Pooling and polling : creation of pooling in inventory and queueing models

DOI:
10.6100/IR732112

Document status and date:
Published: 01/01/2012

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

Pooling and Polling

Creation of Pooling in Inventory
and Queueing Models

van Wijk, Alexandra C.C.

Pooling and Polling: Creation of Pooling in Inventory and Queueing Models / by Alexandra C. C. van Wijk.

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3132-5

This thesis is number D 154 of the thesis series of the Beta Research School for Operations Management and Logistics.

Printed by: Proefschriftmaken.nl

Pooling and Polling

Creation of Pooling in Inventory
and Queueing Models

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 24 april 2012 om 14.00 uur

door

Alexandra Cornelia Catharina van Wijk

geboren te Leuven, België

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. I.J.B.F. Adan
en
prof.dr.ir. G.J.J.A.N. van Houtum

Contents

Part I

Pooling

1

Introduction pooling

In this part of the thesis we focus on pooling of resources. We consider respectively *pooling of inventory* and *pooling of server capacity*. We start by a discussion on pooling of resources in general, followed by introductions to each of the parts. Finally, we address the connection between pooling and polling models.

1.1 Pooling of resources

Pooling refers to the grouping of resources and demand streams in order to maximize the efficient use of it. One can obtain a better match between the resources and demand streams, and hence a scarce resource can be better utilized. Typically this results in a more profitable way of exploiting the resources, a higher utilization rate, and a risk reduction. The same performance level might be achieved with less resources, reducing the capital expenditure, or one achieves shorter waiting times or more service completions. In general, resources that are available to multiple customer streams, can be pooled to achieve better utilization.

Pooling can be applied to a wide variety of resources, e.g. assets, equipment, personnel, effort. Examples include manufacturing facilities sharing production equipment between several job types, clinical departments in a hospital sharing operating rooms, hospital beds, and medical staff, a computer network clustering computational power of the processors, and so on. In the service industry, one groups resources together from which several customer classes can be served, rather than having dedicated, separate resources for each individual customer class. In a separated system, the situation might occur that the dedicated resource of a customer is unavailable, although another resource in the system is available. This leads to extra waiting time for or the loss of this customer. By pooling of the resources, this situation can be prevented. This is both beneficial for the individual customer, as well as for the system as a whole.

We focus on two applications for which resources are pooled. Firstly, we consider pooling of inventory, in two settings. The first setting is a common stockpile from which multiple customer classes are served. Here we pool the otherwise separate inventories for these classes. In the second setting, the inventories of multiple local warehouses are pooled by allowing stock transfers between them. These stock transfers are called *lateral transshipments*. For both settings, one can do with less inventory while keeping the service at the same level, or even improving it, compared to the situation without pooling. For this, we optimize the allocation of the demands. Secondly, we consider pooling of server capacity. In this part we first consider a server farm, where a large number of servers is clustered to serve a stream of arriving work. Then we focus on a manufacturing model where the production capacity of a single server is pooled for two classes of customers, where the server can work on jobs of both classes simultaneously.

In both parts, we study the optimal use of pooling. That is, we characterize the decisions to be taken for the optimal use of the inventory, respectively servers. In all chapters, we derive *optimal dynamic policies*, which are policies that may depend on the whole state of the system. E.g. for the inventory models, we let the decisions depend on the actual on-hand inventories at all warehouses upon the decision epoch. In each chapter, we completely characterize and prove the structure of the optimal policy. That is, we derive the optimal policy for satisfying the demands or servicing the customers, minimizing the average (or discounted) costs of the system in the long-run. For this, we use stochastic dynamic programming. In Chapter 2, we outline the general approach along the lines of a single stockpoint example with multiple customer classes. The optimal policy we typically find, is a state-dependent threshold type policy. That is, the optimal policy is characterized by one or multiple switching curves. In addition, in most of the chapters, we also derive (sufficient) conditions under which the optimal policy simplifies. Only in Chapter 5 we assume a given policy structure characterized by a number of parameters, for which we provide an approximation algorithm that can be used to optimize these parameters. As in all models no pooling is a possible policy, the performance of the systems cannot deteriorate by the use of pooling.

The main research questions we address, are the following:

- What is the optimal pooling policy?
- How can the optimal policy be characterized?

Furthermore, we focus on the question when pooling is beneficial and we address the question how the optimal pooling policy relates to simpler policies. In particular, we focus on the following research questions:

- How much costs may be saved by the application of pooling, and what are the conditions under which savings are obtained?
- Under what conditions is a simple policy optimal?
- What is the improvement of the overall optimal policy compared to simpler policies, or a given class of parameterized policies?
- When we assume a specific form of the pooling policy, which is characterized by a number of parameters, how should we optimize these parameters?

Note that the one but last question is also interesting from an implementation point of view, as a simple, straightforward policy, or a parameterized policy, may be much more attractive than the overall optimal policy, which might be highly complex. This relates to the last question, on the optimization within a given class of parameterized policies.

In all chapters (except Chapter 5), we answer the main research questions, by proving the optimal policy structure for the problems considered. The other research questions are addressed in some of the chapters. In Chapter 5 we solely focus on a parameterized policy.

REMARK 1.1.1 (Notation). In each chapter we introduce notation separately, unless stated otherwise. In particular, we repeatedly use the approach for proving the optimal policy structures as outlined in Chapter 2.

REMARK 1.1.2 (Optimal control). We assume that the pooled resources are owned by a single entity, who takes system optimal actions. In game theoretic settings, problems are considered where the resources are owned by independent parties, which are not the models we focus on. The optimal policies that we derive, are optimal for the system as a whole.

1.2 Pooling of inventory

In the first part we focus our attention on pooling of inventory. We consider models with a single stockpoint where the inventory is pooled between multiple customers classes by the use of stock rationing (Chapters 2 and 6), and we consider multi-location models where inventory is pooled between the stockpoints by the use of stock transfers, so-called lateral transshipments (Chapters 3, 4, and 5).

1.2.1 Inventory models

When multiple customer classes are served by a single stockpoint, this can be seen as a form of pooling. By combining the otherwise separated stocks one achieves higher fill rates, or can achieve the same fill rate with less inventory. However, because of differences in the profitability between the customer classes, *stock rationing* might be applied to achieve higher profit. That is, when the on-hand inventory level becomes low, one may stop serving the least profitable customer classes, in order to keep stock on-hand in case of future demands from more profitable classes. The concept of stock rationing was first addressed by Topkis [183], and has since then been studied for a wide variety of applications and model variations (see [180] for an overview). We study such a model in Chapter 2 for the case that unsatisfied demands are lost, or satisfied via an emergency delivery (from elsewhere). For this, we derive the optimal policy structure for when to accept or reject demands from a certain class, minimizing the average costs in the long run, consisting of the inventory holding costs and the lost sales costs. The optimal action depends on the demand class and the actual stock level. In Chapter 6 we extend this problem by including the option to backorder a demand. Hence, each time a demand arises, we take the optimal action, choosing from satisfying the demand from stock, backlogging it, or a lost sale. For a problem with two demand classes, we derive the optimal dynamic policy structure. That is, we characterize the structure of the optimal actions which depend on both the on-hand stock level and the number of outstanding backorders. Also, we derive conditions under which the optimal policy structure simplifies.

Furthermore, we consider multi-location inventory models, where the inventory is pooled by the use of stock transfers between the locations. These stock transfers are referred to as *lateral transshipments*. Typically, when a lateral transshipment is applied, the request part is available at the location earlier than in case one has to wait for a shipment from a central warehouse. However, it depletes the on-hand inventory level at the location the lateral transshipment originated from. Hence, there is a trade-off between these issues.

The beginning of modern inventory management is attributed to Harris [99, 100]. In his 1913 work, he derives the economic order quantity (EOQ) formula, see also [70]. The analysis of models incorporating stochasticity started in the early 1950s [97, 197]. Since the late 1950s, attention has also been given to more complex systems, including models with multiple stocking locations [52]. More recent books about inventory management, in which more elaborate literature reviews can be found, include Zipkin [222], Porteus [154], and Axsäter [13].

1.2.2 Spare parts inventory models

The motivation for our studies is found in spare parts inventory models. These provide repairable spare parts for a critical component of advanced technical systems. The loss of revenue because of downtime for these systems may be huge. For that purpose, ready-for-use spare parts are kept on stock, to be able to quickly respond to a breakdown of a system. In that case, the system demands a spare part at one of the stockpoints. Depending on the model considered in each of the chapters, there are multiple options how such a demand can be satisfied: directly from stock, via a lateral transshipment, via an emergency repair procedure, or the demand can be backlogged. We consider spare parts that are typically expensive, while demand rates are low. We investigate when it is beneficial to pool this inventory between multiple demands classes, and between multiple warehouses, respectively.

The cheapest option is to satisfy a demand directly from stock, which is possible when on-hand inventory is available at the stockpoint. In this case, the ready-for-use spare part is shipped to the site of the machine, where the failed part in the machine is replaced by this spare part, after which the machine is working properly again. This strategy is called a repair-by-replacement procedure. The defective part is shipped back to the stockpoint, where it is repaired and added back to stock again. In case of a lateral transshipment, a spare part is shipped from another stockpoint. This is possible when another location has on-hand stock available. In this case, the system is down while it is waiting for the part (i.e. higher loss of revenue), and extra transportation costs are incurred, and hence this procedure is slightly more costly. For an emergency repair procedure, even higher costs are incurred. In this case, the part is repaired in a fast repair procedure, e.g. on site. Alternatively, an emergency shipment from a central warehouse might be applied. A fast repair procedure is expensive, and the system is down for a longer period of time. Because of the expensiveness of an emergency procedure, costs can be saved by the efficient use of lateral transshipments. In Chapter 6, we also allow backlogging of demand. In this case, one time costs are incurred, e.g. for the transportation costs, as well as backlog costs per time unit, because of the downtime of the system. Our goal is to choose the optimal action each time a demand occurs. We let the optimal action depend on the state of the system, i.e. on the inventory levels at all locations (and on the number of outstanding backorders in Chapter 6). Hence, we derive optimal dynamic policies.

We study the models in a continuous review setting, where all stockpoints execute a base stock policy. That is, the number of parts in circulation in fixed. We assume

Poisson demand processes, and allow for asymmetric demand rates and asymmetric costs structures at the locations. Each location has ample repair capacity, and we assume repair times to be exponentially distributed. The repairs can e.g. be done in a repair shop or repair facility nearby the stockpoint, or being outsourced to an external party.

For inventory problems with lateral transshipments, currently only limited insights are available on optimal policy structures; see Gross [93], Krishnan and Rao [121], Das [58], Robinson [162], Archibald et al. [6], Axsäter [11], Wee and Dada [204], Zhao et al. [217], and Herer et al. [104]. There is an urgent need for much better insights. There is a substantial amount of literature in which heuristic policies for lateral transshipments are studied; see Wong et al. [213, 212] and Kranenburg [117], and the references therein. Further, it is known that a lot of costs can be saved via lateral transshipments; see in particular Kranenburg [120], who showed this for a spare parts inventory control problem at ASML. However, there is a lack of insights into when exactly a lot of costs can be saved by the use of lateral transshipments. This depends on the inventory holding costs of spare parts, the costs for lateral transshipments, and the costs for emergency procedures. We try to close this gap in the literature by this study, where we specifically focus on systems with ample repair capacity, where unsatisfied demands are lost (i.e. fulfilled by an emergency procedure).

Remark 1.2.1 (Terminology). We have chosen to use the terminology of repairable spare parts throughout this chapter, although the models described apply more generally. Consumable spare parts fit into the same framework, as does basically any stock keeping unit that is replenished from an exogenous source or being produced to stock, provided a base stock policy is executed (see also the discussion in [212, Section 2.2]).

1.2.3 Models

In Chapter 3 we study a model with a so-called *quick response warehouse* (QR warehouse), and multiple local warehouses. The QR warehouse provides a part to stocked-out local warehouses in a shorter time than the time of an emergency shipment from the central warehouse. However, as different locations might have different costs for not being able to satisfy a demand, the QR warehouse should apply a form of stock rationing to minimize the total costs. Namely, if a location with rather low costs for rejecting a customer demand requests a part at the QR warehouse, while another location with high costs is (nearly) stocked-out, one better holds one or a few parts back in the QR warehouse in order to be able to satisfy future lateral transshipment requests from this warehouse. Compared to Chapter 2, the QR warehouse can be seen as a single-location stockpoint serving multiple demand classes (the lateral transshipment requests from the local warehouses) which is applying stock rationing. However, where the optimal decisions in the problem of Chapter 2 depend only on the on-hand inventory level of the single-location, for the QR warehouse problem, these decisions depend also on the stock levels in all local warehouses. We characterize the optimal policy structure for these decisions, and also provide sufficient conditions under which the optimal policy simplifies.

We continue in Chapter 4 with a model consisting of two locations. Now, we allow lateral transshipments in both directions. That is, each warehouse might request for a lateral transshipment from the other warehouse. We again characterize the optimal policy structure. The decision whether a lateral transshipment is optimally applied, depends on the on-hand inventory levels at both locations. Also, we derive sufficient conditions under which simpler policies are optimal. In the optimal policy, a form of stock rationing at one of the locations might appear. That is, if one of the locations has (much) higher costs for not satisfying a demand, the other location might hold back one or a few parts from its own demands, in order to be able to satisfy future lateral transshipment requests from the other location. We derive conditions under which both locations always satisfy demands directly from their own stock, i.e. under which this stock rationing is suboptimal. Then, parts can only be held back from lateral transshipment request from the other stockpoint. When this is the optimal strategy for both stockpoints, we call this a *hold back pooling policy*. Also, we derive conditions under which it is always optimal to apply a lateral transshipment in case of a out-of-stock. Combining these, the optimal policy is a *complete pooling policy*, in which both warehouses basically act as being one large warehouse. In this chapter we also address so-called *proactive lateral transshipment*, which are lateral transshipments not triggered by a demand arrival. For example, when the stock levels between the two location become very asymmetric at some point in time, one might decide to already ship parts from the location with excess inventory to the other one. We also characterize the optimal policy for these transshipments.

In Chapter 5 we extend the model of Chapter 4 to multiple locations. However, we are not able anymore to fully characterize the optimal policy structure. Therefore, we assume a given policy, and use that to determine the performance characteristics of the resulting model. More specifically, we let the system execute a hold back pooling policy. Then, we are interested in the fraction of demand that are satisfied directly from stock, satisfied via a lateral transshipment, or lost. For this we introduce a new approximation algorithm, which uses interrupted Poisson processes. In an extensive numerical study, we show that this algorithm is very accurate and results in much smaller approximation errors than algorithms currently used in the literature.

REMARK 1.2.2 (Demands and repair lead times). In all chapters, we assume that demands arrive according to Poisson processes, and that the repair lead times are exponentially distributed. Hence, we can formulate the problems as Markov decision problems, where the decisions are based on the actual stock levels, as pipeline information can be ignored. It is known that the performance of the optimal policy can be expected to be nearly insensitive to the distribution of the repair lead times, as shown in a.o. [2, 69, 120] for similar models.

1.2.4 Relation to queueing systems

There is a strong relation between models in queueing theory and the inventory models that we consider. Each stockpoint can also be interpreted as a multiserver queue, more specifically, an Erlang loss queueing system. For that, the on-hand inventory corresponds to the idle servers in the queueing model, the demands correspond to customer arrivals, and the replenishments (repairs) are the service completions. The number of servers in the queueing model is equal to the base stock level, i.e., the initial number of spare parts. An Erlang loss system has no waiting positions, and customers arriving when all servers are occupied, are lost. This corresponds to a demand in case of a stock-out. Hence, as the inventory and queueing models coincide from a mathematical point of view, results and insights for the inventory problems can be directly translated into results and insights for queueing problems, and vice versa. Basically, in this way the inventory pooling can also be seen as a form of server pooling between multiple queueing systems.

Pooling in these kind of queueing systems dates back to Smith and Whitt [169]. They show that pooling servers of several Erlang delay systems (i.e. combining the servers into a single system) is always beneficial when the service time distributions are identical. However, as Van Dijk and Van der Sluis [189] show, pooling is not necessarily beneficial if service times are asymmetric, as customers with shorter service time distributions may disproportionately suffer from the customers with long service time distributions. In the models we consider, the service times (i.e. repair times) are typically identically distributed.

1.2.5 Outline

The outline of the part on inventory pooling is as follows. We start by a single stockpoint stock rationing problem in Chapter 2. We mainly use this to demonstrate our approach of proving the optimal policy structure, as we use in all other chapters (except in Chapter 5). Then, in Chapter 3, we consider a multi-location setting, where only a so-called *quick response warehouse* can send out lateral transshipments. In Chapter 4 we consider a two location model, where we allow lateral transshipments between both warehouses. Then, in Chapter 5, we present a new approximation algorithm to determine the performance characteristics of a multi-location setting with lateral transshipments between all locations. Finally, in Chapter 6 we return our attention to a single-location stock rationing problem, now with a combination of backorders and lost sales. We show the remarkable similarity between this model and the two location lateral transshipment model of Chapter 4. We provide a detailed discussion of the relevant literature to each of the models in the chapters concerned.

1.3 Pooling of server capacity

The second part consists of two chapters on the pooling of server capacity, which also is a form of pooling of resources. In both chapters we use the same kind of techniques as in the part on pooling of inventory. Also, we assume Poisson arrival processes, and exponentially distributed service times.

Firstly, in Chapter 7, we consider a so-called *server farm*. This is a cluster of multiple servers which serves a stream of arriving customers. Pooling is achieved by the clustering of the servers. In this way, however, there is an excess of capacity. As costs are incurred for each server that is turned on, e.g. because of power consumption, one can save costs by turning servers off. Moreover, costs are also incurred for switching a server from on to off, and visa versa. When to turn servers on or off might depend on the amount of work the system is facing at the moment. That is, we want to find an optimal dynamic policy for switching the servers on and off. There is a trade-off between the costs for keeping servers idle (keeping them on although there is no work) and the costs for the switching from on to off, and visa versa. Depending on these cost parameters, it might be beneficial to keep one or more server idle, in order to prevent the costs from turning it off and having to turn it on again. We derive the optimal policy structure, determining when one should turn off a server after a service completion, and when it is better to let it idle. This model is a queueing problem with a dynamic number of servers, whereas in the inventory part we considered models where the number of servers is given. By allowing the numbers of servers to be adapted over time, this becomes a decision variable, where one has extra cost factors for these servers.

In the second chapter, Chapter 8, we consider a production system which serves two classes of customers: regular customers and opportunity customers. The latter provide an opportunity for the system to generate some extra income, as these customers can be accepted or rejected, while it is required that the regular customers are always served. Accepting opportunity customers might be particularly interesting when the workload of the regular customers is low. In that way, the problem can be interpreted as a workload control problem. One has to decide whether to accept or reject an arriving opportunity customer. Moreover, when such a customer is accepted, one has to decide how to allocate the server capacity between the two customers classes. The system has a single server, which is applying the head-of-line processor sharing discipline. That is, one regular and one opportunity customer can be in service at the same time, where the amount of server capacity dedicated to each is a continuous decision variable. Again, we derive the optimal dynamic policy structure, for both decisions.

1.4 Pooling and polling

The second part of this thesis focuses on polling models. A polling model consists of multiple queues that are served by a single server. Typically, the server visits the queues in cyclic order, and switchover times are incurred when the server switches the queue it is working on. Basically, this is also a form of pooling. Namely, the server pools its capacity between the queues.

The main differences with the pooling of server capacity, are (i) that we include switchover times between the services of different customers classes, and (ii) the policy structure assumed. Whereas in the pooling studies we derive optimal dynamic control policies, for the polling models we assume a given policy. This policy describes a.o. that the server cyclically works on the customer classes, and when the server switches to the next customer class. For such a policy, we either evaluate the performance characteristics of the studied models, or we optimize the parameters the policy depends on. Also, we introduce new variations on existing policies. When the switchover times tend to zero, the models reduce to the pooling models, when a pre-specified, parameterized control policy is executed. Note that assuming a given policy is typically done in the polling literature, where work on optimal dynamic control policies is scarce. Hence, as this is a different point of view than the optimal dynamic control policies studied in the first part, we devote a separate part of this thesis to polling models.

Pooling of inventory

2

STOCK RATIONING

As an introduction to the inventory models studied in this part of the thesis, we start of with a basic, one dimensional stock rationing problem. For this model, we prove the optimality of a so-called critical level policy using event based dynamic programming. The analysis of this system serves as a leg up to the more complicated multi-location models with stock transshipments in the sequel of this part.

2.1 Introduction

We consider a single stockpoint inventory model, which is facing demands of different importance. These demands are categorized in classes, the so-called *demand classes*. For example, the demands from long term, loyal customers might be more important than those from occasional customers. The costs for not satisfying a demand depend on the class the customer belongs to. In this way, the priorities of the demand classes can be ordered with respect to these cost. When the on-hand inventory level becomes low, it might be beneficial to stop serving lower priority customers, in order to be able to satisfy future demands of higher priority. In this way, stock is reserved for demands of higher importance. This is called *stock rationing*.

A model with multiple demand classes was first studied by Veinott [196] in 1965. He introduced the concept of a *critical level*, an inventory at or below which only high priority demands are served. This results in a *critical level policy* (cf. [183]), a policy under which there exists a number of critical levels, one for each demand class, such that a demand is satisfied when the on-hand stock is above its critical level, and rejected otherwise. Topkis [183] proves the optimality of this policy for both the backlog and lost sales case, for a periodic review model with zero lead times. The critical levels are non-increasing in the priority of the class.

The stock rationing problem has been extensively studied in the literature, for various settings and a wide range of model variations. Below we list a number of references, classifying whether a periodic or continuous review setting is assumed, and whether unsatisfied demands are backlogged or lost. Teunter and Klein Haneveld [180] provide a more detailed overview of the literature in diagram form, distinguishing between the time modeling (discrete or continuous), the shortage treatment (backorders or lost sales), the number of classes (two or multiple), the system (production or inventory), whether an ordering policy is used (yes or no), and the stock rationing policy (no, static or dynamic).

There are two types of decisions to be made, namely when a replenishment order should be placed (e.g. by executing a base stock policy), and whether an arriving demand is satisfied (stock rationing). In the first study of Veinott [196] a periodic review setting is considered, where unsatisfied demands are backlogged. He proves the optimality of a base stock replenishment policy, for a model without rationing. Topkis [183] studies a similar model, allowing stock rationing. He proves the optimality of a critical level policy both for the backordering and lost sales cases. Periodic review models are also considered in [47, 66, 101, 108, 145, 146, 179, 180, 219] for the case of backorders and in [54, 71, 81] for lost sales.

Nahmias and Demmy [148] were the first to study the stock rationing problem in a continuous review setting. They consider both the periodic and continuous review case for a model with backorders. For various settings, continuous review models are also considered in [7, 48, 62, 63, 59, 60, 74, 75, 88, 95, 110, 147, 198, 199, 214, 201] for backorders, and in [15, 16, 19, 45, 61, 75, 94, 96, 118, 127, 141, 139, 140, 190] for lost sales. In Chapter 6 we discuss the relevant literature for models that assume a combination of lost sales and backorders [4, 21, 20, 46, 69, 156, 170, 178, 220, 221].

A system with ample production capacity has (infinitely many) parallel servers. Unless a deterministic lead time is assumed, replenishment orders may cross in time. That is, an order placed at a later moment in time than another order, might arrive earlier. For such a system, the optimality of a simple base stock policy for the replenishment orders, cannot be guaranteed, see Erhardt [67]. Bulut and Fadıloğlu [43] study the optimal order policy for a model similar to ours, for a finite number of parallel production channels, making the assumption that previously placed production orders cannot be canceled. They prove that the optimal production policy is a state-dependent base stock policy, which depends on the number of production channels in use. The number of production channels that should be used, is non-increasing in the on-hand inventory level. Furthermore, the optimal rationing policy is proven, which is a state-dependent threshold type policy.

When dropping the assumption that placed orders cannot be canceled, a so-called bang-bang policy, which is characterized by a single threshold, becomes optimal: use all available production channels, until the on-hand inventory level reaches a given threshold, and from then on all channels are idled. Namely, by using all channels, the expected duration until the first replenishment is minimized. In the limit, when the number of production servers tends to infinity, this would mean that this duration tends to zero. For the model with finitely many servers, a critical level policy is optimal for the stock allocation.

Assuming a base stock policy in a continuous review setting with lost sales, we prove in this chapter that a critical level policy is optimal (cf. [143, 190]). For optimization of the critical levels and the base stock level in this setting, Kranenburg and Van Houtum [119] provide three heuristics. These assume a given base stock level, but for optimization purposes this base stock level can be enumerated in a separate loop, using bounds for this level derived by Dekker et al. [61]. Van Jaarsveld and Dekker [190] prove that two of the algorithms in [119] always provide the optimal critical levels and terminate in finitely many steps.

The on-hand inventory level evolves over time in the same way as the number of idle servers in an Erlang loss queueing model. This is a queueing model with multiple servers and no waiting room. The parts on-stock resemble the idle servers, the demands resemble the customer arrivals, and the replenishments resemble the service completions. Such a queueing model with multiple customers types was first studied by Miller [143] in 1969. He proves that the optimal policy for admitting customers (satisfying demands) is characterized by critical levels, and that customers from the highest priority class are always admitted. In [190] it is proven that these results remain valid when the replenishment assumptions are relaxed.

In this chapter, we consider a continuous review model, with lost sales, where we assume ample production capacity, with circulating stock (base stock policy). As our model is motivated by a spare parts inventory problem with repairable parts, see Chapter 1, we use the terminology of such a problem (see Remark 1.2.1). This justifies the base stock assumption. Moreover, the lost sales are interpreted as emergency procedures, and the productions/replenishments are repairs of broken parts. We prove that the optimal rationing policy is a critical level policy. This result is already known in the literature, however, we use a different technique, namely that of *Event Based Dynamic Programming* (cf. Koole [115, 116]), to prove it. We present this model as an introduction to the work in this part of this thesis, and to introduce the mentioned technique to the extent that it is used in this thesis.

This chapter is organized as follows. Firstly, we introduce the basic stock rationing model and its notation in Section 2.2. Then, in Section 2.3, we give a general outline along which the structural results derived in this thesis are proven. Returning to the stock rationing problem, we describe its dynamic programming formulation in Section 2.4. The structural results are derived in Section 2.5, which also contains an example. All proofs are given in Appendix 2.A.

2.2 Model description and notation

We consider a single stockpoint, keeping repairable spare parts of a single type on stock for technically advanced machines. Initially, there are $S \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$ parts on stock. By x we denote the on-hand stock, $x \in \mathcal{S} = \{0, 1, \ldots, S\}$, where $\mathcal{S}$ is the state space. A demand can be (i) satisfied directly from stock, if $x > 0$, or (ii) be satisfied via an emergency procedure. When the demand is fulfilled from stock, the ready-for-use spare part is installed in the machine. The failed part is brought back to the stockpoint, where it is repaired and added back to stock again. In this way, the down time of the machine is reduced to a minimum. In case the demand is not directly satisfied from stock (which might be the case even if there is on-hand stock), the demand must be satisfied via an expensive emergency repair procedure. Then, the failed part is repaired in a fast repair procedure, e.g. on-site, after which the machine is working properly again.

Holding costs $\tilde{h}(x)$ are incurred per time unit, $\tilde{h}(\cdot)$ being a convex non-decreasing function. There are J demands classes, $j = 1, \ldots, J$, where a demand from a customer belonging to demand class j is referred to as a class j demand. Each customer demands a single part. The demands from class j follow a Poisson process with rate $\lambda_j > 0$. The repair times are exponentially distributed with mean $1/\mu$ ($\mu > 0$), and there is ample repair capacity. That is, the repair rate is linear in the number of outstanding orders. Furthermore, we assume all processes to be mutually independent.

