvement' strategie wordt genoemd. Deze strategie wordt in hoofdstuk 2 geanalyseerd voor productiesystemen met lost sales. Hoofdstuk 7 behandelt de analyse en constructie van deze strategie voor systemen met backlog.

De nieuwe strategie blijkt goed te werken in vergelijking met andere, meer heuristische productie strategieën, in het bijzonder voor systemen met een hoge bezettingsgraad en vraag processen met een hoge variantie. Een zelfde aanpak is gebruikt voor productie systemen met twee machines en lost sales in hoofdstuk 3 en leidt tot goede resultaten, hoewel er ook systemen zijn waarin twee aparte éénstapsverbeteringen lagere kosten geven.

Voorbeelden van meer heuristische productie strategieën zijn gated en exhaustive base-stock strategieën. In deze strategieën hangen beslissingen alleen af van het voorraadniveau van het product type waarop de machine ingesteld is. Omdat ook bij deze strategieën de processen bij de verschillende voorraadpunten afhankelijk zijn van elkaar, is de analyse van deze processen vaak erg complex. Voor productie systemen met backlog kan een vertaling gemaakt worden naar wachtrij systemen door per product type te kijken naar het aantal producten tekort ten opzichte van het base-stock niveau. Deze wachtrij systemen hebben dan één bediende en meerdere wachtrijen, en staan ook wel bekend als polling modellen. Een exacte analyse van bijvoorbeeld de rijlengte verdelingen is vaak mogelijk met behulp van genererende functies en Laplace-Stieltjes getransformeerden. In hoofdstuk 5 is voor zulke polling modellen de verdeling van de verblijftijd van een klant afgeleid, waarbij aangenomen wordt dat de bediende de rijen bedient volgens een (globale) gated strategie. De verblijftijd van een klant in een polling model komt overeen met de doorlooptijd van een product in het corresponderende productie systeem.

# About the author

Josine Bruin was born on August 15, 1983 in Zaanstad, the Netherlands. She graduated from grammar school (Sint Michael College, Zaandam) in June 2001. In November 2005 she received her Master's degree in Econometrics and Operations Research from Vrije Universiteit Amsterdam. In May 2006, she started a PhD project under the supervision of Jan van der Wal, Ton de Kok and Onno Boxma, at research institute Eurandom, in Eindhoven. This project also took place at the faculties of Mathematics & Computer Science and Industrial Engineering & Innovation Sciences at Eindhoven University of Technology. On October 12, 2010, Josine defends her PhD thesis at Eindhoven University of Technology. As of September 2010 she will be working as a consultant at Quintiq in Den Bosch.