When a demand arises, it is either directly satisfied from stock (possible if $x > 0$), or rejected. In the latter case, the demand has to be fulfilled by an emergency procedure, at penalty costs c_j for a class j demand. We assume (w.l.o.g.)

$$c_1 \geq c_2 \geq \ldots \geq c_J > 0. \tag{2.2.1}$$

So, it is most expensive to reject a class 1 demand. Hence, we say that these customers have the highest priority, where class J customers belong to the lowest priority class. Because of the cost structure, it might be beneficial to hold parts back from lower priority customers, in order to be able to satisfy future higher priority demands, as those are more expensive to reject.

In the sequel we derive the optimal policy structure for this problem, that is, the policy that minimizes the average long run inventory holding and penalty costs per time unit. For this, we use Event Based Dynamic Programming, which we introduce in the next section.

2.3 General outline for proving structural results

In this section we present the general outline along which the optimal policy structures presented in this thesis are derived.

2.3.1 Dynamic programming

When facing a decision, we should take into account the direct costs for that decision, as well as the future expected costs this decision brings along. For the expected costs from a state x (in general being a vector), we introduce the *value function* (see Puterman [155]) $V_n : \mathcal{S} \mapsto \mathbb{R}^+$. $V_n(x)$ is the minimum expected total costs when there are n events (typically, demands or repairs) left starting in state $x \in \mathcal{S}$. This V_n can be recursively expressed for $n > 0$, starting from V_0.

The key in deriving structural properties of an optimal policy, is the characterization of structural properties, such as convexity and supermodularity, of the value function. For this, we write the value function in so-called *event operators*. We show that the operators of which V_n is composed, all preserve the structural properties. Then, when V_0 satisfies them, it follows directly by induction that the properties hold for V_n for all $n \geq 0$. A framework for this was introduced by Koole [115] (see also Koole [116]) as *Event Based Dynamic Programming*. The main advantage of this approach is that one can prove the propagation of properties for each of the event operators separately. This reduces the complexity of the problem. Also, changes or extensions to the model can easily be made by replacing or adding operators.

As the interarrival times of demands as well as the repair lead times are independent exponentially distributed random variables, we can apply uniformization (cf. Lippman [131]) to convert the semi-Markov decision problem into an equivalent Markov decision problem (MDP). For that, we add fictitious transitions of states to itself, hence ensuring that the total rate out of a state is equal for all states, the so-called uniformization rate. Then, we consider the embedded discrete-time Markov process by looking at the system only at transitions instants, which occur according to a Poisson process, with as rate the uniformization rate.

The existence of a stationary average costs optimal policy is guaranteed by Puterman [155, Theorem 8.4.5a]: as the state space and action space for every state are finite, the costs are bounded and the model is *unichain* and *aperiodic*, there exists a stationary average costs optimal policy. A model is said to be unichain if the transition matrix of every (deterministic) stationary policy is unichain, that is, if it consists of a single recurrent class plus a possibly empty set of transient states. Typically, the considered models are unichain and aperiodic, as the state with fully replenished stocks is accessible from every state in the state space, for every stationary policy, and this state has a positive transition probability to itself. From the structural properties of V_n the optimal policy follows.

In the next section we apply this framework to the stock rationing problem. First we give the structural properties that are used in the sequel of this thesis, and show the general concept along which we prove optimal policy structures.

2.3.2 Structural properties

Let e_i denotes the unit vector of appropriate length, consisting of all zeros except for a 1 at position i. Consider the following properties of a function f, defined for all x such that the states appearing in the right-hand and left-hand side of the inequalities exist in $\mathcal{S}$:

$$\begin{aligned}
\text{Decr}(i): &\quad f(x) \geq f(x+e_i),\\
\text{Incr}(i): &\quad f(x+e_i) \geq f(x),\\
\text{BFOD}(i,c): &\quad f(x+e_i)+c \geq f(x),\\
\text{BFODD}(i,j,c): &\quad f(x+e_i)+c \geq f(x+e_j),\\
\text{Conv}(i): &\quad f(x)+f(x+2\,e_i) \geq 2f(x+2e_i),\\
\text{Supermod}(i,j): &\quad f(x)+f(x+e_i+e_j) \geq f(x+e_i)+f(x+e_j),\\
\text{SuperC}(i,j): &\quad f(x+2\,e_i)+f(x+e_j) \geq f(x+e_i)+f(x+e_i+e_j).
\end{aligned} \tag{2.3.1}$$

$\text{Decr}(i)$ stands for (non-strict) *decreasingness* of f in x_i, $\text{Incr}(i)$ analogously for (non-strict) *increasingness*. $\text{BFOD}(i,c)$ is the abbreviation of *bounded first order difference* and states that the first order difference of f in component i is bounded below by a constant $-c$. Similarly, BFODD is the *bounded first order diagonal difference*. $\text{Conv}(i)$ stands for *convexity* of f in x_i, that is the difference $f(x)-f(x+e_i)$ is decreasing in x_i. Supermod is *supermodularity*, the definition of which is symmetric in i and j. $\text{SuperC}(i,j)$ stands for *superconvexity*, adopting the terminology of [116]. It is a straightforward result that the combination of Supermod and $\text{SuperC}(i,j)$ implies $\text{Conv}(i)$, which follows by adding the respective inequalities and canceling identical terms.

Decr stands for the combination of all $\text{Decr}(i)$'s,

$$\text{Decr} = \bigcap_{i=1}^{m} \text{Decr}(i),$$

where m is the dimensionality of the state space. Similarly,

$$\text{Conv} = \bigcap_{i=1}^{m} \text{Conv}(i), \qquad \text{SuperC} = \bigcap_{\substack{1\leq i,j\leq m\\ i\neq j}} \text{SuperC}(i,j).$$

These definitions are in accordance with the ones in [116].

Multimodularity (MM) (introduced by Hajek [98]) is, for the case of a two-dimensional domain, equal to the combination of Supermod and SuperC:

$$\text{MM} = \text{Supermod} \cap \text{SuperC}. \tag{2.3.2}$$

We use the following notation, cf. [116], for propagation results by an operator: for an operator X we denote by $X: P_1, \ldots, P_N \rightarrow P_1$ that when a function f satisfies properties $P_1, \ldots, P_N$, then Xf satisfies property P_1. Note that the results for the propagation of structural properties by operators contribute to the literature as they can be used in other models as well. Libraries of propagation results are given in [116] and [51].

From the structural properties of V_n, the structure of the optimal policy for this n, say f_n, can be characterized. Typically, such a policy is a threshold type policy, that is, it can be described by one or more so-called switching curves that partition the state space into subsets where a given action is optimal. As this structure holds for all n, by letting n tend to infinity, the structure of the optimal long-run average costs policy follows.

2.4 Dynamic programming formulation

In this section we give the dynamic programming formulation of the stock rationing problem. We present the value function, which consists a.o. of operators for the demands and for the repairs. Then, we indicate which structural properties we use. We prove that the operators the value function consists of, preserve these and that hence the value function satisfies them. From this, we derive the structure of the optimal stock rationing policy, which is a critical level policy.

Recall that x is the on-hand inventory level. Upon a class j demand, a decision has to be taken whether to fulfill it from stock (action 1), or reject it (action 0). The action taken for a class j demand when in state x, is denoted by $a_j(x) \in \{0,1\}$, and an optimal action is denoted by $a_j^*(x)$. The action space is $\mathcal{A}_j(x) = \{0,1\}$ if $x > 0$ and $\mathcal{A}_j(0) = \{0\}$ (as backorders are not allowed). Note that the model is unichain (as the state S is accessible from every state $x \in \mathcal{S}$ for every stationary policy) and aperiodic (since the transition probability from state S to itself is positive).

The value function V_n is given by:

$$V_{n+1}(x) = CU\left(\mu\, G\, V_n(x), \sum_{j=1}^{J} \lambda_j\, H_j\, V_n(x)\right), \text{ for } x \in \mathcal{S}, n \geq 0, \tag{2.4.1}$$

starting with $V_0 \equiv 0$. All operators (C for the costs, U for the uniformization, G for the repairs, H_j for class j demands) are defined below. Decisions are only made in the way of fulfilling demands (in the operator H_j). The decision is taken each time a demand occurs, and it is based on the inventory level. For the repairs, no decisions are taken. Let $\nu = S\,\mu + \sum_{j=1}^{J} \lambda_j$ be the so-called uniformization rate.

The cost operator C is defined by

$$Cf(x) = h(x) + f(x),$$

where $h(x) = \tilde{h}(x)/\nu$ are the holding costs per time unit $1/\nu$.

The uniformization operator U is, for this model, defined by:

$$U(f_1, f_2) = \frac{1}{\nu}\Big(f_1 + f_2\Big). \tag{2.4.2}$$

The event operator G models (potential) repair completions and is defined by

$$Gf(x) = \begin{cases} (S-x)f(x+1) + xf(x), & \text{if } x < S; \\ xf(x), & \text{if } x = S. \end{cases}$$

When the on-hand inventory level is x, there are $S - x$ parts in repair. Hence, with rate $(S-x)\mu$ the stock level x is increased by one. To assure that the rate at which μG occurs is always equal to $S\mu$, we apply uniformization. That is, we add *fictitious transitions*, to let the rates sum to $S\mu$, by adding the term $xf(x)$.

The event operator H_j models the class j demands, and is defined by

$$H_j f(x) = \begin{cases} \min\{f(x-1), c_j + f(x)\}, & \text{if } x > 0, \\ c_j + f(x), & \text{if } x = 0. \end{cases} \tag{2.4.3}$$

If a demand occurs, it has to be decided whether to fulfill it or to reject it. H_j takes the costs-minimizing action, where the costs consist of the direct costs for an action and the expected remaining costs from the state the system is in after taking that action. Either, the demand is satisfied from stock (at no extra costs, decreasing the on-hand stock level by one), or the demand has to be fulfilled by an emergency procedure (at costs c_j, leaving x unchanged). When there is no stock on-hand ($x = 0$), the only option is to reject the demand.

2.5 Structural results

In this section we prove our main result: the structure of the optimal policy. For this we first prove that the value function V_n satisfies two structural properties, by proving that the operators C, G, and H_j, $j = 1, \ldots, J$, preserve them. It then follows that V_n, for all $n \geq 0$, satisfies them. From this we derive the structure of the optimal stock rationing policy, which is a critical level policy.

2.5.1 Properties of operators and value function

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all x such that the states appearing in the right-hand and left-hand side of the inequalities exist in $\mathcal{S}$:

$$\text{BFOD}(c_1): \quad f(x+1) + c_1 \geq f(x), \tag{2.5.1}$$

$$\text{Conv}: \quad f(x) + f(x+2) \geq 2f(x+1), \tag{2.5.2}$$

The next lemma shows that the operators C, G, and H_j preserve these properties.

LEMMA 2.5.1. *a)* $CU\left(G, \sum_{j=1}^{J} \lambda_j H_j\right)$ *: BFOD(c_1) $\to$ BFOD(c_1),*
b) G *: Conv $\to$ Conv.*
c) H_j *: Conv $\to$ Conv, for* $j = 1, \ldots, J$.

The proofs of this lemma and all forthcoming theorems, are given in Appendix 2.A. As C and U take linear combinations, both trivially preserve Conv (noting that the holding costs function $h(\cdot)$ is Conv as well). By induction on n, and using the results of Lemma 2.5.1, the next theorem immediately follows.

THEOREM 2.5.2. *V_n satisfies* (2.5.1) *and* (2.5.2) *for all $n \geq 0$.*

These properties of V_n are the key in classifying the structure of the optimal policy.

2.5.2 Structure of optimal policy

We now characterize the structure of the optimal stock rationing policy.

THEOREM 2.5.3. *There exist $R_1, R_2, \ldots, R_J$, such that for all $j = 1, \ldots, J$ and for all x:*

$$a_j^*(x) = \begin{cases} 1 & \text{if } x > R_j; \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore,

$$0 = R_1 \leq R_2 \leq \ldots \leq R_J. \tag{2.5.3}$$

The described optimal policy is a critical level policy, where the R_j's are the critical levels. A class j demand is only satisfied, if the on-hand stock level is above its critical level. The existence of the critical levels is a consequence of the fact that the value function is Conv, the ordering follows from the ordering in the penalty costs (cf. (2.2.1)), and $R_1 = 0$ follows from BFOD(c_1).

The intuition behind this theorem is as follows. If the on-hand stock level is high, all customer classes are served. When the on-hand stock becomes low, parts are held back from lower priority classes, because the expected costs for not being able to satisfy a coming high priority demand, are higher then the relatively low costs for rejecting this low priority demand. The lower the on-hand stock becomes, the more classes are rejected. Furthermore, as there is no incentive to hold back stock from the highest priority class, clearly $R_1 = 0$, as otherwise there are part(s) kept on stock which are never handed out.

2.A Appendix: Proofs

2.A.1 Proof of Lemma 2.5.1

PROOF. *a)* Assume that f is BFOD(c_1), cf. (2.5.1), then for all $j = 1, \ldots, J$ and for all $x > 0$:

$$\begin{aligned} H_j f(x+1) + c_1 &= \min\{f(x) + c_1, f(x+1) + c_j + c_1\} \\ &\geq \min\{f(x-1),\ f(x) + c_j\} = H_j f(x), \end{aligned} \tag{2.A.1}$$

since f is BFOD(c_1), applied twice. For $x = 0$:

$$H_j f(1) + c_1 = \min\{f(0) + c_1, f(1) + c_j + c_1\} \geq f(0) + c_j = H_j f(0), \tag{2.A.2}$$

by the fact that $c_1 \geq c_j$ for all j (cf. (2.2.1)), and by applying that f is BFOD(c_1). Also, for $x+1 < S$:

$$\begin{aligned}
&Gf(x+1) + S\,c_1 = (S-x-1)f(x+2) + (x+1)f(x+1) + S\,c_1 \\
&= (S-x-1)\big(f(x+2)+c_1\big) + x\big(f(x+1)+c_1\big) + f(x+1) + c_1 \\
&\geq (S-x-1)f(x+1) + xf(x) + f(x+1) + c_1 \\
&= (S-x)f(x+1) + xf(x) + c_1 \\
&\geq (S-x)f(x+1) + xf(x) = Gf(x),
\end{aligned} \tag{2.A.3}$$

since f is BFOD(c_1), and for $x+1 = S$ analogously:

$$\begin{aligned}
&Gf(S) + S\,c_1 \\
&= S\,f(S) + S\,c_1 \\
&= (S-1)\big(f(S)+c_1\big) + f(S) + c_1 \\
&\geq (S-1)f(S-1) + f(S) + c_1 \\
&\geq f(S) + (S-1)f(S-1) = Gf(S-1).
\end{aligned} \tag{2.A.4}$$

Combining these results yields

$$\begin{aligned}
&CU\left(\mu\,G\,f(x+1), \sum_{i=1}^{J}\lambda_j\,H_j\,f(x+1)\right) + c_1 \\
&= h(x+1) + \frac{1}{S\,\mu + \sum_{i=1}^{J}\lambda_j}\left(\mu G\,f(x+1) + \sum_{i=1}^{J}\lambda_j\,H_j\,f(x+1)\right) + c_1 \\
&= h(x+1) + \frac{1}{S\,\mu + \sum_{i=1}^{J}\lambda_j}\left(\mu(G\,f(x+1) + S\,c_1) + \sum_{i=1}^{J}\lambda_j(H_j\,f(x+1) + c_1)\right) \\
&\geq h(x) + \frac{1}{S\,\mu + \sum_{i=1}^{J}\lambda_j}\left(\mu G\,f(x) + \sum_{i=1}^{J}\lambda_j H_j\,f(x)\right) \\
&= CU\left(\mu\,G\,f(x), \sum_{i=1}^{J}\lambda_j\,H_j\,f(x)\right),
\end{aligned}$$

where in the inequality we use (2.A.1) (or (2.A.2) when $x=0$), (2.A.3) (or (2.A.4) when $x+1=S$), and that $h(\cdot)$ is non-decreasing.

b) Assume that f is Conv, cf. (2.5.2), then for all $x+2 < S$:

$$\begin{aligned}
&Gf(x) + Gf(x+2) \\
&= (S-x)f(x+1) + xf(x) + (S-x-2)f(x+3) + (x+2)f(x+2) \\
&= (S-x-2)\big(f(x+1) + f(x+3)\big) + 2f(x+1) \\
&\qquad + x\big(f(x) + f(x+2)\big) + 2f(x+2) \\
&\geq 2(S-x-2)f(x+2) + 2f(x+1) \\
&\qquad + 2\,xf(x+1) + 2f(x+2) \\
&= 2(S-x-1)f(x+2) + 2(x+1)f(x+1) \\
&= 2\,Gf(x+1),
\end{aligned}$$

since f is Conv, and analogously, for $x+2=S$:

$$\begin{aligned}
&Gf(S-2)+Gf(S)\\
&=2f(S-1)+(S-2)f(S-2)+Sf(S)\\
&=2f(S-1)+(S-2)\big(f(S-2)+f(S)\big)+2f(S)\\
&\geq 2f(S-1)+2(S-2)f(S-1)+2f(S)\\
&=2f(S)+2(S-1)f(S-1)\\
&=2Gf(S-1).
\end{aligned}$$

c) Assume that f is Conv, cf. (2.5.2), then for all $j=1,\ldots,J$ and for all $x>0$:

$$\begin{aligned}
&H_jf(x)+H_jf(x+2)\\
&=\min\begin{cases} f(x-1)+f(x+1)\\ f(x-1)+f(x+2)+c_j\\ f(x)+c_j+f(x+1)\\ f(x)+c_j+f(x+2)+c_j\end{cases} \qquad (*)
\end{aligned}$$

Now note that, since f is Conv:

$$\begin{aligned}
f(x-1)+f(x+1)&\geq 2f(x),\\
f(x-1)+f(x+2)+c_j&\geq f(x-1)+2f(x+1)-f(x)+c_j\\
&\geq f(x)+f(x+1)+c_j,\\
f(x)+c_j+f(x+2)+c_j&\geq 2(f(x+1)+c_j).
\end{aligned}$$

Then, continuing from $(*)$:

$$\begin{aligned}
(*)&\geq\min\begin{cases} 2f(x)\\ f(x)+f(x+1)+c_j\\ f(x)+c_j+f(x+1)\\ 2(f(x+1)+c_j)\end{cases}\\
&\geq 2\min\{f(x),f(x+1)+c_j\}=2H_jf(x+1).
\end{aligned}$$

For $x=0$ analogously:

$$\begin{aligned}
&H_jf(0)+H_jf(2)\\
&=\min\{f(0)+c_j+f(1),\ f(0)+c_j+f(2)+c_j\geq 2(f(1)+c_j)\}\\
&\geq 2\min\{f(0),f(1)+c_j\}=2H_jf(1).
\end{aligned}$$

□

2.A.2 Proof of Theorem 2.5.3

PROOF. Consider a class j demand. For $x\in\mathcal{S}$, $u\in\{0,1\}$, and $n\geq 0$, define

$$w^{(n)}(u,x):=\begin{cases} V_n(x-1) & \text{if } u=1,\\ V_n(x)+c_j & \text{if } u=0,\end{cases} \tag{2.A.5}$$

defining $V_n(-1) := \infty$. Hence $H_j V_n(x) = \min_{u\in\{0,1\}} w^{(n)}(u,x)$. Define, for $u \in \{0,1\}$, $n \geq 0$, and $x \in \{0,1,\dots S-1\}$:

$$\Delta w_x^{(n)}(u,x) := w^{(n)}(u,x+1) - w^{(n)}(u,x).$$

Then for each $n \geq 0$:

$$\begin{aligned}
&\Delta w_x^{(n)}(1,x) - \Delta w_x^{(n)}(0,x) \\
&= w^{(n)}(1,x+1) - w^{(n)}(1,x) - w^{(n)}(0,x+1) + w^{(n)}(0,x) \\
&= V_n(x) - V_n(x-1) - (V_n(x+1) + c_j) + V_n(x) + c_j \\
&= 2\,V_n(x) - V_n(x-1) - V_n(x+1) \leq 0,
\end{aligned}$$

as, by Theorem 2.5.2, V_n is Conv. So, $\Delta w_x^{(n)}(u,x)$ is decreasing in u:

$$\Delta w_x^{(n)}(1,x) \leq \Delta w_x^{(n)}(0,x).$$

This implies that, for every $n \geq 0$, there exists a threshold for the inventory level x, say $R_{j,n}$, from which on it is optimal to satisfy a demand from stock. As this holds for all $j = 1,\dots,J$, if f_{n+1} is the minimizing policy in (2.4.1), then f_{n+1} is a critical level policy. Note that the transition probability matrix of every stationary policy is unichain (since every state can access S, i.e. there exists a path with positive probability from every state to state S) and aperiodic (since the transition probability from state S to itself is positive). Then, by Puterman [155, Theorem 8.5.4], the long run average costs under the stationary policy f_{n+1} converges to the minimal long run average costs as n tends to infinity. Since there are only finitely many stationary critical level policies, this implies that there exists an optimal stationary policy that is a critical level policy, with critical levels, say R_j.

By the ordering of the c_j's, cf. (2.2.1), it follows that when $f(x) \leq f(x+1) + c_j$ also $f(x) \leq f(x+1) + c_{j+1}$, for all $j = 1,\dots,J-1$. Hence, $R_{j+1} \geq R_j$. From BFOD(c_1), applied for $x = 0$, if follows that it is optimal to satisfy a class 1 demand in state $x = 1$. So, the critical level for a class 1 demand is zero: $R_1 = 0$. □

3

Multi-location inventory models with a quick response warehouse

We study a multi-location inventory problem with a so-called quick response warehouse. In case of a stock-out at a local warehouse, a demand can be satisfied by a stock transfer from the quick response warehouse. We derive the long-run average costs optimal policy for when to apply such a stock transfer, as well as conditions under which that is always optimal. In a numerical study we compare the performance of the optimal policy to that of simpler policies. Furthermore, we study model variations.

3.1 Introduction

In this chapter we study a multi-location inventory model, with the special feature of a so-called *Quick Response* (QR) warehouse. When a local warehouse is out-of-stock, a demand can be satisfied by a stock transfer from this QR warehouse. This QR warehouse is situated at close distance to the local warehouses. Hence, by a stock transfer the demand is satisfied much faster compared to an emergency shipment from a central warehouse (or from outside the network).

A relevant application of this is found in spare parts inventory networks, where ready-for-use parts are kept on stock for critical components of advanced technical systems. Examples of these include the key manufacturing machines in a production line, trucks for a transportation company, and expensive medical equipment in a hospital. Upon breakdown of such a system, it demands a spare part. During this time, the system is down at very high costs because of loss of production/revenue. So, in order to reduce down time, it is important that demand is quickly satisfied, and a quick response warehouse is a good option for doing so. Axsäter et al. [14] and Howard et al. [105] describe the setting at Volvo Parts Corporation, a global spare parts service provider, which makes use of QR warehouses (referred to as 'support warehouses'). Rijk [161] studies the stock control of Océ, a company in printing and document management. They use quick response stocks for storing parts that need to be within a short time range of the customers, but for which it is not possible or efficient to store these in the car stocks of the maintenance engineers.

Another application of the model with a QR warehouse is the combination of physical stores and an on-line shop, e.g. for books or fashion. Next to the physical stores, the on-line shop keeps items in inventory as well, located centrally. The demands of customers visiting the physical stores are satisfied immediately when there is stock on-hand, but their demands can be routed to the inventory of the on-line shop in case of a stock-out at the store. Hence, this on-line shop, which also has its own demand stream, acts as a QR warehouse.

Although a relevant problem, to the best of our knowledge, no results are known about the optimal use of a QR warehouse. For that, we study the policy for when the QR warehouse should accept and when it should reject a demand originating at a local warehouse. All warehouses are assumed to follow a base stock policy, with given base stock levels. We formulate the problem as a Markov decision problem (MDP) and use event-based dynamic programming (cf. [116]) to derive the optimal policy structure of the QR warehouse, minimizing the long-run average costs. Also, we derive simple, sufficient conditions under which it is always optimal to accept a demand at the QR warehouse. The analysis builds on the assumption of exponentially distributed replenishment lead times. This implies that the decisions at the QR warehouse are based on the net stock levels at all warehouses, as pipeline information can be ignored. It is known that then performance of the optimal policy can be expected to be nearly insensitive to the distribution of the replenishment times, as shown in a.o. [2, 69, 120] for similar models.

In inventory models, shipments of stock between warehouses of the same echelon are referred to as lateral transshipments (LTs), see [153] for an overview. Typically, either the optimal policy is derived for a two-location model [6, 218], or for a multi-location setting with symmetric parameters for all warehouses [162]. Results for two-location models cannot be extended by the techniques used to results for general multi-location models. We, however, do so by considering a model in which only lateral transshipments from the QR warehouse to the local warehouses are allowed.

Basically, our model is a combination of a so-called *overflow model* and a *stock rationing model*. In overflow models, unsatisfied demands are routed to another source. These models typically arise in telecommunication models, e.g. in call centers (see [87] for an overview). However, in these models, no costs are incorporated for the routing or blocking of demands, whereas we show these costs to play an important role in the optimal policy when incorporated. In stock rationing models (see [180] and the references therein), multiple demand classes are served from a single stock point. If the arrival processes are Poisson, the optimal policy for satisfying demands is a so-called critical level policy (see e.g. [183, 94, 60]), which prescribes a (net) stock level (the critical level) for each demand class from which on their demands are satisfied. However, the overflow demand streams in our model are *not* Poisson processes and, as a consequence, such a critical level policy fails to be optimal. In fact, an overflow demand stream is a special case of a Markov modulated Poisson process (MMPP, cf. [80]). The optimal policy depends on the states of each of these processes, i.e. on the stock levels at all local warehouses.

The outline of this chapter is as follows. We start by describing the model and introducing the notation in Section 3.2. We formulate the problem as an MDP and introduce the value function. In Section 3.3 we show that the value function satisfies certain structural results. From this the optimal policy at the QR warehouse is derived, as well as the simplifying conditions. Section 3.4 shows numerical results on how much cost savings are achieved by executing the optimal policy. Two model variations are discussed in Section 3.5, and we end by a discussion on further research in Section 3.6. All proofs are given in Appendix 3.A. This chapter is based on [194].

3.2 Model and notation

3.2.1 Problem description

We consider the following multi-location inventory model. We have J local warehouses, $j = 1, \ldots, J$, and a quick response (QR) warehouse with index $j = 0$, keeping on stock a single stock-keeping-unit. Warehouse j follows a base stock policy with base stock level S_j, $j = 0, 1, \ldots, J$, where $S_0 \geq 1$ to avoid trivialities ($S_0 = 0$ would be equivalent to a situation without a QR warehouse). All warehouses $j = 0, 1, \ldots, J$ are replenished from a central warehouse with infinite stock (or equivalently, from an external supplier outside the network), these being one-for-one and having i.i.d. exponentially distributed lead times with mean $1/\mu_j$ for warehouse j. The replenishments can also be interpreted as productions to stock, or repair procedures of repairables. Warehouse j faces a demand stream that is Poisson with rate λ_j, $j = 0, 1, \ldots, J$. We assume the interarrival and replenishment times to be all mutually independent. Holding costs are incurred for on-hand inventory: $h_j(x_j)$ are the costs for keeping x_j parts in stock at location j during one time unit. The function $h_0(\cdot)$ is assumed to be convex (it appears that this is the only property needed for the structural results derived below).

When *local* warehouse j is out-of-stock, the demand can be fulfilled by a stock transfer from the QR warehouse, at costs P_j^{QR}. This is referred to as an *overflow demand* of warehouse j. In this case a part from the QR warehouse is directly assigned to this demand, and shipped to local warehouse j. Hence, the demand and part are instantaneously coupled. We assume this procedure to be much faster than waiting for a regular replenishment. When the demand is not fulfilled from the QR warehouse, it has to be fulfilled by a costly emergency procedure, at penalty costs P_j^{EP}, e.g. by a shipment from the central warehouse or an external supplier (equivalently, this can be interpreted as a lost sale). Demands that occur at the QR warehouse itself, are either satisfied directly, or fulfilled by an emergency procedure at penalty costs P_0^{EP}. To avoid trivialities, we assume that $0 \leq P_j^{QR} \leq P_j^{EP}$ for all j, and define $\Delta P_j = P_j^{EP} - P_j^{QR}$. For ease of notation, we define $P_0^{QR} = 0$ and hence $\Delta P_0 = P_0^{EP}$. This inventory model is graphically presented in Figure 3.1. The question is when the QR warehouse should satisfy an (overflow) demand, and when it is better to reject it, based on the stock levels at all warehouses.

The motivation for this setting is an inventory system which provides spare parts for advanced technical systems. These systems are typically used in the primary processes of their users. Hence, any down time of these systems is extremely costly, so ready-for-use spare parts are kept in stock for the critical component of these systems. Upon failure of a system, the defective part is replaced by a ready-for-use part from inventory, from the warehouse the technical system is assigned to. We assume the penalty costs P_j^{QR} and P_j^{EP} to include the down time costs (e.g. because of loss of production) of a machine during the time required for the quick response and emergency procedure, respectively. Note, however, that the model may also apply to other multi-location (or multi-product) inventory systems; see also [153].

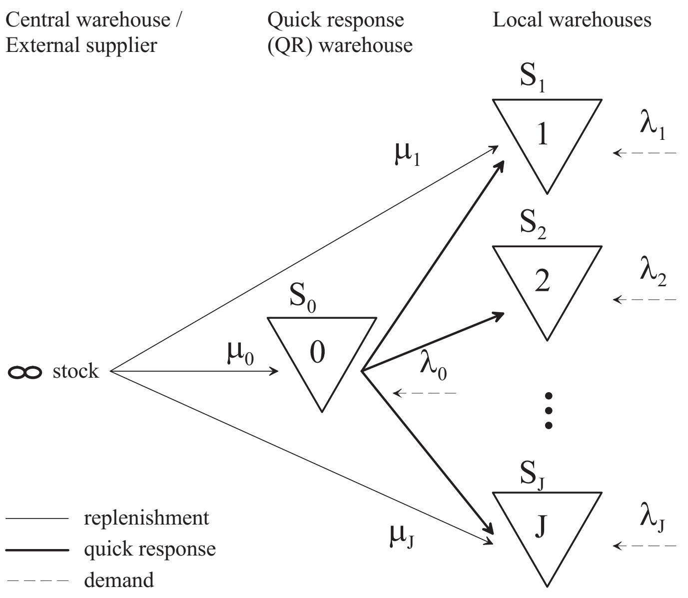


Figure 3.1: Multi-location inventory model with a Quick Response warehouse.

3.2.2 Dynamic programming formulation

We formulate the problem as a Markov decision problem (MDP, cf. [155]) and use event-based dynamic programming (cf. [116]) to derive the optimal policy structure, minimizing the long-run average costs. The existence of such an optimal policy is guaranteed by [155, Theorem 8.4.5a].

Let x_j be the stock level of location j, and let $x = (x_0, x_1, \ldots, x_J)$ be the vector consisting of all stock levels. So, x is the state of the system, on the state space $\mathcal{S}$ consisting of all possible combinations of stock levels. Let $V_n : \mathcal{S} \mapsto \mathbb{R}$ be the *value function*, the minimum cost function when there are n events (demands or replenishments) left. It is given by:

$$V_{n+1}(x) = \frac{1}{\nu}\left(\sum_{j=0}^{J} h_j(x_j) + \sum_{j=0}^{J} \mu_j G_j V_n(x) + \sum_{j=1}^{J} \lambda_j H_j V_n(x) + \lambda_0 H_{QR} V_n(x)\right), \quad (3.2.1)$$

for $x \in \mathcal{S}$, $n \geq 0$, starting with $V_0 \equiv 0$, where $\nu = \sum_{j=0}^{J} S_j \mu_j + \sum_{j=0}^{J} \lambda_j$ is the uniformization rate, and G_j, H_j, and H_{QR} as defined below.

The operator H_j models the demands at local warehouse $j = 1, \ldots, J$, and is defined by:

$$H_j f(x) = \begin{cases} f(x - e_j) & \text{if } x_j > 0; \\ \min\{P_j^{QR} + f(x - e_0), & \\ \qquad P_j^{EP} + f(x)\} & \text{if } x_j = 0, x_0 > 0; \\ P_j^{EP} + f(x) & \text{otherwise.} \end{cases}$$

Here e_j is the unit vector of length $J + 1$ with a 1 at position j ($j = 0, 1, \ldots, J$). When

$x_j > 0$ a part is taken from stock. When $x_j = 0$ (and $x_0 > 0$), H_j selects the costs–minimizing action of taking a part from the QR warehouse (at costs P_j^{QR}), or applying an emergency procedure (at costs P_j^{EP}). If both the local and the QR warehouse are out-of-stock, the only option is an emergency procedure.

Similarly, H_{QR} models the demands at QR warehouse, and is defined by:

$$H_{QR} f(x) = \begin{cases} \min\{f(x - e_0), P_0^{EP} + f(x)\} & \text{if } x_0 > 0; \\ P_0^{EP} + f(x) & \text{if } x_0 = 0. \end{cases}$$

The operator G_j models the (potential) replenishments at warehouse $j = 0, 1, \ldots, J$, and is defined by:

$$G_j f(x) = \begin{cases} (S_j - x_j) f(x + e_j) + x_j f(x) & \text{if } x_j < S_j; \\ S_j f(x) & \text{if } x_j = S_j. \end{cases}$$

The replenishment rate is linear in the number of outstanding orders, which is $S_j - x_j$. The terms $x_j f(x)$ represent fictitious transitions, hence assuring that the total rate at which $\mu_j G_j$ occurs is $\mu_j S_j$, for all x. In Section 3.5.1 we consider state-dependent replenishment rates as a model variation.

3.3 Structural results

In this section we prove our main result: the structure of the optimal policy of the QR warehouse. For this, we first introduce the properties convexity and supermodularity. Each of the operators in the value function preserves these properties, hence the value function satisfies them. From this, the optimal policy structure is derived, as well as conditions under which it simplifies.

3.3.1 Structural properties

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all x such that the states appearing in the right–hand and left–hand side of the inequalities exist in the state space $\mathcal{S}$:

$$\begin{aligned} \text{Conv}(x_0): \quad & f(x) + f(x + 2e_0) \geq 2f(x + e_0), \\ \text{Supermod}(x_0, x_k): \quad & f(x) + f(x + e_0 + e_k) \geq f(x + e_0) + f(x + e_k), \quad \text{for } k \neq 0. \end{aligned}$$

LEMMA 3.3.1. *The operators (i) H_j, (ii) H_{QR}, and (iii) G_j preserve, for all $j = 1, \ldots, J$, the properties Conv(x_0) and Supermod(x_0, x_k), for all $k = 1, \ldots, J$.*

Note that $V_0 \equiv 0$ trivially is Conv(x_0) and Supermod(x_0, x_k) for all $k = 1, \ldots, J$. Hence, by induction and using the results of Lemma 3.3.1, the next result directly follows.

THEOREM 3.3.2. *For all $n \geq 0$, V_n is Conv(x_0) and Supermod(x_0, x_k) for all $k = 1, \ldots, J$.*

Note that $V_0 \equiv 0$ trivially is Conv(x_0) and Supermod(x_0, x_k) for all $k = 1, \ldots, J$. Hence, by induction and using the results of Lemma 3.3.1, the next result directly follows.

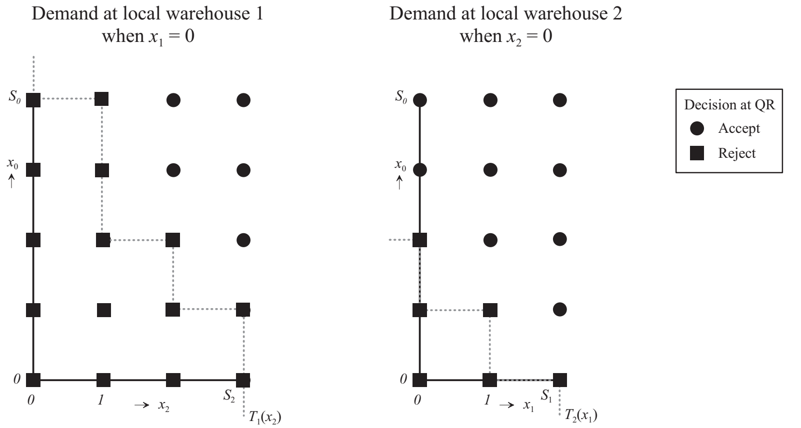


Figure 3.2: Optimal policy structure when $J = 2$ for demands at the local warehouses $j = 1, 2$ when $x_j = 0$.

3.3.2 Structure of optimal policy

The following theorem describes the structure of the optimal policy at the QR warehouse, characterizing when to apply a stock transfer (accepting the demand), or not (rejecting the demand). For this, we denote by $x^{(0,j)}$, $j = 1, \ldots, J$, the vector x without the component x_0 and with $x_j = 0$. That is, $x^{(0,j)} := (x_1, \ldots, x_{j-1}, 0, x_{j+1}, \ldots, x_J)$. Further, $x^{(0,0)} := (x_1, \ldots, x_J)$.

THEOREM 3.3.3. *The optimal policy at the QR warehouse is a state-dependent threshold policy. That is, for all $j = 0, 1, \ldots, J$ there exists a switching curve $T_j(x^{(0,j)})$ that characterizes the optimal decision for a demand at the QR warehouse (i.e., $j = 0$), or an overflow demand from local warehouse j when $x_j = 0$ (for $j = 1, \ldots, J$):*

- *if* $x_0 > T_j(x^{(0,j)})$: *accept;*
- *if* $x_0 \leq T_j(x^{(0,j)})$: *reject.*

$T_j(x^{(0,j)})$ is decreasing in each component of $x^{(0,j)}$.

The proof makes use of the fact that the value function is Conv(x_0) and Supermod(x_0, x_j). Figure 3.2 shows the optimal policy structure for demands at the local warehouses when $J = 2$. It shows that a demand is more likely to be accepted when the QR stock level is high, and/or when the stock levels at the other local warehouses are high. This is in line with the fact that $T_j(x^{(0,j)})$ is decreasing.

We have the following ordering in the optimal actions. Let $j_1, j_2 \in \{1, \ldots, J\}$ with $\Delta P_{j_1} \geq \Delta P_{j_2}$, i.e. the cost difference ΔP_{j_1} at local warehouse j_1 is larger than or equal to the cost difference ΔP_{j_2} at local warehouse j_2. Then their optimal actions for applying a quick response are ordered accordingly. More precisely, denoting by $a_j^*(x)$ the optimal action at local warehouse j in case of a demand, encoding $a_j^*(x) = 1$ for a quick response (i.e. accepting the demand) and $a_j^*(x) = 0$ for an emergency procedure (rejecting the demand), then the following holds.

PROPOSITION 3.3.4. *Let $j_1, j_2 \in \{1,\ldots,J\}$ with $\Delta P_{j_1} \geq \Delta P_{j_2}$. Then $a^*_{j_1}(x) \geq a^*_{j_2}(x)$ for all x with $x_{j_1} = x_{j_2} = 0$.*

When the base stock levels at all local warehouses equal zero, the overflow demand stream from each of the local warehouses is a Poisson process with rate λ_j. By Theorem 3.3.3 the switching curve T_j is a function of $x^{(0,j)}$, however, there is only *one* such vector, namely the all zero vector. Hence, in this case, the switching curve T_j reduces to a constant, say $C_j \in \{0,1,\ldots,S_0\}$, for all j. An (overflow) demand is satisfied when $x_0 > C_j$, and is rejected otherwise. This state-*in*dependent threshold policy is known as a *critical level policy* [183], where the C_j's are called the critical levels. It is known to be optimal in this setting for Poisson demand streams, cf. [94]. So, we have proven this result as a special case of our model. Moreover, Proposition 3.3.4 shows that the critical levels are ordered based on the ΔP_j, that is, if $\Delta P_1 \geq \Delta P_2 \geq \ldots \geq \Delta P_J$, then $C_1 \geq C_2 \geq \ldots \geq C_J$, as in [94]. This model, where inventory at a single warehouse is allocated to multiple customers classes with different costs factors ΔP_1, is known as a stock rationing problem.

An overflow demand stream from local warehouse j is a special case of a Markov modulated Poisson process (MMPP [80]): one with S_j states, demand rate λ_j at the QR warehouse when in state $x_j = 0$, and zero otherwise, where the transition rates follow from the replenishment rate and (local) demand rate. Hence, Theorem 3.3.3 gives the optimal policy for a stock rationing problem with this form of MMPP demand streams, showing that a state-dependent threshold policy is optimal in this case.

3.3.3 Conditions

The following theorem provides conditions under which a simpler policy is always optimal. Only the holding cost of the last part at the QR warehouse turn out to be of importance, hence define $\Delta h_0 = h_0(1) - h_0(0)$.

THEOREM 3.3.5. *It is optimal to always accept a demand from local warehouse j at the QR warehouse (for $j = 1,\ldots,J$, when $x_0 > 0$ and $x_j = 0$), or a demand at the QR warehouse (for $j = 0$, when $x_0 > 0$) if:*

$$\sum_{k=0}^{J} \lambda_k \Big(\Delta P_k - \Delta P_j\Big)^+ \;\leq\; \mu_0 \Delta P_j + \Delta h_0, \tag{3.3.1}$$

where $(x)^+ = \max\{x, 0\}$.

Recall that $\Delta P_k = P_k^{EP} - P_k^{QR}$, $k = 1,\ldots,J$, and $\Delta P_0 = P_0^{EP}$. Furthermore, note that the larger μ_0 (i.e., the shorter the replenishment lead time), the easier the condition is satisfied. This effect is obtained because a larger μ_0 gives a lower risk of not being able to meet a more important demand in the near future. Basically, these conditions give a trade-off between the cost parameters. It is optimal to always satisfy *any* demand at the QR warehouse, if (3.3.1) holds for all $j = 0,1,\ldots,J$.

3.4 Numerical results

In a numerical study we show how much is to be gained by executing the optimal policy, compared to two simpler policies. For two examples, we vary the arrival rates and cost parameters, and compare the average costs per time unit of executing three possible policies: (i) the optimal policy, (ii) a naive policy always satisfying all demands, and (ii) a state-*in*dependent threshold policy with optimal thresholds, the so-called optimal critical level policy.

In a critical level policy, for each warehouse a critical level is prescribed, say C_j for warehouse $j = 0, 1, \ldots, J$ where $C_j \in \{1, \ldots, S_0\}$. Only when the inventory level at the QR warehouse is above this level, an (overflow) demand from warehouse j is satisfied. The critical level is a fixed constant which does not depend on the state of the system. By an exhaustive search we optimize the vector $(C_0, C_1, \ldots, C_J)$. Note that at least one critical level will equal zero, as otherwise at least one part of the stock at the QR warehouse remains untouched in any case.

We consider two examples, both with three local warehouses and all base stock levels equal to 3. All replenishment rates equal $\mu_j = 1$ and all holdings costs are 0. The emergency costs are $P_0^{EP} = 10$, $P_1^{EP} = 50$, $P_2^{EP} = 20$, and $P_3^{EP} = 10$. We specify the quick response costs by setting the ratio P_j^{QR}/P_j^{EP}, taking values in $\{0.1, 0.5, 0.9\}$. In Example 1, the QR warehouse is facing *no* direct demand stream: $\lambda_0 = 0$. Furthermore, we vary λ_1 and $\lambda_2 = \lambda_3 = 2.9$. In Example 2, λ_0 is positive: $\lambda_0 = \lambda_2 = \lambda_3 = 1.7$, and again we vary λ_1. For finding the optimal critical level policy, we optimize over 37 possible critical level policies in Example 1, and 175 in Example 2.

We calculate the relative extra costs per time unit for executing the naive and optimal critical level policy compared to the optimal policy. The results are given in Table 3.1. We see that a naive policy can be considerably worse than the optimal policy. The highest relative cost difference (7.79 %) is obtained for a situation with high demand rates (relative to the base stock levels) and relatively low quick response costs (relative to the emergency costs). Then it is important to hold back stock at the QR warehouse for demands from the local warehouses 1 and 2. In the same situation, the optimal critical level policy has a much smaller relative cost difference (2.34 %), but this is still a significant difference. Apparently, looking at the whole state when holding back stock leads to significantly better results than just looking at the stock level at the QR warehouse. The highest relative cost difference for the optimal critical level policy (4.29 %) is obtained in another situation with high demand rates and low quick response costs.

3.5 Model variations

In this section we study two model variations, namely stock level dependent replenishment rates and backlogging at the local warehouses.

3.5.1 Stock level dependent replenishment rates

When the stock level at warehouse j ($j = 0, 1, \ldots, J$) equals x_j, there are $y_j := S_j - x_j$ outstanding orders, where $0 \le y_j \le S_j$. In the preceding we assumed that the replenishment rate at this warehouse is $y_j \mu_j$. We now investigate the case of stock level dependent replenishment rates. That is, the replenishment rate is given by $\phi_j : \{0, 1, \ldots, S_j\} \mapsto \mathbb{R}^+$, where $\phi_j(0) = 0$ and furthermore $\phi_j(y_j)$ is assumed to be a concave, increasing function of y_j. Hence, its maximum is attained in S_j, so $\max_{y_j \in \{0,1,\ldots,S_j\}} \phi_j(y_j) = \phi_j(S_j) =: \overline{\phi}_j$, assuming $\overline{\phi}_j < \infty$.

$P_j^{QR}:P_j^{EP}$	λ_1	Opt	Naive	vs. Opt	Opt CL	vs. Opt	Opt CLs	QR (Opt)	EP (Opt)
0.1	1.5	14.95	15.30	+ 2.34 %	15.30	+ 2.34 %	{0,0,0}	19.8 %	9.5 %
	2.2	22.34	23.44	+ 4.93 %	22.73	+ 1.72 %	{0,0,1}	18.5 %	12.3 %
	2.2	33.34	35.94	+ 7.79 %	34.13	+ 2.35 %	{0,0,2}	18.0 %	15.4 %
0.5	1.5	26.10	26.30	+ 0.74 %	26.30	+ 0.74 %	{0,0,0}	19.7 %	9.6 %
	2.2	37.50	38.11	+ 1.63 %	37.71	+ 0.57 %	{0,0,1}	18.5 %	12.4 %
	2.2	53.39	54.81	+ 2.66 %	53.88	+ 0.91 %	{0,0,1}	18.0 %	15.3 %
0.9	1.5	37.26	37.30	+ 0.10 %	37.30	+ 0.10 %	{0,0,0}	19.8 %	9.5 %
	2.2	52.64	52.76	+ 0.23 %	52.69	+ 0.08 %	{0,0,1}	18.5 %	12.2 %
	2.9	73.36	73.64	+ 0.39 %	73.46	+ 0.13 %	{0,0,1}	18.0 %	15.4 %

(a) Example 1

$P_j^{QR}:P_j^{EP}$	λ_1	Opt	Naive	vs. Opt	Opt CL	vs. Opt	Opt CLs	QR (Opt)	EP (Opt)
0.1	0.7	8.53	8.54	+ 0.11 %	8.54	+ 0.11 %	{0,0,0,0}	6.7 %	10.4 %
	1.2	10.48	10.65	+ 1.62 %	10.65	+ 1.62 %	{0,0,0,0}	6.9 %	10.7 %
	1.7	14.45	15.07	+ 4.29 %	15.07	+ 4.29 %	{0,0,0,0}	7.7 %	11.5 %
0.5	0.7	11.09	11.13	+ 0.39 %	11.09	+ 0.01 %	{0,0,0,1}	5.4 %	11.2 %
	1.2	14.31	14.39	+ 0.58 %	14.31	+ 0.02 %	{0,0,0,1}	5.8 %	11.1 %
	1.7	20.71	20.87	+ 0.78 %	20.73	+ 0.06 %	{0,0,0,1}	6.7 %	12.0 %
0.9	0.7	12.94	13.72	+ 6.04 %	12.95	+ 0.02 %	{0,1,2,3}	0.9 %	14.0 %
	1.2	17.34	18.13	+ 4.59 %	17.34	+ 0.01 %	{0,0,2,3}	1.8 %	13.8 %
	1.7	25.86	26.68	+ 3.16 %	25.87	+ 0.01 %	{0,0,2,3}	3.4 %	14.2 %

(b) Example 2

Table 3.1: Average costs of the optimal policy, the naive policy (always satisfying all demands), and the optimal critical level policy, as well as the relative difference of the last two policies compared to the optimal policy. Also given are the optimal critical levels, as well as the fraction of the demand satisfied via a quick response and via an emergency procedure under the optimal policy.

The replenishment operator, say $\tilde{G}_j$, now is given by:

$$\tilde{G}_j f(x) = \begin{cases} \phi_j(y_j) f(x+e_j) + (\overline{\phi}_j - \phi_j(y_j)) f(x) & \text{if } x_j < S_j \ (\text{if } y_j > 0) \\ \overline{\phi}_j f(x) & \text{if } x_j = S_j \ (\text{if } y_j = 0). \end{cases}$$

for $j = 0, 1, \ldots, J$. The rate out of each state because of $\tilde{G}_j$ is equal to $\overline{\phi}_j$. Analogously to Lemma 3.3.1(iii), the following results hold.

LEMMA 3.5.1. *The operator $\tilde{G}_j$ preserves, for all $j = 1, \ldots, J$, the properties Decr(x_0), Conv(x_0), and Supermod(x_0, x_k), for all $k = 1, \ldots, J$.*

Here Decr(x_0) stands for (non-strict) decreasingness in x_0, that is: $f(x) \geq f(x+e_0)$ (as introduced in Section 2.3.2).

Example 3.5.1. An example of a stock level dependent replenishment rate where $\phi(.)$ is increasing and concave, is a multi-server model with T servers. Each server processes a replenishment at rate μ. So, the replenishment rate is linear in y, namely $y\mu$, with maximum rate $T\mu$:

$$\phi(y) = \begin{cases} y\mu & \text{if } 0 \leq y < T, \\ T\mu & \text{if } T \leq y \leq S. \end{cases}$$

Special cases are $T = 1$ (single server) and $T = S$ (ample repair capacity, as in the current model). This might also be an appropriate model when T machines are producing (i.e. replenishing) to stock, with exponentially distributed production lead times.

As we need Decr(x_0) in order for $\tilde{G}_j$ to preserve convexity, we cannot include holding costs anymore in the QR warehouse (as these are increasing in x_0). So, the new value function, say $\tilde{V}_n$, becomes:

$$\tilde{V}_{n+1}(x) = \sum_{j=1}^{J} h_j(x_j) + \frac{1}{\tilde{\nu}} \left(\sum_{j=0}^{J} \tilde{G}_j \tilde{V}_n(x) + \sum_{i=1}^{J} \lambda_j H_j \tilde{V}_n(x) \right), \text{ for } x \in \mathcal{S}, n \geq 0,$$

starting again with e.g. $\tilde{V}_0 \equiv 0$, where now $\tilde{\nu} = \sum_{j=0}^{J} \overline{\phi}_j + \sum_{j=1}^{J} \lambda_j$ is the uniformization rate.

As a consequence of Lemma 3.5.1, we have, like in Theorem 3.3.2, that $\tilde{V}_n$ is Decr(x_0), Conv(x_0) and Supermod(x_0, x_k) for all $j = 1, \ldots, J$, when $\tilde{V}_0$ satisfies these properties. Hence, Theorem 3.3.3 remains to hold. Theorem 3.3.5 remains valid when μ_0 in (3.3.1) is replaced by $\mu_0(S_0) - \mu_0(S_0 - 1)$:

$$\sum_{k=0}^{J} \lambda_k \left(\Delta P_k - \Delta P_j \right)^+ \leq \left(\mu_0(S_0) - \mu_0(S_0 - 1) \right) \Delta P_j + \Delta h_0. \tag{3.5.1}$$

Again, it is optimal to always satisfy *any* demand at the QR warehouse, if (3.5.1) holds for all $j = 0, 1, \ldots, J$.

3.5.2 Backlogging at local warehouses

In Ching [49] an approximate evaluation is given for a model that is almost identical to our model as described in Section 3.2. Ching allows backlogging at the local warehouse, up to a (finite) maximum B_j. Only when this maximum is reached, a demand from a local warehouse flows over to the QR warehouse. He assumes that such a demand is always satisfied at the QR warehouse.

Instead of the stock level x_j we now focus on the stock level *plus* the maximum number of outstanding backorders B_j, that is: $x_j^{(b)} := x_j + B_j$. Taking the vector $(x_0, x_1^{(b)}, \ldots, x_J^{(b)})$ as the state of the system, we are now back at the original model, however, with a stock level dependent replenishment rate (cf. Section 3.5.1) at each of the local warehouses. The replenishment rate is given by:

$$\phi_j(x_j^{(b)}) = \begin{cases} (S_j - x_j^{(b)} + B_j)\mu_j & \text{if } x_j^{(b)} > B_j, \\ S_j \mu_j & \text{if } x_j^{(b)} \leq B_j. \end{cases}$$

Hence, the results of Section 3.5.1 apply to this model as well. As a consequence, when (3.5.1) holds for all $j = 0, 1, \ldots, J$, always accepting any (overflow) demand at the QR warehouse, the policy assumed in [49], is optimal in this setting. Even if we charge backlog costs per outstanding backorder per time unit (adding the term $\sum_{j=1}^{J} b_j(\max(0, -x_j))$ where $b_j(\cdot)$ is a non-increasing, convex function with $b_j(0) = 0$), the given structural results and conditions remain valid.

3.6 Further research

It would be interesting for further research to study how well the sufficient condition (3.3.1) covers the parameter settings under which all overflow demands from warehouse j are accepted at the QR warehouse under the optimal policy. Moreover, an interesting question is whether the same structural results of the optimal policy hold for more general arrival processes at the QR warehouse. When the overflow demand streams at the QR warehouse are Poisson processes, the optimal policy is known to be state-independent threshold policy. We generalized this by letting the demand processes be the overflow streams of the local warehouses, hence being a special form of Markov modulated Poisson processes (MMPPs). The question is whether this can be generalized even further, to more general MMPPs or Markov arrival processes (MAPs).

3.A Appendix: Proofs

3.A.1 Proof of Lemma 3.3.1

PROOF. (i) For all $j = 1, \dots, J$ the following holds.
• $H_j : \text{Conv}(x_0) \rightarrow \text{Conv}(x_0)$.
Assume that f is $\text{Conv}(x_0)$, then we show that $H_j f$ is $\text{Conv}(x_0)$ as well. For $x_j > 0$ we have:

$$H_j f(x) + H_j f(x + 2e_0) = f(x - e_j) + f(x + 2e_0 - e_j) \geq 2f(x + e_0 - e_j) = 2H_j f(x + e_0),$$

as f is $\text{Conv}(x_0)$. For $x_j = 0,\ x_0 > 0$ we have:

$$\begin{aligned} &H_j f(x) + H_j f(x + 2e_0) \\ &= \min \begin{cases} f(x - e_0) + P_j^{QR} + f(x + e_0) + P_j^{QR} \\ f(x - e_0) + P_j^{QR} + f(x + 2e_0) + P_j^{EP} \\ f(x) + P_j^{EP} + f(x + e_0) + P_j^{QR} \\ f(x) + P_j^{EP} + f(x + 2e_0) + P_j^{EP} \end{cases} \end{aligned}$$

which has to be greater than or equal to $2H_j f(x + e_0) = 2 \min\{f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\}$. For the third term in the minimization this trivially holds, for the first and fourth term it directly follows as f is $\text{Conv}(x_0)$, and for the second term we have to use this twice:

$$\begin{aligned} f(x - e_0) + P_j^{QR} + f(x + 2e_0) + P_j^{EP} &\geq f(x - e_0) + P_j^{QR} + 2f(x + e_0) + P_j^{EP} - f(x) \\ &\geq f(x) + P_j^{QR} + f(x + e_0) + P_j^{EP}. \end{aligned}$$

For $x_j = 0,\ x_0 = 0$ analogously:

$$\begin{aligned} &H_j f(x) + H_j f(x + 2e_0) \\ &= \min\Big\{f(x) + P_j^{EP} + f(x + e_0) + P_j^{QR}, f(x) + P_j^{EP} + f(x + 2e_0) + P_j^{EP}\Big\} \\ &\geq 2 \min\{f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\} = 2H_j f(x + e_0). \end{aligned}$$

• H_j : Supermod(x_0, x_j), Conv(x_0) $\rightarrow$ Supermod(x_0, x_j).
Assume that f is Supermod(x_0, x_j) and Conv(x_0), then we show that $H_j f$ is Supermod(x_0, x_j). For $x_j > 0$:

$$\begin{aligned} H_j f(x) + H_j f(x + e_0 + e_j) &= f(x - e_j) + f(x + e_0) \\ &\geq f(x + e_0 - e_j) + f(x) = H_j f(x + e_0) + H_j f(x + e_j), \end{aligned}$$

as f is Supermod(x_0, x_j). For $x_j = 0,\ x_0 > 0$ we have:

$$H_j f(x) + H_j f(x + e_0 + e_j) = \min\left\{ f(x - e_0) + P_j^{QR} + f(x + e_0), f(x) + P_j^{EP} + f(x + e_0)\right\},$$

which has to be greater than or equal to

$$H_j f(x + e_0) + H_j f(x + e_j) = \min\left\{ f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\right\} + f(x).$$

For the second term in the minimization this trivially hold, for the first term we use that f is Supermod(x_0, x_j):

$$f(x - e_0) + P_j^{QR} + f(x + e_0) \geq 2f(x) + P_j^{QR}.$$

For $x_j = 0,\ x_0 = 0$ analogously:

$$\begin{aligned} &H_j f(x) + H_j f(x + e_0 + e_j) = f(x) + P_j^{EP} + f(x + e_0) \\ &\quad \geq \min\left\{ f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\right\} + f(x) = H_j f(x + e_0) + H_j f(x + e_j). \end{aligned}$$

• H_j : Supermod(x_0, x_k), Conv(x_0) $\rightarrow$ Supermod(x_0, x_k) for all $k \neq j$.
Assume that f is Supermod(x_0, x_k) and Conv(x_0), then we show that $H_j f$ is Supermod(x_0, x_k). For $x_j > 0$:

$$\begin{aligned} H_j f(x) + H_j f(x + e_0 + e_k) &= f(x - e_j) + f(x + e_0 + e_k - e_j) \\ &\geq f(x + e_0 - e_j) + f(x + e_k - e_j) = H_j f(x + e_0) + H_j f(x + e_k). \end{aligned}$$

as f is Supermod(x_0, x_k). For $x_j = 0,\ x_0 > 0$ we have:

$$\begin{aligned} &H_j f(x) + H_j f(x + e_0 + e_k) \\ &= \min \begin{cases} f(x - e_0) + P_j^{QR} + f(x + e_k) + P_j^{QR}, \\ f(x - e_0) + P_j^{QR} + f(x + e_0 + e_k) + P_j^{EP}, \\ f(x) + P_j^{EP} + f(x + e_k) + P_j^{QR}, \\ f(x) + P_j^{EP} + f(x + e_0 + e_k) + P_j^{EP} \end{cases} \end{aligned}$$

which has to be greater than or equal to $H_j f(x + e_0) + H_j f(x + e_k) = \min\{f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\} + \min\{f(x - e_0 + e_k) + P_j^{QR}, f(x + e_k) + P_j^{EP}\}$. For the third term in the minimization this trivially holds, and for the first and fourth term we use that f is Supermod(x_0, x_k). For the second term we first use this, followed by using that f is Conv(x_0):

$$\begin{aligned} &f(x - e_0) + P_j^{QR} + f(x + e_0 + e_k) + P_j^{EP} \\ &\geq f(x) + f(x - e_0 + e_k) + P_j^{QR} + f(x + e_0 + e_k) + P_j^{EP} - f(x + e_k) \\ &\geq f(x) + P_j^{QR} + f(x + e_k) + P_j^{EP}. \end{aligned}$$

For $x_j = 0,\ x_0 = 0$ analogously:

$$\begin{aligned}
&H_j f(x) + H_j f(x + e_0 + e_k)\\
&= \min\Big\{f(x) + P_j^{EP} + f(x + e_k) + P_j^{QR}, f(x) + P_j^{EP} + f(x + e_0 + e_k) + P_j^{EP}\Big\}\\
&\geq \min\{f(x) + P_j^{QR}, f(x + e_0) + P_j^{EP}\} + \min\{f(x - e_0 + e_k) + P_j^{QR}, f(x + e_k) + P_j^{EP}\}\\
&= H_j f(x + e_0) + H_j f(x + e_k).
\end{aligned}$$

(ii) • H_{QR} : Conv(x_0) $\rightarrow$ Conv(x_0).
Assume that f is Conv(x_0), then we show that $H_{QR}f$ is Conv(x_0) as well. For $x_0 > 0$:

$$\begin{aligned}
H_{QR}f(x) + H_{QR}f(x + 2e_0) = \min\Big\{&f(x - e_0) + f(x + e_0), f(x - e_0) + f(x + 2e_0) + P_0^{EP},\\
&f(x) + P_0^{EP} + f(x + e_0), f(x) + P_0^{EP} + f(x + 2e_0) + P_0^{EP}\Big\}.
\end{aligned}$$

which has to be greater than or equal to $2H_{QR}f(x + e_0) = 2\min\{f(x), f(x + e_0) + P_0^{EP}\}$. For the third term in the minimization this trivially holds, for the first and fourth term we use that f is Conv(x_0), and for the second term we have to use this twice:

$$f(x-e_0)+f(x+2e_0)+P_0^{EP} \geq f(x-e_0)+2f(x+e_0)-f(x)+P_0^{EP} \geq f(x)+f(x+e_0)+P_0^{EP}.$$

For $x_0 = 0$ analogously:

$$\begin{aligned}
&H_{QR}f(x) + H_{QR}f(x + 2e_0)\\
&= \min\Big\{f(x) + P_0^{EP} + f(x + e_0), f(x) + P_0^{EP} + f(x + 2e_0) + P_0^{EP}\Big\}\\
&\geq 2\min\{f(x), f(x + e_0) + P_0^{EP}\} = 2H_{QR}f(x + e_0).
\end{aligned}$$

• H_{QR} : Supermod(x_0, x_k), Conv(x_0) $\rightarrow$ Supermod(x_0, x_k), for $j = 1, \ldots, J$.
Assume that f is Supermod(x_0, x_k) and Conv(x_0), then we show that $H_{QR}f$ is Supermod(x_0, x_k).
For $x_0 > 0$:

$$\begin{aligned}
&H_{QR}f(x) + H_{QR}f(x + e_0 + e_k)\\
&= \min\begin{cases}
f(x - e_0) + f(x + e_k)\\
f(x - e_0) + f(x + e_0 + e_k) + P_0^{EP},\\
f(x) + P_0^{EP} + f(x + e_k)\\
f(x) + P_0^{EP} + f(x + e_0 + e_k) + P_0^{EP}
\end{cases}
\end{aligned}$$

which has to be greater than or equal to $H_{QR}f(x + e_0) + H_{QR}f(x + e_k) = \min\{f(x), f(x + e_0) + P_0^{EP}\} + \min\{f(x + e_k - e_0), f(x + e_k) + P_0^{EP}\}$. For the third term in the minimization this trivially holds, and for the first and fourth term we use that f is Conv(x_0). For the second term we first use this, followed by using that f is Conv(x_0):

$$\begin{aligned}
&f(x - e_0) + f(x + e_0 + e_k) + P_0^{EP}\\
&\qquad\geq f(x) + f(x - e_0 + e_k) + f(x + e_0 + e_k) + P_0^{EP} - f(x + e_k) \geq f(x) + f(x + e_k) + P_0^{EP}.
\end{aligned}$$

For $x_0 = 0$ analogously:

$$
\begin{aligned}
&H_{QR}f(x) + H_{QR}f(x+e_0+e_k)\\
&= \min\Big\{f(x) + P_0^{EP} + f(x+e_k), f(x) + P_0^{EP} + f(x+e_0+e_k) + P_0^{EP}\Big\}\\
&\min\{f(x), f(x+e_0) + P_0^{EP}\} + \min\{f(x+e_k-e_0), f(x+e_k) + P_0^{EP}\}\\
&= H_{QR}f(x+e_0) + H_{QR}f(x+e_k).
\end{aligned}
$$

(iii) We prove the following, for all $j = 0, 1, \ldots, J$:

1) G_j :Conv(x_j) $\rightarrow$ Conv(x_j),
2) G_j :Conv(x_k) $\rightarrow$ Conv(x_k) for all $k \neq j$,
3) G_0 :Supermod(x_0, x_j) $\rightarrow$ Supermod(x_0, x_j) for $j \neq 0$,
4) G_j :Supermod(x_0, x_j) $\rightarrow$ Supermod(x_0, x_j) for $j \neq 0$,
5) G_j :Supermod(x_0, x_k) $\rightarrow$ Supermod(x_0, x_k) for $j \neq 0$ and all $k \neq 0,\ j$.

From this the result of the lemma directly follows. For that, we note that 1) and 2) imply that G_j : Conv$(x_0) \rightarrow$ Conv(x_0) for $j = 0, 1, \ldots, J$.

• 1) G_j : Conv(x_j) $\rightarrow$ Conv(x_j).

Assume that f is Conv(x_j), then we show that $G_j f$ is Conv(x_j) as well. For $x_j + 2 < S_j$:

$$
\begin{aligned}
&G_j f(x) + G_j f(x+2e_j)\\
&= (S_j - x_j)f(x+e_j) + x_j f(x) + (S_j - x_j - 2)f(x+3e_j) + (x_j+2)f(x+2e_j)\\
&= (S_j - x_j - 2)\Big[f(x+e_j) + f(x+3e_j)\Big] + x_j\Big[f(x) + f(x+2e_j)\Big]\\
&\qquad + 2f(x+e_j) + 2f(x+2e_j)\\
&\geq 2(S_j - x_j - 2)f(x+2e_j) + 2x_j f(x+e_j) + 2f(x+e_j) + 2f(x+2e_j)\\
&= 2(S_j - x_j - 1)f(x+2e_j) + 2(x_j+1)f(x+e_j) = 2G_j f(x+e_j),
\end{aligned}
$$

where the inequality holds by applying that f is Conv(x_j) on the parts between brackets. For $x_j + 2 = S_j$ analogously:

$$
\begin{aligned}
G_j f(x) + G_j f(x+2e_j) &= 2f(x+e_j) + (S_j - 2)f(x) + S_j f(x+2e_j)\\
&= 2f(x+e_j) + (S_j - 2)[f(x) + f(x+2e_j)] + 2f(x+2e_j)\\
&\geq 2f(x+e_j) + 2(S_j - 2)f(x+e_j) + 2f(x+2e_j)\\
&= 2f(x+2e_j) + 2(S_j - 1)f(x+e_j) = 2G_j f(x+e_j).
\end{aligned}
$$

• 2) G_j : Conv(x_k) $\rightarrow$ Conv(x_k) for all $k \neq j$.

Assume that f is Conv(x_k), then we show that $G_j f$ for $k \neq j$ is Conv(x_k) as well. For $x_j < S_j$:

$$
\begin{aligned}
&G_j f(x) + G_j f(x+2e_k)\\
&= (S_j - x_j)f(x+e_j) + x_j f(x) + (S_j - x_j)f(x+e_j+2e_k) + x_j f(x+2e_k)\\
&\geq 2(S_j - x_j)f(x+e_j+e_k) + 2x_j f(x+e_k) = 2G_j f(x+e_k),
\end{aligned}
$$

and for $x_j = S_j$:

$$G_j f(x) + G_j f(x + 2\,e_k) = S_j f(x) + S_j f(x + 2\,e_k) \geq 2\,S_j f(x + e_k) = 2\,G_j f(x + e_k).$$

• 3) G_0 : Supermod(x_0, x_j) $\rightarrow$ Supermod(x_0, x_j) for $j \neq 0$.
Assume that f is Supermod(x_0, x_j) for $j \neq 0$, then we show that $G_0 f$ is Supermod(x_0, x_j) as well (for $j \neq 0$). For $x_0 + 1 < S_0$:

$$\begin{aligned}
&G_0 f(x) + G_0 f(x + e_0 + e_j) \\
&= (S_0 - x_0) f(x + e_0) + x_0 f(x) + (S_0 - x_0 - 1) f(x + 2\,e_0 + e_j) \\
&\qquad + (x_0 + 1) f(x + e_0 + e_j) \\
&= (S_0 - x_0 - 1)\Big[f(x + e_0) + f(x + 2\,e_0 + e_j)\Big] + f(x + e_0) + x_0\Big[f(x) + f(x + e_0 + e_j)\Big] \\
&\qquad + f(x + e_0 + e_j) \\
&\geq (S_0 - x_0 - 1)\Big[f(x + e_0 + e_j) + f(x + 2\,e_0)\Big] + f(x + e_0) + x_0\Big[f(x + e_j) + f(x + e_0)\Big] \\
&\qquad + f(x + e_0 + e_j) \\
&= (S_0 - x_0) f(x + e_0 + e_j) + x_0 f(x + e_j) + (S_0 - x_0 - 1) f(x + 2\,e_0) + (x_0 + 1) f(x + e_0) \\
&= G_0 f(x + e_j) + G_j f(x + e_0),
\end{aligned}$$

and for $x_0 + 1 = S_0$:

$$\begin{aligned}
G_0 f(x) + G_0 f(x + e_0 + e_j) &= f(x + e_0) + (S_0 - 1) f(x) + S_0 f(x + e_0 + e_j) \\
&= (S_0 - 1)\Big[f(x) + f(x + e_0 + e_j)\Big] + f(x + e_0) + f(x + e_0 + e_j) \\
&\geq (S_0 - 1)\Big[f(x + e_j) + f(x + e_0)\Big] + f(x + e_0) + f(x + e_0 + e_j) \\
&= f(x + e_0 + e_j) + (S_0 - 1) f(x + e_j) + S_0 f(x + e_0) = G_0 f(x + e_j) + G_j f(x + e_0).
\end{aligned}$$

• 4) G_j : Supermod(x_0, x_j) $\rightarrow$ Supermod(x_0, x_j) for $j \neq 0$.
This follows directly from 3) by symmetry of Supermod(x_0, x_j) in x_0 and x_j. Hence, there is no need to distinguish between G_0 and G_j, and the statement follows.

• 5) G_j : Supermod(x_0, x_k) $\rightarrow$ Supermod(x_0, x_k) for $j \neq 0$ and all $k \neq 0,\ j$.
Assume that f is Supermod(x_0, x_k), then we show that $G_j f$ is Supermod(x_0, x_k) as well (for $j \neq 0$). For $x_j < S_j$:

$$\begin{aligned}
&G_j f(x) + G_j f(x + e_0 + e_k) \\
&= (S_j - x_j) f(x + e_j) + x_j f(x) + (S_j - x_j) f(x + e_0 + e_j + e_k) + x_j f(x + e_0 + e_k) \\
&\geq (S_j - x_j) f(x + e_0 + e_j) + x_j f(x + e_0) + (S_j - x_j) f(x + e_k + e_j) + x_j f(x + e_k) \\
&= G_j f(x + e_0) + G_j f(x + e_k),
\end{aligned}$$

and for $x_j = S_j$:

$$\begin{aligned}
G_j f(x) + G_j f(x + e_0 + e_k) &= S_j f(x) + S_j f(x + e_0 + e_k) \\
&\geq S_j f(x + e_0) + S_j f(x + e_k) \\
&= G_j f(x + e_0) + G_j f(x + e_k). \qquad \square
\end{aligned}$$

3.A.2 Proof of Theorem 3.3.3

PROOF. Consider a demand directly at the QR warehouse ($j = 0$), or an overflow demand from local warehouse $j \in \{1,\ldots,J\}$ when $x_j = 0$. There are two options for such a demand: accepting it (if $x_0 > 0$) or rejecting it at the QR warehouse. Let

$$w_j(u,x) := \begin{cases} P_j^{QR} + V_n(x - e_0) & \text{if } u = 1 \text{ (accept)}, \\ P_j^{EP} + V_n(x) & \text{if } u = 0 \text{ (reject)}. \end{cases}$$

Then $H_j V_n(x) = \min_{u\in\{0,1\}} w_j(u,x)$ for x such that $x_j = 0$ and $x_0 > 0$. Also, when defining $P_0^{QR} = 0$, then $H_{QR} V_n(x) = \min_{u\in\{0,1\}} w_j(u,x)$ for x such that $x_0 > 0$.

Let $\Delta_{x_k} w_j(u,x) := w_j(u,x+e_k) - w_j(u,x)$ for all x with $x_0 > 0$ and $x_k < S_k$. Then, for $x_0 > 0$:

$$\begin{aligned} \Delta_{x_0} w_j(1,x) - \Delta_{x_0} w_j(0,x) &= w_j(1,x+e_0) - w_j(1,x) - w_j(0,x+e_0) + w_j(0,x) \\ &= V_n(x) - V_n(x-e_0) - V_n(x+e_0) + V_n(x) \leq 0, \end{aligned}$$

as V_n is Conv(x_0). Furthermore, for $x_0 > 0,\ k \neq j$:

$$\begin{aligned} \Delta_{x_k} w_j(1,x) - \Delta_{x_k} w_j(0,x) &= w_j(1,x+e_k) - w_j(1,x) - w_j(0,x+e_k) + w_j(0,x) \\ &= V_n(x-e_0+e_k) - V_n(x-e_0) - V_n(x+e_k) + V_n(x) \leq 0, \end{aligned}$$

as V_n is Supermod(x_0, x_k).

This implies that, for every $n \geq 0$, there exists a switching curve, say T_j^n, which is a function of $x^{(0,j)}$, such that the optimal decision at the QR warehouse is to accept the demand if $x_0 > T_j^n(x^{(0,j)})$, and to reject it if $x_0 \leq T_j^n(x^{(0,j)})$. As overflow demands from local warehouse j can only occur when $x_j = 0$, the switching curve does not depend on x_j. Moreover, it follows that T_j^n is decreasing in each of its components.

Hence, if f_{n+1} is the minimizing policy in (3.2.1), then f_{n+1} is a state-dependent threshold policy described by the switching curves T_j^{n+1}, $j = 0,1,\ldots,J$. Note that the transition probability matrix of every stationary policy is unichain (since every state can access $(S_0,S_1,\ldots,S_J)$) and aperiodic (since the transition probability from state $(S_0,S_1,\ldots,S_J)$ to itself is positive). Then, by [155, Theorem 8.5.4], the long run average costs under the stationary policy f_{n+1} converges to the minimal long run average costs as n tends to infinity. Since there are only finitely many stationary threshold policies, this implies that there exists an optimal stationary policy that is a state-dependent threshold type policy. □

3.A.3 Proof of Proposition 3.3.4

PROOF. We show that when $a_{j_2}^*(x) = 1$ then $a_{j_1}^*(x) = 1$ as well, for all states x such that $x_{j_1} = x_{j_2} = 0$. Suppose that $a_{j_2}^*(x) = 1$, i.e. applying a quick response at warehouse j_2 in case of a stock out is optimal, then

$$f(x-e_0) + P_{j_2}^{QR} \leq f(x) + P_{j_2}^{EP}.$$

Hence

$$f(x-e_0) \leq f(x) + \Delta P_{j_2} \leq f(x) + \Delta P_{j_1},$$

where the second inequality holds by the condition $\Delta P_{j_1} \geq \Delta P_{j_2}$. Now

$$f(x-e_0)+P_{j_1}^{QR} \leq f(x)+P_{j_1}^{EP},$$

and so $a_{j_1}^*(x)=1$. □

3.A.4 Proof of Theorem 3.3.5

PROOF. We prove the theorem 1) for local warehouse j, $j=1,\dots,J$, and 2) for the QR warehouse.

1) We prove that when (3.3.1) is satisfied, the following holds:

$$V_n(x-e_0)+P_j^{QR} \leq V_n(x)+P_j^{EP}, \text{ for all } x \text{ with } x_j=0 \text{ and } x_0=1.$$

Hence, a demand from local warehouse j is always accepted at the QR warehouse in this case. It follows from the structural results of Theorem 3.3.3 that this action is optimal as well for $x_0=2,\dots,S_0$ (and $x_j=0$).

Write $x_{a,b}$ for x with $x_0=a$ and $x_j=b$:

$$x_{a,b} := (a, x_1, \dots, x_{j-1}, b, x_{j+1}, \dots, x_J).$$

We prove that when (3.3.1) holds, then $V_n(x_{0,0})-V_n(x_{1,0}) \leq P_j^{EP}-P_j^{QR}$, for all $n \geq 0$, where the entries not x_0 and x_j are equal for $x_{0,0}$ and $x_{1,0}$. For this, we use induction on V_n and consider each of the operators separately. For $V_0 \equiv 0$ the inequality clearly holds. Assume that it holds for a certain n and denote this V_n by f. That is, the induction hypothesis $(i.h.)$ is given by:

$$f(x_{0,0})-f(x_{1,0}) \leq P_j^{EP}-P_j^{QR}. \qquad (i.h.)$$

We apply each of the operators separately to $f(x_{0,0})-f(x_{1,0})$. All inequalities hold by $(i.h.)$ unless stated otherwise.

• H_{QR}:

$$H_{QR}f(x_{0,0})-H_{QR}f(x_{1,0}) = \max\{P_0^{EP}, f(x_{0,0})-f(x_{1,0})\} \leq \max\{P_0^{EP}, P_j^{EP}-P_j^{QR}\}.$$

• H_j:

$$H_jf(x_{0,0})-H_jf(x_{1,0}) = \max\{P_j^{EP}-P_j^{QR}, f(x_{0,0})-f(x_{1,0})\} \leq P_j^{EP}-P_j^{QR}.$$

• H_k (for $k \neq j$): if $x_k>0$:

$$H_kf(x_{0,0})-H_kf(x_{1,0}) = f(x_{0,0}-e_k)-f(x_{1,0}-e_k) \leq P_j^{EP}-P_j^{QR},$$

and if $x_k=0$:

$$H_kf(x_{0,0})-H_kf(x_{1,0}) = \max\{P_k^{EP}-P_k^{QR}, f(x_{0,0})-f(x_{1,0})\} \leq \max\{P_k^{EP}-P_k^{QR}, P_j^{EP}-P_j^{QR}\}.$$

Denoting by $\mathbb{I}_{\{x_k=0\}}$ the indicator function being 1 when $x_k=0$ and zero otherwise, we have:

$$H_kf(x_{0,0})-H_kf(x_{1,0}) \leq \max\{\mathbb{I}_{\{x_k=0\}} \cdot (P_k^{EP}-P_k^{QR}), P_j^{EP}-P_j^{QR}\}.$$

• G_j:

$$G_j f(x_{0,0}) - G_j f(x_{1,0}) = S_j[f(x_{0,1}) - f(x_{1,1})] \le S_j[f(x_{0,0}) - f(x_{1,0})] \le S_j[P_j^{EP} - P_j^{QR}],$$

where the first inequality holds as f (i.e. V_n) is Supermod(x_0, x_k) (cf. Theorem 3.3.5).
• G_0:

$$\begin{aligned}
&G_0 f(x_{0,0}) - G_0 f(x_{1,0})\\
&= S_0 f(x_{1,0}) - (S_0 - 1) f(x_{2,0}) - f(x_{1,0})\\
&= S_0[f(x_{1,0}) - f(x_{2,0})]\\
&\le (S_0 - 1)[f(x_{0,0}) - f(x_{1,0})] \le (S_0 - 1)[P_j^{EP} - P_j^{QR}],
\end{aligned}$$

where the first inequality holds as f (i.e. V_n) is Conv(x_0) (cf. Theorem 3.3.5).
• G_k (for $k \neq 0,\ j$):

$$\begin{aligned}
&G_k f(x_{0,0}) - G_k f(x_{1,0})\\
&= (S_k - x_k) f(x_{0,0} + e_k) + x_k f(x_{0,0}) - (S_k - x_k) f(x_{1,0} + e_k) - x_k f(x_{1,0})\\
&= (S_k - x_k)[f(x_{0,0} + e_k) - f(x_{1,0} + e_k)] + x_k[f(x_{0,0}) - f(x_{1,0})]\\
&\le (S_k - x_k)(P_j^{EP} - P_j^{QR}) + x_k(P_j^{EP} - P_j^{QR}) = S_k[P_j^{EP} - P_j^{QR}].
\end{aligned}$$

Combining these results yields (recall that $\nu = \sum_{k=0}^J \lambda_k + \sum_{k=0}^J \mu_k S_k$):

$$\begin{aligned}
\nu \left(V_{n+1}(x_{0,0}) - V_{n+1}(x_{1,0})\right) &= \left(h_0(0) - h_0(1)\right) + \lambda_0 \left(H_{QR} f(x_{0,0}) - H_{QR} f(x_{1,0})\right)\\
&\quad + \sum_{k=1}^J \lambda_k \left(H_k f(x_{0,0}) - H_k f(x_{1,0})\right) + \sum_{k=0}^J \mu_k \left(G_k f(x_{0,0}) - G_k f(x_{1,0})\right)\\
&\le \left(h_0(0) - h_0(1)\right) + \lambda_0 \max\{P_0^{EP}, P_j^{EP} - P_j^{QR}\} + \lambda_j \left(P_j^{EP} - P_j^{QR}\right)\\
&\quad + \sum_{\substack{k=1\\k\neq j}}^J \lambda_k \max\{\mathbb{I}_{\{x_k=0\}} \cdot (P_k^{EP} - P_k^{QR}), P_j^{EP} - P_j^{QR}\} + \mu_0(S_0 - 1)[P_j^{EP} - P_j^{QR}]\\
&\quad + \mu_j S_j[P_j^{EP} - P_j^{QR}] + \sum_{\substack{k=1\\k\neq j}}^J \mu_k S_k[P_j^{EP} - P_j^{QR}] \quad (*)
\end{aligned} \tag{3.A.1}$$

Now assume that $P_j^{EP} - P_j^{QR} \ge P_0^{EP}$ and (when $x_k = 0$) $P_j^{EP} - P_j^{QR} \ge P_k^{EP} - P_k^{QR}$, then, continuing from (3.A.1) we have:

$$(*) = \left(h_0(0) - h_0(1)\right) + (\nu - \mu_0)\left(P_j^{EP} - P_j^{QR}\right) \le \nu\left(P_j^{EP} - P_j^{QR}\right),$$

as $h_0(0) - h_0(1)$ is negative. So, under this assumptions the induction step holds. Now assume $P_j^{EP} - P_j^{QR} \le P_0^{EP}$ and (when $x_k = 0$) $P_j^{EP} - P_j^{QR} \le P_k^{EP} - P_k^{QR}$, then again continuing from (3.A.1) we have:

$$\begin{aligned}
(*) &= \left(h_0(0) - h_0(1)\right) + \lambda_0 P_0^{EP} + \sum_{k=1}^J \lambda_k[P_k^{EP} - P_k^{QR}] + \left(\sum_{k=0}^J \mu_k S_k - \mu_0\right)\left(P_j^{EP} - P_j^{QR}\right)\\
&\le \left(\mu_0 + \sum_{k=0}^J \lambda_k\right)\left(P_j^{EP} - P_j^{QR}\right) + \left(\sum_{k=0}^J \mu_k S_k - \mu_0\right)\left(P_j^{EP} - P_j^{QR}\right) = \nu\left(P_j^{EP} - P_j^{QR}\right),
\end{aligned}$$

where the inequality holds by (3.3.1). Hence, also in this case the induction step holds. The other possible cases can easily be checked to follow analogously.

2) We show that when (3.3.1) is satisfied, the following holds:

$$V_n(x-e_0) \le V_n(x) + P_0^{EP}, \text{ for all } x \text{ with } x_0 = 1.$$

Hence, a demand at the QR warehouse j is always accepted in this case. It follows from the structural results of Theorem 3.3.3 that this action is optimal as well for $x_0 = 2, \ldots, S_0$.

Write x_a for x with $x_0 = a$, that is $x_{(a)} := (a, x_1, \ldots, x_J)$. We prove that when (3.3.1) holds, then $V_n(x_{(0)}) - V_n(x_{(1)}) \le P_0^{EP}$, for all $n \ge 0$, where the entries not x_0 are equal for $x_{(0)}$ and $x_{(1)}$. For this, we use induction on V_n and consider each of the operators separately. For $V_0 \equiv 0$ the inequality clearly holds. Assume that it holds for a certain n and denote this V_n by f. That is, the induction hypothesis $(i.h.)$ is given by:

$$f(x_{(0)}) - f(x_{(1)}) \le P_0^{EP}. \qquad (i.h.)$$

We apply each of the operators separately to $f(x_{(0)}) - f(x_{(1)})$. All inequalities hold by $(i.h.)$ unless stated otherwise.

• H_{QR}:

$$H_{QR}f(x_{(0)}) - H_{QR}f(x_{(1)}) = \max\{P_0^{EP}, f(x_{(0)}) - f(x_{(1)})\} \le P_0^{EP}.$$

• H_k: if $x_k > 0$:

$$H_k f(x_{(0)}) - H_k f(x_{(1)}) = f(x_{(0)} - e_k) - f(x_{(1)} - e_k) \le P_0^{EP},$$

and if $x_k = 0$:

$$H_k f(x_{(0)}) - H_k f(x_{(1)}) = \max\{P_k^{EP} - P_k^{QR}, f(x_{(0)}) - f(x_{(1)})\} \le \max\{P_k^{EP} - P_k^{QR}, P_0^{EP}\}.$$

• G_0:

$$\begin{aligned} G_0 f(x_{(0)}) - G_0 f(x_{(1)}) &= S_0 f(x_{(1)}) - (S_0 - 1) f(x_{(2)}) - f(x_{(1)}) \\ &= S_0[f(x_{(1)}) - f(x_{(2)})] \le (S_0 - 1)[f(x_{(0)}) - f(x_{(1)})] \le (S_0 - 1) P_0^{EP}, \end{aligned}$$

where the first inequality holds as f (i.e. V_n) is Conv(x_0) (cf. Theorem 3.3.5).

• G_k (for $k \neq 0$):

$$\begin{aligned} & G_k f(x_{(0)}) - G_k f(x_{(1)}) \\ &= (S_k - x_k) f(x_{(0)} + e_k) + x_k f(x_{(0)}) - (S_k - x_k) f(x_{(1)} + e_k) - x_k f(x_{(1)}) \\ &= (S_k - x_k)[f(x_{(0)} + e_k) - f(x_{(1)} + e_k)] + x_k [f(x_{(0)}) - f(x_{(1)})] \\ &\le (S_k - x_k) P_0^{EP} + x_k P_0^{EP} = S_k P_0^{EP}. \end{aligned}$$

Combining these results yields:

$$\nu\left(V_{n+1}(x_{(0)})-V_{n+1}(x_{(1)})\right)=\left(h_0(0)-h_0(1)\right)+\lambda_0\left(H_{QR}f(x_{(0)})-H_{QR}f(x_{(1)})\right)$$
$$+\sum_{k=1}^{J}\lambda_k\left(H_kf(x_{(0)})-H_kf(x_{(1)})\right)+\sum_{k=0}^{J}\mu_k\left(G_kf(x_{(0)})-G_kf(x_{(1)})\right)$$
$$\leq\left(h_0(0)-h_0(1)\right)+\lambda_0P_0^{EP}+\sum_{k=1}^{J}\lambda_k\max\{\mathbb{I}_{\{x_k=0\}}(P_k^{EP}-P_k^{QR}),P_0^{EP}\}\tag{3.A.2}$$
$$+P_0^{EP}\Big(\sum_{k=0}^{J}\mu_kS_k-\mu_0\Big).\ (**)\tag{3.A.3}$$

When all $x_k>0$, or $P_0^{EP}\geq P_k^{EP}-P_k^{QR}$ for $x_k=0$, then, continuing from (3.A.3) we have:

$$(**)=\left(h_0(0)-h_0(1)\right)+(\nu-\mu_0)P_0^{EP}\leq\nu P_0^{EP},$$

as $h_0(0)-h_0(1)$ is negative. So, in this case the induction step holds. When all $x_k=0$ and $P_0^{EP}\leq P_k^{EP}-P_k^{QR}$ then, again continuing from (3.A.3) we have:

$$(**)=\left(h_0(0)-h_0(1)\right)+P_0^{EP}\left(\lambda_0+\sum_{k=0}^{J}\mu_kS_k-\mu_0\right)+\sum_{k=1}^{J}\lambda_k\left(P_k^{EP}-P_k^{QR}\right)$$
$$\leq P_0^{EP}\left(\mu_0+\sum_{k=1}^{J}\lambda_k\right)+P_0^{EP}\left(\lambda_0+\sum_{k=0}^{J}\mu_kS_k-\mu_0\right)=\nu P_0^{EP},$$

where the inequality holds by (3.3.1). Hence, also in this case the induction step holds. The other possible cases can easily be checked to follow analogously. □

3.A.5 Proof of Lemma 3.5.1

PROOF. • $\tilde{G}_j:\mathrm{Decr}(x_0)\ \to\ \mathrm{Decr}(x_0)$.
Assume that f is Decr(x_0), then we show that $\tilde{G}_jf$ is Decr(x_0) as well. The cases $j\in\{1,\ldots,J\}$ are trivial, so we only show $j=0$. For $x_0+1<S_0$ we have:

$$\tilde{G}_0f(x)-\tilde{G}_0f(x+e_0)$$
$$=\phi_0(y_0)f(x+e_0)+\left(\overline{\phi}_0-\phi_0(y_0)\right)f(x)-\phi_0(y_0-1)f(x+2e_0)$$
$$-\left(\overline{\phi}_0-\phi_0(y_0-1)\right)f(x+e_0)$$
$$=\phi_0(y_0-1)\left(f(x+e_0)-f(x+2e_0)\right)+\left(\overline{\phi}_0-\phi_0(y_0)\right)\left(f(x)-f(x+e_0)\right)\ \geq 0,$$

and for $x_0+1=S_0$:

$$\tilde{G}_0f(x)-\tilde{G}_0f(x+e_0)$$
$$=\phi_0(y_0)f(x+e_0)+\left(\overline{\phi}_0-\phi_0(y_0)\right)f(x)-\overline{\phi}_0f(x+e_0)$$
$$=\left(\overline{\phi}_0-\phi_0(y_0)\right)\left(f(x)-f(x+e_0)\right)\geq 0.$$

- $\tilde{G}_j : \mathrm{Conv}(x_0), \mathrm{Decr}(x_0) \rightarrow \mathrm{Conv}(x_0)$.

For $j \neq 0$ trivially $\tilde{G}_j : \mathrm{Conv}(x_0) \rightarrow \mathrm{Conv}(x_0)$. Hence we only show the proof for $j = 0$. Assume that f is $\mathrm{Conv}(x_0)$ and $\mathrm{Decr}(x_0)$, then we show that $\tilde{G}_0 f$ is $\mathrm{Conv}(x_0)$. Write $\overline{\phi}_j(y_j) = \overline{\phi}_j - \phi_j(y_j)$. For $x_0 + 2 < S_0$ we have

$$
\begin{aligned}
&G_0 f(x) + \tilde{G}_0 f(x + 2e_0) - \tilde{G}_0 f(x + e_0) \\
&= \phi_0(y_0) f(x+e_0) + \overline{\phi}_0(y_0) f(x) + \phi_0(y_0 - 2) f(x + 3e_0) \\
&\quad + \overline{\phi}_0(y_0 - 2) f(x + 2e_0) - 2\phi_0(y_0 - 1) f(x + 2e_0) - 2\overline{\phi}_0(y_0 - 1) f(x + e_0) \\
&= \phi_0(y_0 - 2)\Big(f(x + 3e_0) + f(x + e_0) - 2f(x + 2e_0)\Big) + 2\Big(\overline{\phi}_0 - \overline{\phi}_0(y - 2)\Big) f(x + 2e_0) \\
&\quad - \Big(\overline{\phi}_0 - \overline{\phi}_0(y - 2)\Big) f(x + e_0) + \Big(\overline{\phi}_0 - \overline{\phi}_0(y)\Big) f(x + e_0) - 2\Big(\overline{\phi}_0 - \overline{\phi}_0(y - 1)\Big) f(x + 2e_0) \\
&\quad + \overline{\phi}_0(y_0) f(x) + 2\overline{\phi}_0(y_0 - 2) f(x + 2e_0) - 2\overline{\phi}_0(y_0 - 1) f(x + e_0) \\
&\geq -\overline{\phi}_0(y - 2) f(x + 2e_0) + \overline{\phi}_0(y - 2) f(x + e_0) - \phi_0(y_0) f(x + e_0) + 2\phi_0(y_0 - 1) f(x + 2e_0) \\
&\quad + \overline{\phi}_0(y) f(x) - 2\overline{\phi}_0(y_0 - 1) f(x + e_0) \\
&= \overline{\phi}_0(y - 2)\Big(f(x + e_0) - f(x + 2e_0)\Big) - 2\overline{\phi}_0(y - 1)\Big(f(x + e_0) - f(x + 2e_0)\Big) \\
&\quad + \overline{\phi}_0(y)\Big(f(x) - f(x + e_0)\Big) \\
&\geq \Big(\overline{\phi}_0(y - 2) - 2\overline{\phi}_0(y - 1) + \overline{\phi}_0(y)\Big)\Big(f(x + e_0) - f(x + 2e_0)\Big) \geq 0,
\end{aligned}
$$

where the first two inequalities hold as f is $\mathrm{Conv}(x_0)$, and the last as f is $\mathrm{Decr}(x_0)$ and $\overline{\phi}_0(\cdot)$ is convex (which holds as $\phi_0(\cdot)$ is concave).

For $x_0 + 2 = S_0$ we have:

$$
\begin{aligned}
&\tilde{G}_0 f(x) + \tilde{G}_0 f(x + 2e_0) - \tilde{G}_0 f(x + e_0) \\
&= \phi_0(2) f(x + e_0) + \overline{\phi}_0(2) f(x) + \overline{\phi}_0(0) f(x + 2e_0) - 2\phi_0(1) f(x + 2e_0) - 2\overline{\phi}_0(1) f(x + e_0) \\
&= \Big(\overline{\phi}_0 - \overline{\phi}_0(2)\Big) f(x + e_0) + \overline{\phi}_0(2) f(x) + \overline{\phi}_0(0) f(x + 2e_0) \\
&\qquad - 2\Big(\overline{\phi}_0 - \overline{\phi}_0(1)\Big) f(x + 2e_0) - 2\overline{\phi}_0(1) f(x + e_0) \\
&= \overline{\phi}_0(2)\Big(f(x) - f(x + e_0)\Big) - 2\overline{\phi}_0(1)\Big(f(x + e_0) - f(x + 2e_0)\Big) \\
&\qquad + \overline{\phi}_0(0)\Big(f(x + e_0) - f(x + 2e_0)\Big) \\
&\geq \Big(\overline{\phi}_0(2) - 2\overline{\phi}_0(1) + \overline{\phi}_0(0)\Big)\Big(f(x + e_0) - f(x + 2e_0)\Big) \geq 0.
\end{aligned}
$$

where the last inequality holds as f is $\mathrm{Decr}(x_0)$ and $\overline{\phi}_0(\cdot)$ is convex. Note that we have used that $\overline{\phi}_0(0) = \overline{\phi}_0 - \phi_0(0) = \overline{\phi}_0$.

- $\tilde{G}_j : \mathrm{Supermod}(x_0, x_j) \rightarrow \mathrm{Supermod}(x_0, x_j)$, for$j = 1, \ldots, J$.

Assume that f is $\mathrm{Conv}(x_0)$ and $\mathrm{Supermod}(x_0, x_j)$ for $j \neq 0$, then we show that $\tilde{G}_k f$ is $\mathrm{Supermod}(x_0, x_j)$ as well. We consider $\tilde{G}_0$ (then $\tilde{G}_j$ follows by symmetry) and $\tilde{G}_k$ for

$k \neq j$ separately. For $\tilde{G}_0$ with $x_0 + 1 < S_0$ we have:

$$\begin{aligned}
&\tilde{G}_0 f(x) + \tilde{G}_0 f(x+e_0+e_j) - \tilde{G}_0 f(x+e_0) - \tilde{G}_0 f(x+e_j) \\
&= \phi_0(y_0) f(x+e_0) + (\overline{\phi}_0 - \phi_0(y_0)) f(x) + \phi_0(y_0-1) f(x+2\,e_0+e_j) \\
&\quad + (\overline{\phi}_0 - \phi_0(y_0-1)) f(x+e_0+e_j) - \phi_0(y_0-1) f(x+2\,e_0) - (\overline{\phi}_0 - \phi_0(y_0-1)) f(x+e_0) \\
&\quad - \phi_0(y_0) f(x+e_0+e_j) - (\overline{\phi}_0 - \phi_0(y_0)) f(x+e_j) \\
&= \phi_0(y_0-1) \Big(f(x+2\,e_0+e_j) + f(x+e_0) - f(x+2\,e_0) - f(x+e_0+e_j) \Big) \\
&\quad + (\overline{\phi}_0 - \phi_0(y_0)) \Big(f(x) + f(x+e_0+e_j) - f(x+e_0) - f(x+e_j) \Big) \geq 0,
\end{aligned}$$

and for $x_0 + 1 = S_0$ we have:

$$\begin{aligned}
&\tilde{G}_0 f(x) + \tilde{G}_0 f(x+e_0+e_j) - \tilde{G}_0 f(x+e_0) - \tilde{G}_0 f(x+e_j) \\
&= \phi_0(y_0) f(x+e_0) + (\overline{\phi}_0 - \phi_0(y_0)) f(x) + \overline{\phi}_0 f(x+e_0+e_j) \\
&\qquad - \overline{\phi}_0 f(x+e_0) - \phi_0(y_0) f(x+e_0+e_j) - (\overline{\phi}_0 - \phi_0(y_0)) f(x+e_j) \\
&= (\overline{\phi}_0 - \phi_0(y_0)) \Big(f(x) + f(x+e_0+e_j) - f(x+e_0) - f(x+e_j) \Big) \geq 0.
\end{aligned}$$

For $\tilde{G}_k$ $(k \neq 0, j)$ with $x_k < S_k$ we have:

$$\begin{aligned}
&\tilde{G}_k f(x) + \tilde{G}_k f(x+e_0+e_j) - \tilde{G}_k f(x+e_0) - \tilde{G}_k f(x+e_j) \\
&= \phi_k(y_k) f(x+e_k) + (\overline{\phi}_k - \phi_k(y_k)) f(x) + \phi_k(y_k) f(x+e_0+e_j+e_k) \\
&\quad + (\overline{\phi}_k - \phi_k(y_k)) f(x+e_0+e_j) - \phi_k(y_k) f(x+e_0+e_k) \\
&\quad - (\overline{\phi}_k - \phi_k(y_k)) f(x+e_0) - \phi_k(y_k) f(x+e_j+e_k) - (\overline{\phi}_k - \phi_k(y_k)) f(x+e_j) \\
&= \phi_k(y_k) \Big(f(x+e_k) + f(x+e_0+e_j+e_k) - f(x+e_0+e_k) - f(x+e_j+e_k) \Big) \\
&\quad + (\overline{\phi}_k - \phi_k(y_k)) \Big(f(x) + f(x+e_0+e_j) - f(x+e_0) - f(x+e_j) \Big) \geq 0,
\end{aligned}$$

and for $x_k = S_k$ we have:

$$\begin{aligned}
&\tilde{G}_k f(x) + \tilde{G}_k f(x+e_0+e_j) - \tilde{G}_k f(x+e_0) - \tilde{G}_k f(x+e_j) \\
&\qquad = \overline{\phi}_k \Big(f(x) + f(x+e_0+e_j) - f(x+e_0) - f(x+e_j) \Big) \geq 0. \qquad \square
\end{aligned}$$

4

Optimal lateral transshipment policy for a two-location inventory model

For an spare parts inventory model with two stockpoints, we completely characterize and prove the structure of the optimal lateral transshipment policy, using dynamic programming. That is, we derive the optimal policy for satisfying demands, minimizing the average costs of the system in the long-run. This optimal policy is a threshold type policy. In addition, we derive conditions under which the so-called *hold back* and *complete pooling* policies are optimal, two policies that are often assumed in the literature.

4.1 Introduction

In this chapter we study an inventory model with two stockpoints, which provide spare parts for advanced technical systems. These systems are typically used in the primary processes of their users. Hence, any down-time of these systems is extremely costly, so ready-for-use spare parts are kept in stock for the critical component of these systems. We focus on a single, repairable part, for which a repair-by-replacement strategy is executed: upon failure of a system, the defective part is replaced by a part from inventory. The defective part is returned to the stockpoint, where it is repaired and added to the inventory. We take the initial number of spare parts on hand at each location to be given.

The stockpoints service two groups of technical systems, where each group is assigned to one stockpoint. In case of a breakdown of one of the systems, it demands a spare part at its dedicated stockpoint. The demands form a Poisson process at each stockpoint, of which the rates may differ. If a demand for a spare part is directly met at the stockpoint, we refer to this as a demand that is *directly* fulfilled. Otherwise, there are two other possibilities. The first option is a *lateral transshipment*, which means that a part is shipped from the other stockpoint. In this case, the system is down while it is waiting for the part, and extra transportation costs are incurred. The second option is an *emergency repair procedure*: the defective part is repaired in a fast repair procedure, for which high costs are incurred, and the system is down for a longer period of time. As downtime costs are huge, mainly because of loss of production, this option is much more expensive than a lateral transshipment. Due to these downtime costs, backordering of demands is not allowed.

By the use of lateral transshipments, significant costs can be saved, as the costs for a lateral transshipment are much smaller than the costs for an emergency procedure. For example, Kranenburg and Van Houtum [120] show that the company ASML, an original equipment manufacturer in the semiconductor industry, can reduce their spare parts provisioning costs by up to 50% through the efficient use of lateral transshipments, while keeping the service at the same level. Robinson [162] shows that substantial costs savings can be realized by the use of lateral transshipments, even when the transportation costs are high. Based on two case studies in the computer and automobile industry, Cohen and Lee [55] maintain that stock pooling is an effective way to improve the service levels, even with less on-hand inventory. Also, Cohen et al. [53] point out that the pooling of spare parts is one of the best ways for companies to realize cost reductions.

In this chapter, we focus on the optimization of the lateral transshipment policy. That is, we determine the optimal decision on how to fulfill a demand to minimize the average running costs of the system in the long-run: (i) directly from own stock, (ii) via a lateral transshipment, or (iii) via an emergency procedure. When is it beneficial to apply a lateral transshipment, and when is it better to apply an emergency procedure? A straightforward strategy would be to always fulfill demands from the own stockpoint, if possible, and otherwise via a lateral transshipment, if possible. This strategy is known as *complete pooling* (or *full pooling*) of inventory.

Depending on the cost parameters, a complete pooling strategy is suboptimal in certain cases. If a stockpoint has, for example, only one part left in stock, it could be beneficial to hold this one back, even if the other stockpoint requests a lateral transshipment. This situation can occur when the cost parameters for both stockpoints are equal (i.e. symmetric), but its effect may be even larger for asymmetric costs parameters. It could, in fact, be better to hold back parts even in case of a demand at the stockpoint itself, so as to be able to respond to a future lateral transshipment request of the other stockpoint. This situation, where stockpoints can hold back some inventory, is known as *partial pooling*. A *hold back strategy* (cf. Xu et al. [215]) is a special case of this in which outgoing transshipments are limited.

A considerable amount of work has already been done on the use of lateral transshipments in various settings (see e.g. Wong et al. [212] and Paterson et al. [153] for an overview). Most of it focuses on (approximate) evaluation of performance characteristics, (approximate) optimization of parameters when the policy is given, or optimization of the replenishment strategy. However, only limited results seem to be known about the optimal lateral transshipments policy. Archibald et al. [6] study a periodic review model and prove the optimal transshipment policy in case of stock-outs. They find a threshold type policy where the thresholds depend on the remaining time in a period. However, they assume zero replenishment lead times and their approach is inappropriate when these are positive. For optimal lateral transshipments rules in a continuous time review setting, hardly any results are available. In Zhao et al. [218] a two location make-to-stock system is considered for which the optimal production and optimal transshipment policy are derived. They show that both of them can be described by a switching curve, i.e. by state-dependent thresholds. However, they do not allow inventory to be held back at a location. By generalization of this decision, we show that always satisfying a demand directly from stock might be suboptimal. So the optimization of the lateral transshipment policy in the current setting has not been done before. In Section 2 we present a more detailed literature review.

Our main contribution is as follows. For the described model, (a) we completely characterize and prove the structure of the optimal lateral transshipment policy, which is a threshold type policy. For this model, with positive lead times, the possibility to hold back stock and to allow for asymmetric cost parameters, the optimal policy structure has not been derived before. Next to this, (b) we give conditions under which the optimal policy simplifies to either a *hold back policy* or a *complete pooling policy*. The latter strategy is often assumed in the literature about lateral transshipments, see e.g. [2, 10, 90, 123, 126, 165, 174, 212, 218]. So we contribute to the literature by presenting conditions on the cost parameters under which this policy is indeed optimal.

We model the inventory problem as a Markov decision problem (see e.g. Puterman [155]). Based on the inventory levels, a decision has to be taken each time a demand arises at one of the two stockpoints. Using Event Based Dynamic Programming (see Koole [115, 116]), we build up the n-period minimal cost function (the *value function*) of so-called event operators, where the possible events are demands and repairs. Proving structural properties of the value function, such as monotonicity and multimodularity, can then be done by considering each of these operators separately. Hence, this reduces the complexity of the problem. From this we derive the optimal lateral transshipment policy as well as conditions under which it is simplified.

Our model was inspired by an inventory system with repairables, which is common practice in the spare parts industry. However, the model applies much more generally. Inventory systems with replenishments (or productions to stock) fall within the same model, in case a base stock policy is executed, with one-for-one replenishments (or productions). The repair times in the current model resemble the replenishment (or production) lead times and the emergency procedures resemble lost sales. This interpretation would be more suitable when the inventory consists of consumables. Also inventory systems in which substitutions are allowed, fall within the same model. We study these options as model extensions.

The outline of this chapter is as follows. We start with a literature review in Section 4.2. In Section 4.3 we describe the model in more detail and we introduce the notation. We model the system as a Markov decision problem, and introduce the technique of Event Based Dynamic Programming. In Section 4.4 we give the structural properties of the event operators and of the value function. This leads to the characterization of the structure of the optimal policy which is a threshold type policy. Conditions are given under which certain simple policies are optimal and some examples are shown. In Section 4.5 we consider several extensions to the model. Finally, we summarize the results and indicate possibilities for further research in Section 4.6. Appendix 4.A contains all proofs. This chapter is based on [193].

4.2 Literature review

A considerable amount of work has already been done on the use of lateral transshipments in various settings. Wong et al. [212] and Paterson et al. [153] provide good overviews. We distinguish between two types of work, depending on the lateral transshipment rule. It can either be a predetermined and fixed rule, or it is subject to optimization. In most of the literature, a given predetermined fixed rule is assumed, and performance characteristics of the system are evaluated, either exactly or approximately (e.g. [126, 10, 165, 175, 123]), or optimal reorder policies are derived (e.g. [162, 150]).

More relevant in relation to our work, is the literature on the *optimization* of the lateral transshipment rule. For the periodic-review case, we mention the following results. For a system with two locations, Archibald et al. [6] prove the optimal transshipment policy in case of stock-outs. It states when it is optimal to apply a lateral transshipment in case of a demand at a location with zero stock. The decision is based both on the stock level of the other location, and on the time left until the next replenishment opportunity, where it is assumed that the replenishments occur instantaneously (i.e., if we would let the period length go to zero, we would get a model with zero replenishment lead time). The costs consist of the regular and emergency ordering costs, transshipments costs and holding costs. The model they consider, however, considerably differs from our model. Firstly, we study a model with positive lead times for which the approach of [6] is inappropriate as there are no regeneration points any more. It is generally known in inventory theory, that there is a major difference between positive and zero lead times. Furthermore, it is assumed that a demand at a stockpoint is automatically fulfilled when the stockpoint has at least one unit on stock. In our model, on the contrary, we allow that in any state one can choose to apply a lateral transshipment or an emergency order. In other words, if the stockpoint where the demand occurs has positive stock, then one can choose between satisfying the demand directly from the own stock, applying a lateral transshipment or applying an emergency shipment. I.e., our model has more decisions and thus a higher complexity. Our results show that it is not always optimal to choose for directly satisfying a demand from the own stock if possible. This relates to the following point, namely, that we allow for non-equal emergency order costs. This relaxation is important as a consequence of it is, that it may be optimal to reserve a last unit on stock at one stockpoint for a future demand at the other stockpoint. In that case a demand at that stockpoint is not directly satisfied from stock, although the stockpoint has positive stock.

For multiple locations, Archibald et al. [5] come up with heuristics for the lateral transshipment rule. Hu et al. [106] characterize the optimal production and transshipment decisions in a related setting, also assuming that replenishments occur instantaneously at the beginning of a period. Herer et al. [104] approximate the optimal transshipment rule for multiple locations. In Herer and Tzur [103] an optimal transshipment policy is derived, but here lateral transshipments are not used in reaction to stock-outs, but only to balance stock because of different holding and replenishment costs at the locations. These are called pro-active transshipments. Wee and Dada [204] study the decision for a single period in a system with multiple locations and one central warehouse and give five protocols for a transshipment attempt in case of a stock-out. They prove that the optimal transshipment policy is described by exactly one of them, which can be determined by evaluating a set of conditions.

For optimal lateral transshipments rules in a continuous time review setting, hardly any results are available. Zhao et al. [216, 217] prove the optimal transshipment policy for so-called decentralized networks, where the locations are independently owned and operated. They find a policy where two, respectively three parameters determine when to send and when to accept a transshipment request. The latter resembles work of Xu et al. [215], which consider a hold-back parameter, that limits the amount of outgoing transshipments, however, they work with an (Q, R) replenishment policy. For the case of compound Poisson processes, Axsäter [11] comes up with a heuristic rule determining which part of a given demand should be covered by a lateral transshipment. Evers [73] provides two heuristics giving critical values for on-hand inventory, above which a stock transfers should be applied. Minner et al. [144] improve these heuristics, using an approach based on net present value.

In Zhao et al. [218] a two location make-to-stock system is considered. The demands arrive according to two independent Poisson processes, and at each location the production is modeled by a single-server make-to-stock queue, with exponentially distributed production times. The optimal production and optimal transshipment policy are derived, which both can be described by a switching curve, e.g. the optimal decisions are described by state-dependent thresholds. In their model, demands are backordered, and lateral transshipments can be applied both at the moment of a demand and at the moment of a production completion. However, they do not allow inventory to be held back at a location in order to be able to respond to future lateral transshipment requests of the other location, if it is, or is having a large risk of, facing a stock-out. So, the optimization of the lateral transshipment policy in the current setting (positive lead times in a continuous setting, the possibility to hold back stock and the possibility of asymmetric costs parameters) has not been done before.

The inventory model we consider is closely related to a queueing system. By viewing the stockpoints as multi-server queues, demands as arriving customers and repair lead times as service times, the problem translates into a routing problem in a queueing model with two parallel queues. These problems are often modeled as Markov decision problems, as is the case in the current chapter. Stidham and Weber [172] provide an overview on related problems for the control of (networks of) queues. For example, Menich and Serfozo [142] show optimality of a join-the-shortest-queue routing policy. Brouns [42] gives a partial characterization of the optimal routing policy to two two parallel multi-server queues with no buffers, which is related to our work. The main difference is, however, that in these kind of problems from the queueing literature, no costs are incurred for the routing of customers. These costs, however, turn out here to play an important role in the characterization of optimal policies.

4.3 Model and notation

In Section 4.3.1 we introduce the problem, followed by its modeling as a Markov decision problem in Section 4.3.2. We introduce the *value function* (the n-period minimal cost function) and two types of event operators (for the demands and for the repairs), by which the value function can be recursively expressed.

4.3.1 Problem description

We consider a spare part inventory system consisting of two stockpoints, which provision a single spare part for the critical component of an advanced technical system. Initially, each stockpoint has a predetermined number of ready-for-use spare parts in stock of a given stock keeping unit (SKU), $S_i \in \mathbb{N} \cup \{0\}$ at stockpoint i, $i = 1, 2$. There are two groups of systems, where each group is assigned to one stockpoint. When a system breaks down, the critical component has to be replaced by a spare part, i.e., the system demands a spare part at its designated stockpoint. The demands arrive continuously, according to two independent Poisson processes with arrival rate $\lambda_i \geq 0$ at stockpoint i, such that $\lambda_1 + \lambda_2 > 0$. A demand can be fulfilled in one of the following three ways: (i) directly from own stock, (ii) via a lateral transshipment, or (iii) via an emergency procedure. In either of the three cases, the defective part is returned to the stockpoint of which the spare part originated from. The repair times of broken parts are exponentially distributed with mean $1/\mu$, where $\mu > 0$, and both stockpoints have ample repair capacity. As both stockpoints repair the same kind of parts, the repair rates are equal. In Section 4.5.3 we investigate unequal repair rates. We assume that parts can be repaired an unlimited number of times, and that repaired parts attain their original quality. The interarrival and replenishment times are all mutually independent.

Our goal is to minimize the average costs in the long-run. The costs are composed of the costs for the lateral transshipments, emergency procedures, and the downtimes of the systems. We are only interested in the influence of the decisions on the costs, i.e., in the *extra* costs a lateral transshipment or an emergency procedure causes, compared to a fulfillment of a demand directly from stock. The number of spare parts S_1 and S_2 is given, and we do not take into account the acquisition costs of these. Neither do we take into account holding costs, as we have circulating stock. We set the costs when a demand is met directly from own stock to zero. These would be the costs for the downtime of the machine and for the shipment of the spare part to the system, as well as replacing, shipping back and repairing the broken part. But these costs are made in any case, independently of the chosen action. If a lateral transshipment is applied, higher transportation costs are incurred, and as the system is down during the extra transportation time, extra costs for loss of production are incurred too. All these costs together, for applying a lateral transshipment from the other stockpoint to stockpoint i, are put in the penalty costs for a lateral transshipment *to* stockpoint i, denoted by P_{LT_i}. The third option for fulfilling a demand, is an emergency procedure. The broken part is repaired in a fast repair procedure, during which the machine is down. This can be a considerable amount of time. These extra costs form the penalty costs for an emergency procedure for a demand at stockpoint i, denoted by P_{EP_i}. We assume $P_{EP_i} \geq P_{LT_i} \geq 0$, $i = 1, 2$.

We model the delays for lateral transshipments and emergency procedures entirely in the cost factors P_{LT_i} and P_{EP_i}. This is because, compared to the repair lead times, these delays are on a different time scale. We work together with several companies in the spare parts industry and know the typical orders of magnitude for repair lead times, lateral transshipment times, and emergency shipments. Repair lead times are typically in the order of multiple weeks or months, whereas lateral transshipments and emergency shipments are typically in the order of hours or at most one day, say. As a result, it is very unlikely that a normal repair lead time of a part would be completed in the few hours that a lateral transshipment or emergency procedure is executed. Hence, we model the lateral transshipments and emergency procedures to occur instantaneously and put all costs in the factors P_{LT_i} and P_{EP_i}.

We allow for non-equal lateral transshipment and emergency procedure costs at the two stockpoints. The two groups of customers served by the two stockpoints, possibly have different downtime costs. So, as these are incorporated in P_{LT_i} and P_{EP_i}, these cost factors may differ at both locations. This also reflects possible differences in transportation costs.

Under the given repair strategy, we have a system with circulating stock. The inventory position (the total number of parts in stock and parts in repair) is constant at each stockpoint and equal to the initial amount of spare parts, which is S_i at stockpoint i. The model, however, is suitable in a much more general setting, namely with replenishments (or productions to stock) instead of repairs. This holds as long as a base stock policy is executed, where the base stock level is S_i, and one-for-one replenishments (or productions) are assumed. The repair times are then the equivalent of the replenishment (or production) lead times, and the emergency repair procedures are the equivalent of lost sales.

4.3.2 Dynamic programming formulation

The state x of the system is given by the inventory levels at both stockpoints: $x = (x_1, x_2)$, where $x_i \in \{0, 1, \ldots, S_i\}$ is the on-hand stock at stockpoint i. The state space $\mathcal{S}$ is given by all possible combinations of inventory levels, $\mathcal{S} = \{0, 1, \ldots, S_1\} \times \{0, 1, \ldots, S_2\}$. Upon a demand at stockpoint i, a decision has to be taken how to fulfill it, in one of the following three ways: (0) directly from own stock, (1) via a lateral transshipment or (2) via an emergency procedure. The action taken for a demand at i when in state x, is denoted by $a_i(x) \in \{0, 1, 2\}$, respectively, and an optimal action is denoted by $a_i^*(x)$. Backorders are not allowed. Hence, the decision space of each $a_i(x)$ consist of the decisions under which x_1 and x_2 remain greater than or equal to zero.

As the interarrival times of demands as well as the replenishment times are independent exponentially distributed random variables, we can apply uniformization (e.g. Lippman [131]) to convert the semi-Markov decision problem into an equivalent Markov decision problem (MDP).

The existence of a stationary average costs optimal policy is guaranteed by Puterman [155, Theorem 8.4.5a]: if the state space and action space for every state are finite, the costs are bounded and the model is *unichain*, then there exists a stationary average costs optimal policy. A model is said to be unichain if the transition matrix of every (deterministic) stationary policy is unichain, that is, if it consists of a single recurrent class plus a possibly empty set of transient states. The current model is unichain, as the state (S_1, S_2) is accessible from every state $(x_1, x_2) \in \mathcal{S}$ for every stationary policy.

When facing a decision, we should take into account the direct costs for a decision as well as the future expected costs this decision brings along. For the expected costs from a state, we introduce the *value function* (see e.g. Puterman [155]) $V_n : \mathcal{S} \mapsto \mathbb{R}^+$. $V_n(x_1, x_2)$ is the minimum expected total costs when there are n events (demands or repairs) left starting in state $(x_1, x_2) \in \mathcal{S}$. This V_n can be recursively expressed. The two types of operators it consists of (G_i for the repairs at a stockpoint, and H_i for the demands) are defined below. V_n is given by:

$$V_{n+1}(x_1, x_2) = \frac{1}{\nu} \left(\sum_{i=1}^{2} \mu\, G_i\, V_n(x_1, x_2) + \sum_{i=1}^{2} \lambda_i\, H_i\, V_n(x_1, x_2) \right), \text{ for } (x_1, x_2) \in \mathcal{S}, n \geq 0, \tag{4.3.1}$$

starting with $V_0 \equiv 0$, and $\nu = \lambda_1 + \lambda_2 + (S_1 + S_2)\mu$ is the uniformization rate. Decisions are only made in the way of fulfilling demands (in the operator H_i). The decision is taken each time a demand arrives, and is based on the inventory levels. For the repairs no decisions are taken.

Note that we do not include holding costs, as one can, without loss of generality, assume that these are also charged for items in repair. So they only add a constant factor to the value function, and hence, we set them to zero. In Section 4.5.1 we consider including these costs as a model extension.

The operator G_1 models the repairs at stockpoint 1, and is defined by

$$G_1 f(x_1, x_2) = \begin{cases} (S_1 - x_1) f(x_1 + 1, x_2) + x_1 f(x_1, x_2) & \text{if } x_1 < S_1, \\ S_1 f(x_1, x_2) & \text{if } x_1 = S_1, \end{cases} \tag{4.3.2}$$

where f is an arbitrary function $f : \mathcal{S} \mapsto \mathbb{R}^+$. G_2 is defined analogously. If the inventory level is x_1, there are $S_1 - x_1$ outstanding repairs, hence, the repairs occur at a rate proportional to $S_1 - x_1$. The term $x_1 f(x_1, x_2)$ corresponds to *fictitious* transitions. In this way, we assure that the total rate at which G_1 occurs is always equal to S_1.

The operator H_1 models the demands at stockpoint 1, and is defined by

$$H_1 f(x_1, x_2) \\ = \begin{cases} P_{EP_1} + f(x_1, x_2) & \text{if } x_1 = 0, x_2 = 0, \\ \min\{f(x_1 - 1, x_2), P_{EP_1} + f(x_1, x_2)\} & \text{if } x_1 > 0, x_2 = 0, \\ \min\{P_{LT_1} + f(x_1, x_2 - 1), P_{EP_1} + f(x_1, x_2)\} & \text{if } x_1 = 0, x_2 > 0, \\ \min\{f(x_1 - 1, x_2), P_{LT_1} + f(x_1, x_2 - 1), P_{EP_1} + f(x_1, x_2)\} & \text{if } x_1 > 0, x_2 > 0. \end{cases} \tag{4.3.3}$$

H_2 is defined analogously. If a demand occurs, it has to be decided how to fulfill it. There are three options for this: directly from stock, via a lateral transshipment, or via an emergency procedure. H_i takes the costs-minimizing action, where the costs consist of the direct costs for an action and the expected remaining costs from the state the system is in after taking that action. Depending on the stock levels x_1 and x_2, four cases are distinguished over which the minimization is carried out as stock levels cannot become negative.

4.4 Structural results

In this section we prove our main result: the structure of the optimal policy. For this we first prove that the value function V_n satisfies certain structural properties, such as monotonicity and multimodularity. We show that the operators of which V_n is composed, all preserve these properties. Then, as V_0 satisfies them, it follows directly by induction that the properties hold for V_n for all $n \geq 0$. A framework for this was introduced by Koole [115] (see also Koole [116]) as *Event Based Dynamic Programming*. The main advantage of this approach is that one can prove the propagation of properties for each of the event operators separately, reducing the complexity of the problem. Changes or extensions to the model can easily be made by replacing or adding events.

In Section 4.4.1 we introduce the properties and prove that G_1, G_2, H_1 and H_2 preserve them. It then follows that V_n, for all $n \geq 0$, satisfies them as well. From this we derive, in Section 4.4.2, the structure of the optimal lateral transshipment policy, which is a threshold type policy. We give conditions under which it reduces to a simple policy, such as a hold back or a complete pooling strategy in Section 4.4.3. Some examples are given in Section 4.4.4, and the special case with symmetric system parameters is considered in Section 4.4.5. All proofs are given in the appendix.

4.4.1 Properties of operators and value function

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all x such that the states appearing in the right-hand and left-hand side of the inequalities exist in $\mathcal{S}$:

$$\text{Decr}(1): \quad f(x_1,x_2) \geq f(x_1+1,x_2), \tag{4.4.1}$$
$$\text{Decr}(2): \quad f(x_1,x_2) \geq f(x_1,x_2+1), \tag{4.4.2}$$
$$\text{Conv}(1): \quad f(x_1,x_2)+f(x_1+2,x_2) \geq 2f(x_1+1,x_2), \tag{4.4.3}$$
$$\text{Conv}(2): \quad f(x_1,x_2)+f(x_1,x_2+2) \geq 2f(x_1,x_2+1), \tag{4.4.4}$$
$$\text{Supermod}: \quad f(x_1,x_2)+f(x_1+1,x_2+1) \geq f(x_1+1,x_2)+f(x_1,x_2+1), \tag{4.4.5}$$
$$\text{SuperC}(1,2): \quad f(x_1+2,x_2)+f(x_1,x_2+1) \geq f(x_1+1,x_2)+f(x_1+1,x_2+1), \tag{4.4.6}$$
$$\text{SuperC}(2,1): \quad f(x_1,x_2+2)+f(x_1+1,x_2) \geq f(x_1,x_2+1)+f(x_1+1,x_2+1), \tag{4.4.7}$$
$$\text{MM}: \quad \text{Supermod} \cap \text{SuperC}(1,2) \cap \text{SuperC}(2,1). \tag{4.4.8}$$

Furthermore, Decr = Decr(1) ∩ Decr(2), Conv = Conv(1) ∩ Conv(2), and SuperC = SuperC(1,2) ∩ SuperC(2,1).

The following two lemmas give useful properties of the operators G_i and H_i, which enable us to derive the structure of the optimal policy.

LEMMA 4.4.1. *a) Operator G_i, $i = 1,2$, preserves each of the following properties:*
(i) Decr; (ii) Conv; (iii) Supermod.
b) The sum of the operators $G_1 + G_2$ preserves each of the following properties:
(i) Decr; (ii) Conv; (iii) Supermod; (iv) SuperC; (v) MM.

Note that SuperC (and hence MM), is only preserved by the sum of the operators $G_1 + G_2$, and not by G_1 and G_2 separately. When $G_1 + G_2$ is applied to (4.4.6) and (4.4.7) respectively, some terms introduced by G_1 cancel out against terms introduced by G_2.

LEMMA 4.4.2. *Operator H_i, $i = 1,2$, preserves each of the following properties:*
(i) Decr, (ii) MM.

By induction on n, and using the results of Lemmas 4.4.1 and 4.4.2, the next theorem immediately follows.

THEOREM 4.4.3. *V_n satisfies* (4.4.1)–(4.4.7) *for all $n \geq 0$.*

The properties (4.4.1)–(4.4.7) of V_n are the key in classifying the structure of the optimal policy.

4.4.2 Structure of optimal policy

We now characterize the structure of the optimal policy in the following two theorems. We state the optimal policy for fulfilling a demand at stockpoint 1 (see Figure 4.1); for stockpoint 2, analogous results hold. First we give the result for x_2 fixed, next for x_1 fixed.

THEOREM 4.4.4. *The optimal policy for fulfilling a demand at stockpoint 1 for fixed x_2 is a threshold type policy: for each $x_2 \in \{0,1,\ldots,S_2\}$, there exist thresholds $T_1^{lt}(x_2) \in \{0,1,\ldots,S_1+1\}$ and $T_1^{di}(x_2) \in \{1,\ldots,S_1+1\}$, with $T_1^{lt}(x_2) \leq T_1^{di}(x_2)$, such that:*

$$a_1^*(x) = 2 \text{ (emergency procedure), for } 0 \leq x_1 \leq T_1^{lt}(x_2) - 1;$$
$$a_1^*(x) = 1 \text{ (lateral transshipment), for } T_1^{lt}(x_2) \leq x_1 \leq T_1^{di}(x_2) - 1;$$
$$a_1^*(x) = 0 \text{ (directly from own stock), for } T_1^{di}(x_2) \leq x_1 \leq S_1,$$

where $T_1^{lt}(0) = T_1^{di}(0) \geq 1$.
The analogous result holds for demands at stockpoint 2 under a fixed $x_1 \in \{0,1,\ldots,S_1\}$.

This structure is graphically represented below the horizontal axis in Figure 4.1. For each x_2, the thresholds divide the set $\{0,\ldots,S_1\}$ into (at most) three subsets. In the first subset, where x_1 is small, an emergency procedure is optimal; in the second one a lateral transshipment; and in the third one, where x_1 is large, it is optimal to take a part from stock. A threshold can be equal to S_1+1, hence, implying that for a given x_2 taking parts from stock or applying lateral transshipments is never optimal.

A special case is $x_2 = 0$: as lateral transshipments are not possible at stockpoint 1, we have $T_1^{lt}(0) = T_1^{di}(0)$, where $T_1^{di}(0) \geq 1$. In this case, there are (at most) two subsets: an emergency procedure is applied for $0 \leq x_1 < T_1^{di}(0)$, and a demand is directly delivered from stock for $T_1^{di}(0) \leq x_1 \leq S_1$.

The intuition behind this theorem is as follows. If the stock level x_1 is high one is willing to take a part from stock as there are still plenty left afterwards. But if the stock level is low, one might, depending on the costs parameters, decide to hold some parts back for future lateral transshipment requests of the other stockpoint. If $x_1 = 0$ one is forced to apply either an emergency procedure or a lateral transshipment.

A similar characterization of the optimal policy can be made for fixed x_1, which is given in the following theorem.

THEOREM 4.4.5. *For the optimal policy for fulfilling a demand at stockpoint 1 for fixed $x_1 \in \{0,1,\ldots,S_1\}$, there exist $\hat{T}_1^{di}(x_1) \in \{0,1,\ldots,S_2+1\}$ and $\hat{T}_1^{lt}(x_1) \in \{1,\ldots,S_2+1\}$, with $\hat{T}_1^{di}(x_1) \leq \hat{T}_1^{lt}(x_1)$, such that:*

$$a_1^*(x) = 2 \text{ (emergency procedure), for } 0 \leq x_2 \leq \hat{T}_1^{di}(x_1) - 1;$$
$$a_1^*(x) = 0 \text{ (direct from own stock), for } \hat{T}_1^{di}(x_1) \leq x_2 \leq \hat{T}_1^{lt}(x_1) - 1;$$
$$a_1^*(x) = 1 \text{ (lateral transshipment), for } \hat{T}_1^{lt}(x_1) \leq x_2 \leq S_2,$$

where $\hat{T}_1^{di}(0) = \hat{T}_1^{lt}(0) \geq 1$.
The analogous result holds for demands at stockpoint 2 under a fixed $x_2 \in \{0,1,\ldots,S_2\}$.

This structure is graphically represented next to the vertical axis in Figure 4.1. For a given x_1, the set $\{0,\ldots,S_2\}$ is divided into subsets, such that in each subset one of the three decisions is optimal. Again, a $\hat{T}_1^{di}(x_1)$ or $\hat{T}_1^{lt}(x_1)$ larger than the maximum stock level indicates that a certain subset is empty, hence, that decision is never optimal. A special case is $x_1 = 0$, when it is not possible to deliver a demand directly from stock. Hence $\hat{T}_1^{di}(0) = \hat{T}_1^{lt}(0)$, where $\hat{T}_1^{lt}(0) \geq 1$.

If the stock level at the other stockpoint, x_2, is high, a lateral transshipment can be a good option as there are still plenty of parts left after the transshipment is carried

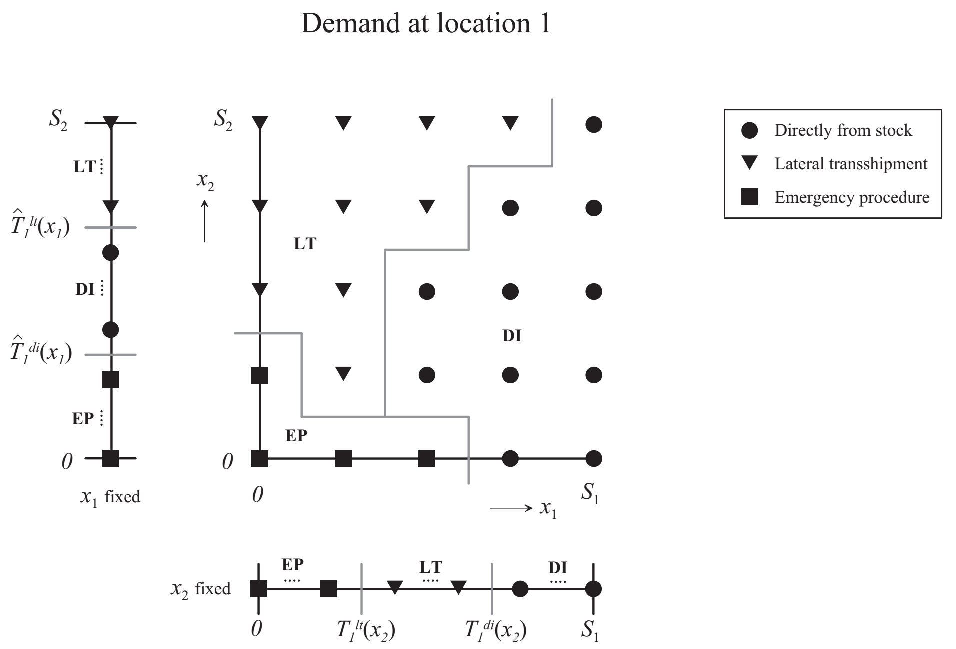


Figure 4.1: General structure of the optimal policy for a demand at location 1. For fixed x_2 the optimal policy structure is indicated below the horizontal axis, for fixed x_1 next to the vertical axis.

out. When x_2 decreases, lateral transshipments are less likely to become the best option. If x_2 is low, or even zero, stockpoint 1 might hold stock back by applying emergency procedures, which can be optimal if the emergency costs of stockpoint 2 are much higher than those of 1. We note that this is the general form of the structure. It is unlikely that it turns out to be optimal to take parts via lateral transshipments when x_2 is large, but hold parts back for stockpoint 2 when x_2 is small.

Combining Theorem 4.4.4 and Theorem 4.4.5 restricts the possibilities for the optimal policy significantly. The states where an emergency procedure is an optimal action for a demand at stockpoint 1, i.e. the subset $\mathrm{EP} = \{x \in \mathcal{S} \mid a_1^*(x) = 2\}$, form a connected part of the state space, located in the lower left corner. This follows as given that $a_1^*(\tilde{x}) = 2$ for some $\tilde{x}$, we have $a_1^*(x) = 2$ for all x with $x_1 \leq \tilde{x}_1$ (by Theorem 4.4.4), and all x with $x_2 \leq \tilde{x}_2$ (by Theorem 4.4.5). For the remaining states, a lateral transshipment or a delivery from stock is optimal. The curve dividing these two subsets, $\mathrm{LT} = \{x \in \mathcal{S} \mid a_1^*(x) = 1\}$ respectively $\mathrm{DI} = \{x \in \mathcal{S} \mid a_1^*(x) = 0\}$, is non-decreasing in x_1. This implies that the general structure of the optimal policy is as given in Figure 4.1.

4.4.3 Conditions simplifying the optimal policy

Under simple, sufficient conditions for the cost parameters, the structure of the optimal policy is simplified. We give two such conditions: under the first one, (i) it is optimal to fulfill a demand directly from own stock, whenever possible, but with a parameter limiting the amount of outgoing lateral transshipment. We refer to this as a *hold back* *policy* (see e.g. Xu et al. [215]) as the parameters indicate the amount of stock that is held back from a transshipment request. Hence, we refer to these as the *hold back levels*. Next to this, under the second condition, (ii) it is optimal to fulfill a demand directly from own stock, whenever possible, and otherwise to apply a lateral transshipment, whenever possible. For an individual stockpoint, we call this a *zero hold back policy*, as the hold back level equal zero. When both stockpoints execute this policy, this is called a *complete pooling* policy.

The following theorem states conditions under which it is optimal to always fulfill a demand directly from own stock:

THEOREM 4.4.6. *1a) If*

$$P_{EP_2} \leq P_{LT_2} + \left(1 + \frac{\mu}{\lambda_2}\right) P_{EP_1}, \tag{4.4.9}$$

then $T_1^{di}(x_2) = 1$ *for all* $x_2 \in \{0, 1, \ldots, S_2\}$*, i.e. a hold back policy is optimal at stockpoint 1.*
b) If

$$P_{EP_1} \leq P_{LT_1} + \left(1 + \frac{\mu}{\lambda_1}\right) P_{EP_2}, \tag{4.4.10}$$

then $T_2^{di}(x_1) = 1$ *for all* $x_1 \in \{0, 1, \ldots, S_1\}$*, i.e. a hold back policy is optimal at stockpoint 2.*
2) If (4.4.9) *and* (4.4.10) *hold, then it is optimal for both stockpoints to execute a hold back policy.*

Under condition (4.4.9), whenever there are items in stock at stockpoint 1, they should always be used in case of a demand at stockpoint 1, see Figure 4.2(a). However, stock can possibly be held back from lateral transshipment requests. If both stockpoints execute a hold back policy, the entire policy is prescribed by only 2 parameters ($\hat{T}_1^{lt}(0)$ and $\hat{T}_2^{lt}(0)$). The case of symmetric costs at both stockpoints, i.e. $P_{LT_1} = P_{LT_2}$ and $P_{EP_1} = P_{EP_2}$, clearly satisfies conditions (4.4.9) and (4.4.10).

Next we give conditions under which the application of lateral transshipments in case of a stock-out, is optimal, when possible:

THEOREM 4.4.7. *1a) If*

$$P_{LT_1} + \frac{\lambda_2}{\lambda_2 + \mu} P_{EP_2} \leq P_{EP_1}, \tag{4.4.11}$$

then $\hat{T}_1^{lt}(0) = 1$*, i.e. a zero hold back policy is optimal at stockpoint 1.*
1b) If

$$P_{LT_2} + \frac{\lambda_1}{\lambda_1 + \mu} P_{EP_1} \leq P_{EP_2}, \tag{4.4.12}$$

then $\hat{T}_2^{lt}(0) = 1$*, i.e. a zero hold back policy is optimal at stockpoint 2.*
2) If (4.4.11) *and* (4.4.12) *hold, then a complete pooling policy is optimal.*

Note that conditions (4.4.11) and (4.4.12) are symmetric in their arguments. Under condition (4.4.11), stockpoint 2 should not hold back stock if stockpoint 1 requests a lateral transshipment when it is out-of-stock, see Figure 4.2(b). As condition (4.4.11) is stronger than condition (4.4.9), i.e. as (4.4.11) implies (4.4.9), it follows that under condition (4.4.11) a zero hold back pooling policy is optimal, see Figure 4.2(c). Under a zero hold back pooling policy, a demand is directly met from own stock if possible, and otherwise always via a lateral transshipment, if possible. The stockpoint does not

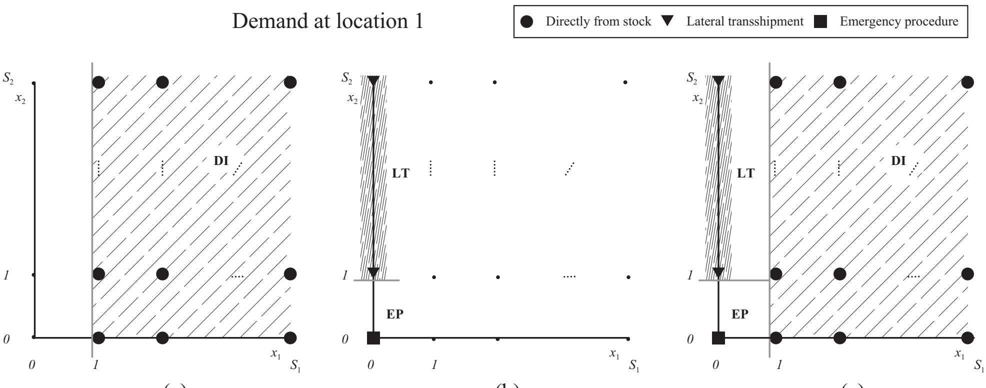


Figure 4.2: (a) Always DI (hold back policy, Theorem 4.4.6); (b) Always LT if out-of-stock (Theorem 4.4.7); (c) Complete pooling (again Theorem 4.4.7, as it implies Theorem 4.4.6).

hold back stock in any case. When both stockpoints execute this strategy, this is called a complete pooling policy, a strategy that is often assumed in the literature (see e.g. [2, 10, 90, 123, 126, 165, 174, 212, 218] to mention only a few). Theorem 4.4.7 gives sufficient conditions under which such a policy is indeed optimal.

The implication of (4.4.9) by (4.4.11) can be seen as follows. Rewriting (4.4.11) gives $\frac{\lambda_2}{\lambda_2+\mu}P_{EP_2} \leq P_{EP_1} - P_{LT_1}$, but this implies $\frac{\lambda_2}{\lambda_2+\mu}P_{EP_2} \leq P_{EP_1} + \frac{\lambda_2}{\lambda_2+\mu}P_{LT_2}$ (as both P_{LT_1} and $\frac{\lambda_2}{\lambda_2+\mu}P_{LT_2}$ are non-negative), which is equivalent to (4.4.9). Analogously, (4.4.12) implies (4.4.10).

The given conditions in Theorem 4.4.6 and Theorem 4.4.7 are, in general, sufficient, but not necessary. For the cases $S_1 = 1, S_2 = 0$, respectively $S_1 = 0, S_2 = 1$, the conditions are necessary and sufficient. There exist examples *not* satisfying these conditions, in which case, the optimal policy is neither a hold back nor a zero hold back pooling policy (see Section 4.4.4, Example 4.4.1).

There is an interesting relation between the conditions when considered for both stockpoints: either condition (4.4.9), or condition (4.4.12) holds (or both hold); and either condition (4.4.10), or condition (4.4.11) holds (or both hold). These statements follow from the following (e.g. for the first one): (i) if condition (4.4.9) does *not* hold, then surely condition (4.4.12) holds; and (ii) if condition (4.4.12) does *not* hold, then surely condition (4.4.9) holds. This follows by rewriting the conditions: for (i) we have that if (4.4.9) does not hold, then $P_{EP_2} \geq P_{LT_2} + \frac{\lambda_2+\mu}{\lambda_2}P_{EP_1}$, but this implies $P_{EP_2} \geq P_{LT_2} + \frac{\lambda_1}{\lambda_1+\mu}P_{EP_1}$ (as $\frac{\lambda_2+\mu}{\lambda_2} \geq 1$, but $\frac{\lambda_1}{\lambda_1+\mu} \leq 1$), which is exactly (4.4.12); and (ii) follows as $\frac{\lambda_1}{\lambda_1+\mu} \leq 1$ in (4.4.12), but $1 + \frac{\mu}{\lambda_2} \geq 1$ in (4.4.9). The analogous reasoning holds for conditions (4.4.10) and (4.4.11). Combined with the properties that (4.4.11) implies (4.4.9), and that (4.4.12) implies (4.4.10), this immediately leads to the following corollary.

COROLLARY 4.4.8. *The optimal lateral transshipment policy is*

1. *either (at least) a hold back policy at both locations;*
2. *or a zero hold back policy for (at least) one location.*

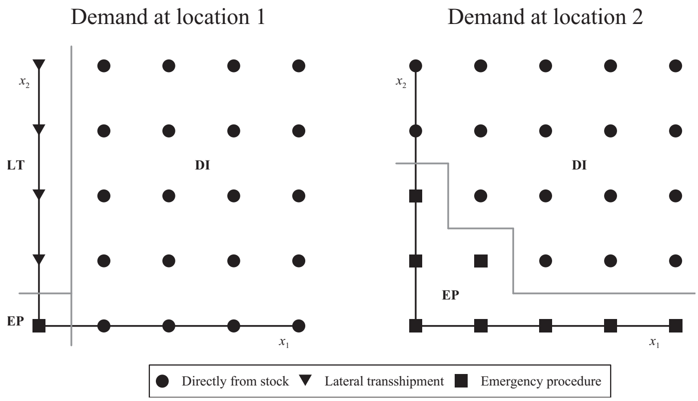


Figure 4.3: Optimal policy for the case with $S_1 = S_2 = 4, \lambda_1 = 2, \lambda_2 = 1,\ \mu = 1/3$ *and penalty costs* $P_{EP_1} = 25, P_{LT_1} = 5, P_{EP_2} = 10, P_{LT_2} = 2.$

Here, by 'at least' a hold back policy we mean either a hold back policy or a zero hold back policy. In the second case, the optimal policy for one location is a zero hold back policy, and the optimal policy for the other location can be a hold back policy, a zero hold back policy, or neither of the two.

4.4.4 Examples

We illustrate our results by two examples.

Example 4.4.1. Consider the following example: $S_1 = S_2 = 4$, and $\lambda_1 = 2$, $\lambda_2 = 1$, $\mu = 1/3$, and cost parameters given by $P_{EP_1} = 25$, $P_{LT_1} = 5$ and $P_{EP_2} = 10$, $P_{LT_2} = 2$. Hence, an emergency procedure is five times as expensive as a lateral transshipment, and at location 1, the demand rate as well as the costs are higher. The optimal policy is given in Figure 4.3.

At stockpoint 1 a zero hold back policy is optimal: the demands are fulfilled directly from own stock if possible ($a_1^*(x_1, x_2) = 0$ for $x_1 > 0$), or via a lateral transshipment in case of a stock-out ($a_1^*(0, x_2) = 1$, for $x_2 > 0$). Only if stockpoint 2 is stocked-out as well, an emergency procedure is applied ($a_1^*(0, 0) = 2$). This structure is implied by the fact that the parameters satisfy condition (4.4.11), and hence, part 1a) of Theorem 4.4.7 holds.

For stockpoint 2, no further conclusions can be drawn for the optimal policy. Demands are only fulfilled directly from stock if the sum of the inventory levels at both locations is large enough, i.e. if $x_1 + x_2 \geq 3$ (and $x_2 > 0$), otherwise an emergency procedure is applied. This can be explained in the following way. The costs for lateral transshipments to and emergency procedures at stockpoint 1 are much higher than those at stockpoint 2. This results in the fact that stockpoint 2 will hold back parts, even when it faces a demand. By holding back parts, the expensive costs for an emergency procedure at stockpoint 1 are saved in case of a demand there, when it is stocked out. This is at the expense of a lateral transshipment from 2 and possibly one or more emergency procedures at 2. The option of holding back parts at 2, however, is on average less costly.

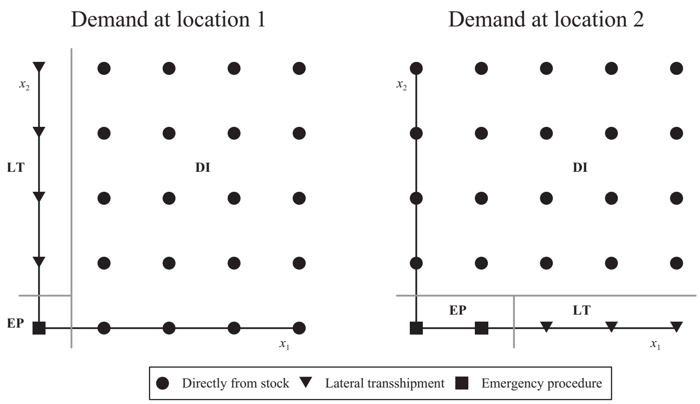


Figure 4.4: Optimal policy for Example 4.4.1, now with $P_{EP_2} = 20$, $P_{LT_2} = 4$.

The optimal policy at location 2 resembles a so-called *critical level* policy, which is common in stock rationing problems for single inventory point (see e.g. Ha [94]). In such problems, one stockpoint satisfies demands of two (or more) types of customers, which differs in penalty costs for lost sales. In the typical optimal policy, demands from the most expensive customers are always satisfied, and a threshold (called the *critical level*) exists for the inventory level. From this level on demands for the less expensive customers should be satisfied as well. In the optimal policy in this example the sum of the stock levels at 1 and 2 resembles such a critical level. The sum x_1+x_2 here determines whether a demand at location 2 is directly satisfied or lost.

The optimal policy gives expected average costs per time unit of 18.2. Without lateral transshipments, these costs would be 25.5; hence, the optimal policy reduces this by almost 29%. A complete pooling policy has expected average costs per time unit of 20.0, in this case, so the optimal policy reduces these costs by 9.4%.

Example 4.4.2. In Example 4.4.1, condition (4.4.11) (and hence, condition (4.4.9)) was satisfied for stockpoint 1, but not for stockpoint 2. By doubling the penalty costs at 2, into $P_{EP_2} = 20$ and $P_{LT_2} = 4$, condition (4.4.10) is satisfied as well. Hence, by Theorem 4.4.6, this results in the optimality of a hold back policy at both locations (with still zero hold back at 1). The optimal policy is given in Figure 4.4.

The two thresholds (the *hold back levels*), determine the entire policy, and are given by $\hat{T}_1^{lt}(0) = 1$ and $\hat{T}_2^{lt}(0) = 2$. These are the inventory levels from which on lateral transshipments ($a_i^* = 1$) are applied instead of emergency procedures ($a_i^* = 2$). The expected average costs per time unit in this case are 22.9. For a policy without lateral transshipments these would be 27.6 (almost 17% reduction for optimal policy), and complete pooling would give 23.2. This is only 1.4% reduction, but this policy differs from the optimal policy only in $a_2(1,0)$.

4.4.5 Symmetric parameters

A special case is the system in which all parameters are symmetric, i.e. in which all parameters for both stockpoints are equal: $S_1 = S_2 =: S, \lambda_1 = \lambda_2 =: \lambda, P_{LT_1} = P_{LT_2} =: P_{LT}, P_{EP_1} = P_{EP_2} =: P_{EP}$. It is straightforward that in this case there exists a symmetric

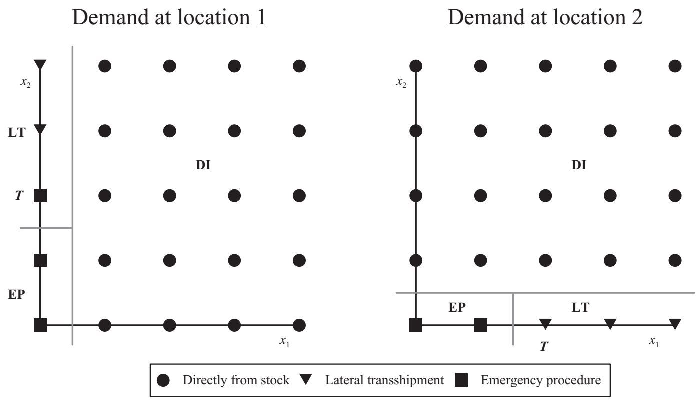


Figure 4.5: For symmetric system parameters, the optimal policy can be described by only one threshold T (e.g., here $T = 3$).

optimal policy. As the conditions of Theorem 4.4.6 are clearly satisfied, it follows that for both stockpoints a hold back policy is optimal. The entire policy can now be described by a single (for both stockpoints equal) hold back level $\hat{T}_1^{lt}(0) = \hat{T}_2^{lt}(0) =: T \in \{1, 2, \ldots, S+1\}$, see Figure 4.5. $T = 1$ indicates zero hold back, $T = 2$ indicates that one part is held back, and so on, and $T = S + 1$ indicates that no stock in shared in any way, i.e. there is no interaction between the stockpoints. Hence, there are only $S + 1$ possible optimal policies.

Given λ/μ, it turns out that the optimal policy is now determined by only the ratio P_{LT}/P_{EP}. For $S = 4$ it is indicated in Figure 4.6a when each of the five possible policies is optimal. These areas are determined by solving the steady-state distribution of the Markov process for each of the policies and deriving the average costs of a policy from this.

For the symmetric case Theorem 4.4.7 reduces to the following corollary, which also holds when $S_1 \neq S_2$.

COROLLARY 4.4.9. *In case of symmetric system parameters, and if*

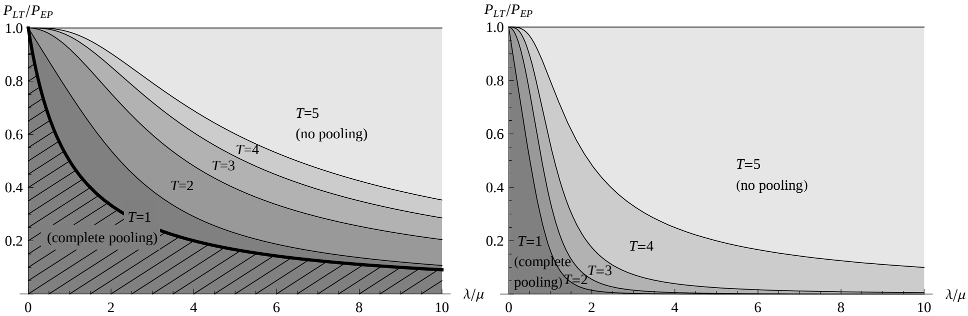
$$P_{LT} \leq \frac{\mu}{\lambda + \mu} P_{EP}, \tag{4.4.13}$$

a complete pooling policy is optimal.

In Figure 4.6a the curve $P_{LT}/P_{EP} = \mu/(\lambda + \mu)$ is plotted as well. Below it complete pooling is optimal. From this figure it turns out that this condition, although only sufficient, covers a large part of the total, exact area.

4.5 Model extensions

In this section we consider several model extensions. First we incorporate holding costs, which would be of interest if we interpret the model as an inventory system with replenishments. The general structure still holds, but the conditions do change. Secondly,



(a) Ample repair capacity, and condition (4.4.13).

(b) Single repair server.

Figure 4.6: For $S = 4$ and symmetric system parameters, the $S + 1$ regions where each of the thresholds T is optimal.

we restrict lateral transshipments to be carried out in only one direction. This is a suitable model for a substitution problem, which fits exactly within the lateral transshipment model. Then we investigate the case of unequal repair rates, and we study the case where the repair capacity is limited. Finally, we add the option of so-called *proactive lateral transshipment*, which are lateral transshipments not triggered by a demand. Note that each of the extensions is made to the basic model, although it is possible to combine some of them.

4.5.1 Holding costs

The model was presented for an inventory system with repairables. It is, however, also suitable when a base stock policy is executed, either with replenishments or productions to stock. The base stock level then is S_i. For repairables one can, without loss of generality, charge holding costs for items in repair. Hence, as the inventory position is constant, this constant factor can be left out of the value function. However, if we would include it, the definition of the value function (4.3.1) becomes:

$$V_{n+1}(x_1, x_2) = \sum_{i=1}^{2} h_i(x_i) + \frac{1}{\nu}\left(\sum_{i=1}^{2} \mu\, G_i\, V_n(x_1, x_2) + \sum_{i=1}^{2} \lambda_i\, H_i\, V_n(x_1, x_2)\right), \qquad (4.5.1)$$

for $(x_1, x_2) \in \mathcal{S}$ and $n \geq 0$, where $h_i : \{0, 1, \ldots, S_i\} \mapsto \mathbb{R}$ are the holding costs at stockpoint i per part per time unit. We assume that $h_i(0) = 0$ and that $h_i(x_i)$ is non-decreasing and concave in x_i. Recall that $\nu = \lambda_1 + \lambda_2 + (S_1 + S_2)\mu$.

It is easily checked that Lemmas 4.4.1 and 4.4.2 still hold, except Decr. Hence, Theorems 4.4.3, 4.4.4 and 4.4.5 still hold (as Decr is not used in the proofs). In the conditions under which the (zero) hold back policy is optimal, however, an extra term incorporation the holding costs should be added.

Analogously to inequalities (4.4.9) and (4.4.10) of Theorem 4.4.6, if

$$P_{EP_2} \leq P_{LT_2} + \left(1 + \frac{\mu}{\lambda_2}\right) P_{EP_1} + \frac{h_1(1)}{\nu\, \lambda_2},$$

respectively if

$$P_{EP_1} \leq P_{LT_1} + \left(1 + \frac{\mu}{\lambda_1}\right) P_{EP_2} + \frac{h_2(1)}{\nu\,\lambda_1}$$

a hold back policy is optimal at stockpoint 1, respectively at stockpoint 2. If both conditions hold, it is optimal for both stockpoints to execute a hold back policy.

Analogously to inequalities (4.4.11) and (4.4.12) of Theorem 4.4.7, if

$$P_{LT_1} + \frac{\lambda_2}{\lambda_2+\mu} P_{EP_2} \leq P_{EP_1} + \frac{h_2(1)}{\nu\,(\lambda_2+\mu)},$$

respectively if

$$P_{LT_2} + \frac{\lambda_1}{\lambda_1+\mu} P_{EP_1} \leq P_{EP_2} + \frac{h_1(1)}{\nu\,(\lambda_1+\mu)}$$

a zero hold back policy is optimal at stockpoint 1, respectively at stockpoint 2. If both conditions hold, a complete pooling policy is optimal.

4.5.2 Unidirectional transshipments and substitutions

In the model we allow transshipments from both stockpoint 1 to 2, as well as vice versa. A simplification of this is an *unidirectional transshipment* (c.f. Axsäter [12]) in which case a lateral transshipment can take place in only one way, say only from 1 to 2. An application of this is a problem with substations, where the parts of stockpoint 1 can substitute those of stockpoint 2, but not the other way around. Another example is a single stockpoint with two types of SKUs on stock, in which one can serve as a substitute of the other. This might be a way to deal with stock-outs in such a system. A substitution might bring along some extra costs, which are the P_{LT_1}.

The restriction to unidirectional transshipments can be achieved by putting $P_{LT_2} = P_{EP_2}$. In this way a lateral transshipment will never be an optimal action at stockpoint 2 (as it is as expensive as an emergency procedure, but does not reduce the stock level at 1). Obviously, all structural result will remain to hold.

If $P_{LT_2} = P_{EP_2}$ inequality (4.4.9) is always satisfied, and so, as is to be expected, a hold back policy is optimal at stockpoint 1. Note that although (4.4.12) is never satisfied, we can *not* conclude that a zero hold back policy is suboptimal at stockpoint 1. This is because the conditions in Theorems 4.4.6 and 4.4.7 are sufficient but not necessary.

4.5.3 Asymmetric repair rates

As both stockpoints repair the same type of parts, we have taken their repair rates to be equal. In fact, we need this assumption in order to be able to derive the structural results. This is because according to Lemma 4.4.1 only $G_1 + G_2$ preserves MM, and not G_1 and G_2 individually. This is a sufficient assumption but not a necessary one, as we can easily construct examples with unequal repair rates for which the structural results *do* hold. However, there are also examples with unequal repair rates for which the structural results fail to hold, as shown in the following example.

Let $S_1 = 1, S_2 = 2, \lambda_1 = \lambda_2 = 1$, and $\mu_1 = 1/3 \neq \mu_2 = 1$, denoting by μ_i the repair rate at stockpoint i. Furthermore, let $P_{EP_1} = 1000, P_{LT_1} = 175$ and $P_{EP_2} = P_{LT_2} = 10$. The

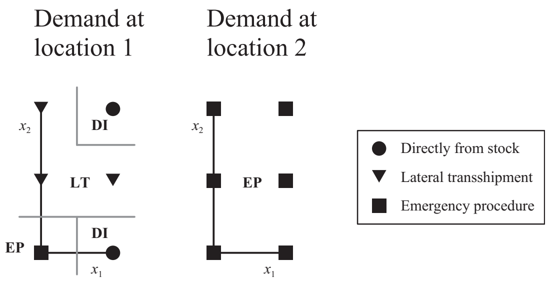


Figure 4.7: Example in which $\mu_1 \neq \mu_2$: the structural properties of the optimal policy does not hold.

(unique) optimal policy is given in Figure 4.7. Clearly, for demands at stockpoint 1 when $x_1 = 1$, the structure of the optimal policy is not the threshold type policy as described by Theorem 4.4.4. This example illustrates that if $\mu_1 \neq \mu_2$ the structural results do not necessarily have to hold.

4.5.4 Limited repair capacity

A variant of the described system is a system in which there is limited repair capacity: at each stockpoint there is only one server to repair the returned parts. The repair times remain exponentially distributed with mean $1/\mu_i$ at stockpoint i, where, for generality, we allow for non-identical repair rates at both locations. We only have to change the operator G_i into, say, $\tilde{G}_i$, where

$$\tilde{G}_1 f(x_1, x_2) = \begin{cases} f(x_1+1, x_2) & \text{if } x_1 < S_1, \\ f(x_1, x_2) & \text{if } x_1 = S_1, \end{cases} \tag{4.5.2}$$

and $\tilde{G}_2$ analogously. The value function $\tilde{V}_n$ becomes

$$\tilde{V}_{n+1}(x_1, x_2) = \frac{1}{\tilde{\nu}} \left(\sum_{i=1}^{2} \mu_i \tilde{G}_i \tilde{V}_n(x_1, x_2) + \sum_{i=1}^{2} \lambda_i H_i \tilde{V}_n(x_1, x_2) \right), \text{ for } (x_1, x_2) \in \mathcal{S}, n \geq 0,$$

with $\tilde{V}_0 \equiv 0$, $\tilde{\nu} = \lambda_1 + \lambda_2 + \mu_1 + \mu_2$, and $\mathcal{S}$ and H_i, $i = 1, 2$, unchanged. The following holds for $\tilde{G}_j$:

Lemma 4.5.1. *The operator $\tilde{G}_j$, $j = 1, 2$, preserves each of the following properties: (i) Decr; (ii) Conv and Decr(j); (iii) Supermod; (iv) SuperC and Conv($j+1$); (v) MM and Conv($j+1$).*

Here $j+1$ should be read as $j+1$ mod 2, and e.g. *(ii)* states that if f is Conv *and* Decr(j), then $\tilde{G}_j f$ is so as well. However, it does *not* hold that f Conv implies $\tilde{G}_j f$ Conv.

Corollary 4.5.2. *$\tilde{V}_n$ satisfies* (4.4.1)–(4.4.7) *for all $n \geq 0$.*

The following theorem is a direct consequence of this corollary.

THEOREM 4.5.3. *In case of a single repair server at both stockpoints, the same structural results for the optimal policy hold for this system, i.e. Theorem 4.4.4 and Theorem 4.4.5 still hold, even if $\mu_1 \neq \mu_2$.*

For Theorem 4.5.3 we do not need equal μ_i's, as $\tilde{G}_1$ and $\tilde{G}_2$ separately preserve MM, and not only the sum of both. For symmetric system parameters, we compare the optimal policy for a single repair server (see Figure 4.6b) with the case of ample repair capacity (see Figure 4.6a). From the graphs it follows that the set of system parameters where one can benefit from lateral transshipments, is much smaller in the case of a single repair server.

4.5.5 Proactive lateral transshipments

We can include pro-active lateral transshipments (i.e. lateral transshipments not triggered by a demand) in our two location inventory model. For this, we have to define an operator modeling such an LT. We show that this operator propagates MM, so all structural results remain valid, and we prove that the optimal policy for proactive LTs is a threshold type policy. We assume that, when a proactive LT is carried out, that we ship a part in repair in the other direction. In this way, we the inventory positions S_1, S_2 remain constant. Hence, a proactive LT from location 2 to 1 is possible when $x_2 > 0$ and $x_1 < S_1$, and analogously for a shipment from 1 to 2.

Let P_i^{PA} be the costs for a pro-active LT to location i, and let transshipment time be exponentially distributed with mean $1/\tau_i$. Let F_i be the operator for a pro-active LT to location i, for $i = 1$ defined by:

$$F_1 f(x_1, x_2) \begin{cases} \min\{P_1^{PA} + f(x_1 + 1, x_2 - 1), \\ \qquad f(x_1, x_2)\} & \text{if } x_1 > 0 \text{ and } x_2 < S_2; \\ f(x_1, x_2) & \text{otherwise,} \end{cases}$$

and F_2 defined analogously.

LEMMA 4.5.4. *For $i = 1, 2$:*

$$F_i : MM \to MM.$$

Including pro-active LTs, the value function becomes:

$$V_{n+1}^F(x_1, x_2) = \frac{1}{\nu^F}\left(\sum_{i=1}^{2} \mu\, G_i\, V_n^F(x_1, x_2) + \sum_{i=1}^{2} \lambda_i\, H_i\, V_n^F(x_1, x_2) + \sum_{i=1}^{2} \tau_i\, F_i\, V_n^F(x_1, x_2)\right),$$

for $(x_1, x_2) \in \mathcal{S}$, $n \geq 0$, again starting with $V_0^F \equiv 0$, where $\nu^F = \lambda_1 + \lambda_2 + S_1\mu_1 + S_2\mu_2 + \tau_1 + \tau_2$ in the uniformization rate.

THEOREM 4.5.5. *1) V^F is MM for all $n \geq 0$.*
2) The optimal policy for applying proactive LTs to stockpoint 1 is described by a switching curve $T_1^{pa}(x_1)$, such that a proactive LT is optimal if $x_2 \geq T_1^{pa}(x_1)$, and suboptimal otherwise. Furthermore, $T_1^{pa}(x_1)$ is increasing in x_1.
The analogous result holds for proactive LTs to stockpoint 2.

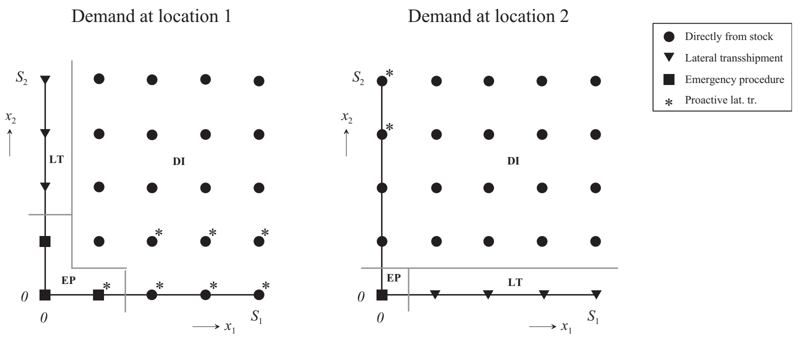


Figure 4.8: Optimal policy, including proactive lateral transshipments, for Example 4.5.1.

Example 4.5.1. Consider an example with $S_1 = S_2 = 4$ and the following parameters: $\lambda_1 = \lambda_2 = 2$, $\mu_1 = \mu_2 = 1$, $\tau_1 = \tau_2 = 5$, and cost parameters $P_1^{EP} = 10$, $P_2^{EP} = 50$, $P_1^{LT} = 1$, $P_2^{LT} = 5$, and $P_1^{PA} = P_2^{PA} = 0.2$. The optimal policy for the demand fulfillments and proactive lateral transshipments is given in Figure 4.8.

4.6 Conclusion

In this chapter, we proved that the structure of the optimal lateral transshipment policy is a threshold type policy, and we gave sufficient conditions under which a (zero) hold back policy or a complete pooling policy is optimal. We studied a number of model extensions fitting within the same framework.

Interesting problems for further research would be the extension to three or more stockpoints, variations in the repair time distribution (such as Erlang$_k$ distributed repair times, or state-dependent repair rates), and the incorporation of so-called pro-active lateral transshipments, i.e. rebalancing of the stock levels triggered by a replenishment.

4.A Appendix: Proofs

4.A.1 Proof of Lemma 4.4.1

Proof. a) We give the proofs for the operator G_1. By interchanging the numbering of the locations, the results directly follow for the operator G_2 as well.

(i) It is straightforward to check that if f is Decr(1) (cf. (4.4.1)), then $G_1 f$ is Decr(1) as well, i.e. if $f(x_1, x_2) \geq f(x_1+1, x_2)$, then $G_1 f(x_1, x_2) \geq G_1 f(x_1+1, x_2)$, for all (x_1, x_2) such that the states appearing exists $\in \mathcal{S}$. Along the same lines it follows that if f is Decr(2) (cf. (4.4.2)), then $G_1 f$ is Decr(2) as well, i.e. then $G_1 f(x_1, x_2) \geq G_1 f(x_1, x_2+1)$. Combining this proves that the operator G_1 preserves Decr.

(ii) Assume that f is Conv(1) (cf. (4.4.3)), then we show that $G_1 f$ is Conv(1) as well.

For $x_1 + 2 < S_1$:

$$\begin{aligned}
&G_1 f(x_1, x_2) + G_1 f(x_1+2, x_2) \\
&= (S_1 - x_1) f(x_1+1, x_2) + x_1 f(x_1, x_2) \\
&\quad + (S_1 - x_1 - 2) f(x_1+3, x_2) + (x_1+2) f(x_1+2, x_2) \\
&= (S_1 - x_1 - 2)\Big[f(x_1+1, x_2) + f(x_1+3, x_2)\Big] + x_1\Big[f(x_1, x_2) + f(x_1+2, x_2)\Big] \\
&\quad + 2f(x_1+1, x_2) + 2f(x_1+2, x_2) \\
&\geq 2(S_1 - x_1 - 2) f(x_1+2, x_2) + 2x_1 f(x_1+1, x_2) + 2f(x_1+1, x_2) + 2f(x_1+2, x_2) \\
&= 2(S_1 - x_1 - 1) f(x_1+2, x_2) + 2(x_1+1) f(x_1+1, x_2) \\
&= 2G_1 f(x_1+1, x_2),
\end{aligned}$$

where the inequality holds by applying that f is Conv(1) on the parts between brackets. And for $x_1 + 2 = S_1$:

$$\begin{aligned}
&G_1 f(S_1 - 2, x_2) + G_1 f(S_1, x_2) \\
&= 2f(S_1 - 1, x_2) + (S_1 - 2) f(S_1 - 2, x_2) + S_1 f(S_1, x_2) \\
&= 2f(S_1 - 1, x_2) + (S_1 - 2)[f(S_1 - 2, x_2) + f(S_1, x_2)] + 2f(S_1, x_2) \\
&\geq 2f(S_1 - 1, x_2) + 2(S_1 - 2) f(S_1 - 1, x_2) + 2f(S_1, x_2) \\
&= 2f(S_1, x_2) + 2(S_1 - 1) f(S_1 - 1, x_2) \\
&= 2G_1 f(S_1 - 1, x_2),
\end{aligned}$$

where again the inequality holds by applying that f is Conv(1) on the part between brackets.

It is straightforward to check that if f is Conv(2) (cf. (4.4.4)), then $G_1 f$ is Conv(2) as well, i.e. then $G_1 f(x_1, x_2) + G_1 f(x_1, x_2 + 2) \geq 2G_1 f(x_1, x_2 + 1)$. Combining this proves that the operator G_1 preserves Conv.

(iii) Along the same lines of the proof of (ii) one can prove that if f is Supermod (cf. (4.4.5)), then $G_1 f$ is Supermod as well. Hence the operator G_1 preserves Supermod.
b) (i)–(iii) trivially follow from part a).

(iv) We show that $G_1 + G_2$ preserves SuperC$(1,2)$; then SuperC$(2,1)$ follows by interchanging the numbering of the locations. Assume that f is SuperC$(1,2)$ (cf. (4.4.6)), then, for $x_1 + 2 < S_1$ and $x_2 + 1 < S_2$:

$$\begin{aligned}
&(G_1 + G_2) f(x_1, x_2+1) + (G_1 + G_2) f(x_1+2, x_2) \\
&= (S_1 - x_1 - 2)\Big[f(x_1+1, x_2+1) + f(x_1+3, x_2)\Big] + 2f(x_1+1, x_2+1) \\
&\quad + x_1\Big[f(x_1, x_2+1) + f(x_1+2, x_2)\Big] + 2f(x_1+2, x_2) \\
&\quad + (S_2 - x_2 - 1)\Big[f(x_1, x_2+2) + f(x_1+2, x_2+1)\Big] + f(x_1+2, x_2+1) \\
&\quad + (x_2+1)\Big[f(x_1, x_2+1) + f(x_1+2, x_2)\Big] - f(x_1+2, x_2).
\end{aligned}$$

Now we use that f is SuperC$(1,2)$, and apply this to the terms between brackets. This gives

$$
\begin{aligned}
&(G_1+G_2)f(x_1,x_2+1)+(G_1+G_2)f(x_1+2,x_2)\\
&\geq (S_1-x_1-2)\Big[f(x_1+2,x_2)+f(x_1+2,x_2+1)\Big]+2f(x_1+1,x_2+1)\\
&\quad +x_1\Big[f(x_1+1,x_2)+f(x_1+1,x_2+1)\Big]+2f(x_1+2,x_2)\\
&\quad +(S_2-x_2-1)\Big[f(x_1+1,x_2+1)+f(x_1+1,x_2+2)\Big]+f(x_1+2,x_2+1)\\
&\quad +(x_2+1)\Big[f(x_1+1,x_2)+f(x_1+1,x_2+1)\Big]-f(x_1+2,x_2)\\
&=(S_1-x_1-1)f(x_1+2,x_2)+(x_1+1)f(x_1+1,x_2)\\
&\quad +(S_1-x_1-1)f(x_1+2,x_2+1)+(x_1+1)f(x_1+1,x_2+1)\\
&\quad +(S_2-x_2)f(x_1+1,x_2+1)+x_2f(x_1+1,x_2)\\
&\quad +(S_2-x_2-1)f(x_1+1,x_2+2)+(x_2+1)f(x_1+1,x_2+1)\\
&=(G_1+G_2)f(x_1+1,x_2)+(G_1+G_2)f(x_1+1,x_2+1).
\end{aligned}
$$

The cases $x_1+2=S_1$ and/or $x_2+1=S_2$ are along the same lines.

(v) As MM $=$ Supermod $\cap$ SuperC (cf. (4.4.8)), it directly follows from parts (iii) and (iv) that G_1+G_2 preserves MM. □

4.A.2 Proof of Lemma 4.4.2

Proof. (i) It is straightforward to check that if f is Decr(j), then $H_i f$ is Decr(j), for $i,j=1,2$.

(ii) In order to prove that H_i preserves MM, we prove (cf. (4.4.8)) that it preserves Supermod, SuperC(1, 2) and SuperC(2, 1) (cf. (4.4.5)–(4.4.7)) together, that is, given that f is Supermod, SuperC(1, 2) and SuperC(2, 1), we show that $H_i f$ is Supermod, SuperC(1, 2) and SuperC(2, 1) as well. We show this for H_1; then for H_2 it follows by interchanging the numbering of the locations. Recall that Supermod and SuperC(i,j) imply Conv(i) (cf. (4.4.3) and (4.4.4)).

The proofs come down to case checking: applying H_1 to $f(x)$ introduces a minimization over three terms, so the sum of two gives a total of $3\times 3=9$ possibilities, which we all check separately. For this we use the trivial result:

$$a\geq \min\{a,b\},\ \forall a,b\in\mathbb{R}.$$

The proofs are given for $x_1>0, x_2>0$, but it is straightforward to check that they also hold for the cases $x_1=0, x_2>0$, and $x_1>0, x_2=0$, and $x_1=0, x_2=0$.

Assume that f is Supermod, SuperC(1, 2) and SuperC(2, 1), which implies that f is also Conv(1) and Conv(2). Below we prove that H_1 preserves (i) Supermod, (ii) SuperC(1, 2), and (iii) SuperC(2, 1).

(i) Supermod

For $x_1>0, x_2>0$:

$$
\begin{aligned}
&H_1f(x_1,x_2)+H_1f(x_1+1,x_2+1)\\
&=\min\Big\{f(x_1-1,x_2),\ f(x_1,x_2-1)+P_{LT_1},\ f(x_1,x_2)+P_{EP_1}\Big\}
\end{aligned}
$$

$$
\begin{aligned}
&+\min\Big\{f(x_1,x_2+1),\ f(x_1+1,x_2)+P_{LT_1},\ f(x_1+1,x_2+1)+P_{EP_1}\Big\}\\
=\min\Big\{&f(x_1-1,x_2)+f(x_1,x_2+1),\ f(x_1-1,x_2)+f(x_1+1,x_2)+P_{LT_1},\\
&f(x_1-1,x_2)+f(x_1+1,x_2+1)+P_{EP_1},\ f(x_1,x_2-1)+P_{LT_1}+f(x_1,x_2+1),\\
&f(x_1,x_2-1)+P_{LT_1}+f(x_1+1,x_2)+P_{LT_1},\\
&f(x_1,x_2-1)+P_{LT_1}+f(x_1+1,x_2+1)+P_{EP_1},\\
&f(x_1,x_2)+P_{EP_1}+f(x_1,x_2+1),\ f(x_1,x_2)+P_{EP_1}+f(x_1+1,x_2)+P_{LT_1},\\
&f(x_1,x_2)+P_{EP_1}+f(x_1+1,x_2+1)+P_{EP_1}\Big\}.
\end{aligned}
$$

It holds that:

$$
\begin{aligned}
&f(x_1-1,x_2)+f(x_1,x_2+1)\geq f(x_1,x_2)+f(x_1-1,x_2+1)\text{ (by (4.4.5))},\\
&f(x_1-1,x_2)+f(x_1+1,x_2)+P_{LT_1}\geq 2f(x_1,x_2)+P_{LT_1}\text{ (by (4.4.3))},\\
&f(x_1-1,x_2)+f(x_1+1,x_2+1)+P_{EP_1}\\
&\quad\geq 2f(x_1,x_2)-f(x_1+1,x_2)+f(x_1+1,x_2+1)+P_{EP_1}\\
&\quad\geq f(x_1,x_2)+f(x_1,x_2+1)+P_{EP_1}\text{ (by (4.4.3), resp. (4.4.5))},\\
&f(x_1,x_2-1)+P_{LT_1}+f(x_1,x_2+1)\geq 2f(x_1,x_2)+P_{LT_1}\text{ (by (4.4.4))},\\
&f(x_1,x_2-1)+P_{LT_1}+f(x_1+1,x_2)+P_{LT_1}\\
&\quad\geq f(x_1,x_2)+P_{LT_1}+f(x_1+1,x_2-1)+P_{LT_1}\text{ (by (4.4.5))},\\
&f(x_1,x_2-1)+P_{LT_1}+f(x_1+1,x_2+1)+P_{EP_1}\\
&\quad\geq 2f(x_1,x_2)-f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2+1)+P_{EP_1}\\
&\quad\geq f(x_1,x_2)+P_{LT_1}+f(x_1+1,x_2)+P_{EP_1}\text{ (by (4.4.4), resp. (4.4.5))},\\
&f(x_1,x_2)+P_{EP_1}+f(x_1+1,x_2+1)+P_{EP_1}\\
&\quad\geq f(x_1+1,x_2)+P_{EP_1}+f(x_1,x_2+1)+P_{EP_1}\text{ (by (4.4.5))}.
\end{aligned}
$$

This implies that:

$$
\begin{aligned}
&H_1f(x_1,x_2)+H_1f(x_1+1,x_2+1)\\
&\geq\min\Big\{f(x_1,x_2)+f(x_1-1,x_2+1),\ 2f(x_1,x_2)+P_{LT_1},\\
&\qquad f(x_1,x_2)+f(x_1,x_2+1)+P_{EP_1},\ f(x_1,x_2)+f(x_1+1,x_2-1)+2P_{LT_1},\\
&\qquad f(x_1,x_2)+P_{LT_1}+f(x_1+1,x_2)+P_{EP_1},\ f(x_1+1,x_2)+f(x_1,x_2+1)+2P_{EP_1}\Big\}\\
&\geq\min\Big\{f(x_1,x_2),\ f(x_1+1,x_2-1)+P_{LT_1},\ f(x_1+1,x_2)+P_{EP_1}\Big\}\\
&\quad+\min\Big\{f(x_1-1,x_2+1),\ f(x_1,x_2)+P_{LT_1},\ f(x_1,x_2+1)+P_{EP_1}\Big\}\\
&=H_1f(x_1+1,x_2)+H_1f(x_1,x_2+1).
\end{aligned}
$$

(ii) SuperC(1,2)
For $x_1>0, x_2>0$:

$$H_1f(x_1+2,x_2)+H_1f(x_1,x_2+1)$$

$$
\begin{aligned}
&= \min\Big\{ f(x_1+1,x_2),\ f(x_1+2,x_2-1)+P_{LT_1},\ f(x_1+2,x_2)+P_{EP_1} \Big\} \\
&\quad + \min\Big\{ f(x_1-1,x_2+1),\ f(x_1,x_2)+P_{LT_1},\ f(x_1,x_2+1)+P_{EP_1} \Big\} \\
&= \min\Big\{ f(x_1+1,x_2)+f(x_1-1,x_2+1),\ f(x_1+1,x_2)+f(x_1,x_2)+P_{LT_1}, \\
&\qquad f(x_1+1,x_2)+f(x_1,x_2+1)+P_{EP_1},\ f(x_1+2,x_2-1)+P_{LT_1}+f(x_1-1,x_2+1), \\
&\qquad f(x_1+2,x_2-1)+P_{LT_1}+f(x_1,x_2)+P_{LT_1}, \\
&\qquad f(x_1+2,x_2-1)+P_{LT_1}+f(x_1,x_2+1)+P_{EP_1}, \\
&\qquad f(x_1+2,x_2)+P_{EP_1}+f(x_1-1,x_2+1),\ f(x_1+2,x_2)+P_{EP_1}+f(x_1,x_2)+P_{LT_1}, \\
&\qquad f(x_1+2,x_2)+P_{EP_1}+f(x_1,x_2+1)+P_{EP_1} \Big\}
\end{aligned}
$$

It holds that:

$$
\begin{aligned}
& f(x_1+1,x_2)+f(x_1-1,x_2+1) \geq f(x_1,x_2)+f(x_1,x_2+1) \text{ (by (4.4.6))}, \\
& f(x_1+2,x_2-1)+P_{LT_1}+f(x_1-1,x_2+1) \\
&\quad \geq f(x_1+1,x_2-1)+f(x_1+1,x_2)-f(x_1,x_2)+P_{LT_1}+f(x_1-1,x_2+1) \\
&\quad \geq f(x_1+1,x_2-1)+P_{LT_1}+f(x_1,x_2+1) \text{ (by twice (4.4.6))}, \\
& f(x_1+2,x_2-1)+P_{LT_1}+f(x_1,x_2)+P_{LT_1} \\
&\quad \geq f(x_1+1,x_2-1)+P_{LT_1}+f(x_1+1,x_2)+P_{LT_1} \text{ (by (4.4.6))}, \\
& f(x_1+2,x_2-1)+P_{LT_1}+f(x_1,x_2+1)+P_{EP_1} \\
&\quad \geq f(x_1+1,x_2-1)+f(x_1+1,x_2)-f(x_1,x_2)+P_{LT_1}+f(x_1,x_2+1)+P_{EP_1} \\
&\quad \geq 2f(x_1+1,x_2)+P_{LT_1}+P_{EP_1} \text{ (by (4.4.6), resp. (4.4.7))}, \\
& f(x_1+2,x_2)+P_{EP_1}+f(x_1-1,x_2+1) \\
&\quad \geq f(x_1+1,x_2)+f(x_1+1,x_2+1)-f(x_1,x_2+1)+P_{EP_1}+f(x_1-1,x_2+1) \\
&\quad \geq f(x_1+1,x_2)+f(x_1,x_2+1)+P_{EP_1} \text{ (by (4.4.6), resp. (4.4.3))}, \\
& f(x_1+2,x_2)+P_{EP_1}+f(x_1,x_2)+P_{LT_1} \geq 2f(x_1+1,x_2)+P_{EP_1}+P_{LT_1} \text{ (by (4.4.3))}, \\
& f(x_1+2,x_2)+P_{EP_1}+f(x_1,x_2+1)+P_{EP_1} \\
&\quad \geq f(x_1+1,x_2)+P_{EP_1}+f(x_1+1,x_2+1)+P_{EP_1} \text{ (by (4.4.6))}.
\end{aligned}
$$

This implies that:

$$
\begin{aligned}
& H_1 f(x_1+2,x_2)+H_1 f(x_1,x_2+1) \\
& \geq \min\Big\{ f(x_1,x_2)+f(x_1,x_2+1),\ f(x_1+1,x_2)+f(x_1,x_2)+P_{LT_1}, \\
&\qquad f(x_1+1,x_2-1)+f(x_1,x_2+1)+P_{LT_1},\ f(x_1+1,x_2-1)+f(x_1+1,x_2)+2P_{LT_1}, \\
&\qquad f(x_1+1,x_2)+f(x_1,x_2+1)+P_{EP_1},\ 2f(x_1+1,x_2)+P_{LT_1}+P_{EP_1}, \\
&\qquad f(x_1+1,x_2)+f(x_1+1,x_2+1)+2P_{EP_1} \Big\} \\
& \geq \min\Big\{ f(x_1,x_2),\ f(x_1+1,x_2-1)+P_{LT_1},\ f(x_1+1,x_2)+P_{EP_1} \Big\} \\
&\quad + \min\Big\{ f(x_1,x_2+1),\ f(x_1+1,x_2)+P_{LT_1},\ f(x_1+1,x_2+1)+P_{EP_1} \Big\}
\end{aligned}
$$

$$= H_1 f(x_1+1,x_2) + H_1 f(x_1+1,x_2+1).$$

(iii) SuperC(2,1)
For $x_1 > 0, x_2 > 0$:

$$\begin{aligned}
&H_1 f(x_1,x_2+2) + H_1 f(x_1+1,x_2)\\
&= \min\Big\{f(x_1-1,x_2+2),\ f(x_1,x_2+1)+P_{LT_1},\ f(x_1,x_2+2)+P_{EP_1}\Big\}\\
&\quad + \min\Big\{f(x_1,x_2),\ f(x_1+1,x_2-1)+P_{LT_1},\ f(x_1+1,x_2)+P_{EP_1}\Big\}\\
&= \min\Big\{f(x_1-1,x_2+2)+f(x_1,x_2),\ f(x_1-1,x_2+2)+f(x_1+1,x_2-1)+P_{LT_1},\\
&\qquad f(x_1-1,x_2+2)+f(x_1+1,x_2)+P_{EP_1},\ f(x_1,x_2+1)+P_{LT_1}+f(x_1,x_2),\\
&\qquad f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2-1)+P_{LT_1},\\
&\qquad f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2)+P_{EP_1},\\
&\qquad f(x_1,x_2+2)+P_{EP_1}+f(x_1,x_2),\ f(x_1,x_2+2)+P_{EP_1}+f(x_1+1,x_2-1)+P_{LT_1},\\
&\qquad f(x_1,x_2+2)+P_{EP_1}+f(x_1+1,x_2)+P_{EP_1}\Big\}.
\end{aligned}$$

It holds that:

$$\begin{aligned}
&f(x_1-1,x_2+2)+f(x_1,x_2) \geq f(x_1-1,x_2+1)+f(x_1,x_2+1) \text{ (by (4.4.7))},\\
&f(x_1-1,x_2+2)+f(x_1+1,x_2-1)+P_{LT_1}\\
&\quad \geq f(x_1-1,x_2+1)+f(x_1,x_2+1)-f(x_1,x_2)+f(x_1+1,x_2-1)+P_{LT_1}\\
&\quad \geq f(x_1-1,x_2+1)+f(x_1+1,x_2)+P_{LT_1} \text{ (by twice (4.4.7))},\\
&f(x_1-1,x_2+2)+f(x_1+1,x_2)+P_{EP_1}\\
&\quad \geq f(x_1-1,x_2+1)+f(x_1,x_2+1)-f(x_1,x_2)+f(x_1+1,x_2)+P_{EP_1}\\
&\quad \geq 2f(x_1,x_2+1)+P_{EP_1} \text{ (by (4.4.7), resp. (4.4.6))},\\
&f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2-1)+P_{LT_1} \geq f(x_1,x_2)+f(x_1+1,x_2)+2P_{LT_1} \text{ (by (4.4.7))},\\
&f(x_1,x_2+2)+P_{EP_1}+f(x_1,x_2) \geq 2f(x_1,x_2+1)+P_{EP_1} \text{ (by (4.4.4))},\\
&f(x_1,x_2+2)+P_{EP_1}+f(x_1+1,x_2-1)+P_{LT_1}\\
&\quad \geq f(x_1,x_2+1)+f(x_1+1,x_2+1)-f(x_1+1,x_2)+P_{EP_1}+f(x_1+1,x_2-1)+P_{LT_1}\\
&\quad \geq f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2)+P_{EP_1} \text{ (by (4.4.7), resp. (4.4.4))},\\
&f(x_1,x_2+2)+P_{EP_1}+f(x_1+1,x_2)+P_{EP_1}\\
&\quad \geq f(x_1,x_2+1)+f(x_1+1,x_2+1)+2P_{EP_1} \text{ (by (4.4.7))}.
\end{aligned}$$

This implies that:

$$\begin{aligned}
&H_1 f(x_1,x_2+2) + H_1 f(x_1+1,x_2)\\
&\geq \min\Big\{f(x_1-1,x_2+1)+f(x_1,x_2+1),\ f(x_1-1,x_2+1)+f(x_1+1,x_2)+P_{LT_1},\\
&\qquad f(x_1,x_2+1)+P_{LT_1}+f(x_1,x_2),\ f(x_1,x_2)+f(x_1+1,x_2)+2P_{LT_1},
\end{aligned}$$

$$
\begin{aligned}
&\quad 2f(x_1,x_2+1)+P_{EP_1},\ f(x_1,x_2+1)+P_{LT_1}+f(x_1+1,x_2)+P_{EP_1},\\
&\quad f(x_1,x_2+1)+f(x_1+1,x_2+1)+2P_{EP_1}\Big\}\\
&\geq \min\Big\{f(x_1-1,x_2+1),\ f(x_1,x_2)+P_{LT_1},\ f(x_1,x_2+1)+P_{EP_1}\Big\}\\
&\quad+\min\Big\{f(x_1,x_2+1),\ f(x_1+1,x_2)+P_{LT_1},\ f(x_1+1,x_2+1)+P_{EP_1}\Big\}\\
&= H_1f(x_1,x_2+1)+H_1f(x_1+1,x_2+1).
\end{aligned}
$$

$\square$

4.A.3 Proof of Theorem 4.4.4

Proof. Consider a demand at stockpoint 1. For $(x_1,x_2)\in\mathcal{S}$ and $u\in\{0,1,2\}$, define

$$
w(u,x_1,x_2):=\begin{cases} V_n(x_1-1,x_2) & \text{if } u=0,\\ V_n(x_1,x_2-1)+P_{LT_1} & \text{if } u=1,\\ V_n(x_1,x_2)+P_{EP_1} & \text{if } u=2,\end{cases} \tag{4.A.1}
$$

where $V_n(x_1,x_2):=\infty$ if $(x_1,x_2)\notin\mathcal{S}$. Hence $H_1V_n(x_1,x_2)=\min_{u\in\{0,1,2\}}w(u,x_1,x_2)$. Define, for $u\in\{0,1,2\}$ and $x_1\in\{0,1,\ldots S_1-1\}, x_2\in\{0,1,\ldots S_2\}$:

$$
\Delta w_{x_1}(u,x_1,x_2):=w(u,x_1+1,x_2)-w(u,x_1,x_2).
$$

Then for each $n\geq 0$, and for $x_2>0$:

$$
\begin{aligned}
&\Delta w_{x_1}(1,x_1,x_2)-\Delta w_{x_1}(0,x_1,x_2)\\
&=V_n(x_1+1,x_2-1)-V_n(x_1,x_2-1)-V_n(x_1,x_2)+V_n(x_1-1,x_2)\geq 0
\end{aligned}
$$

(as, by Theorem 4.4.3, V_n is SuperC$(1,2)$), and:

$$
\begin{aligned}
&\Delta w_{x_1}(2,x_1,x_2)-\Delta w_{x_1}(1,x_1,x_2)\\
&=V_n(x_1+1,x_2)-V_n(x_1,x_2)-V_n(x_1+1,x_2-1)+V_n(x_1,x_2-1)\geq 0
\end{aligned}
$$

(as V_n is Supermod). So, for $x_2>0$, $\Delta w_{x_1}(u,x_1,x_2)$ is increasing in u:

$$
\Delta w_{x_1}(2,x_1,x_2)\geq \Delta w_{x_1}(1,x_1,x_2)\geq \Delta w_{x_1}(0,x_1,x_2).
$$

This implies that, for every $n\geq 0$, there exists a threshold for the inventory level x_1, which can depend on x_2, say $T^{di}_{n,1}(x_2)$, from which on it is optimal to fulfill demands directly from stock. Next there exists a threshold, say $T^{lt}_{n,1}(x_2)$, such that $T^{lt}_{n,1}(x_2)\leq T^{di}_{n,1}(x_2)$, from which on (until $T^{di}_{n,1}(x_2)-1$) it is optimal to fulfill demands via a lateral transshipment, and on the interval $x_1=0$ up till $T^{lt}_{n,1}(x_2)-1$ an emergency procedure is optimal. Hence, if f_{n+1} is the minimizing policy in (4.3.1), then f_{n+1} is a threshold policy. Note that the transition probability matrix of every stationary policy is unichain (since every state can access (S_1,S_2)) and aperiodic (since the transition probability from state (S_1,S_2) to itself is positive). Then, by Puterman [155, Theorem 8.5.4], the long run average costs under the stationary policy f_{n+1} converges to the minimal long run average costs as n tends to infinity. Since there are only finitely many stationary threshold policies, this implies that there exists an optimal stationary policy that is a threshold type policy.

For $x_2 = 0$, lateral transshipments ($u = 1$) are not possible, and we have, for each $n \geq 0$:

$$\begin{aligned}&\Delta w_{x_1}(2, x_1, x_2) - \Delta w_{x_1}(0, x_1, x_2)\\&= V_n(x_1 + 1, x_2) - V_n(x_1, x_2) - V_n(x_1, x_2) + V_n(x_1 - 1, x_2) \geq 0\end{aligned}$$

(as V_n is Conv(1)). Hence $\Delta w_{x_1}(2, x_1, 0) \geq \Delta w_{x_1}(0, x_1, 0)$, and so, for the special case $x_2 = 0$, there exists only one threshold. By the analogous reasoning as for $x_2 > 0$, it follows that there exists a $T_1^{di}(0)$ (which is equal to $T_1^{lt}(0)$). As it is only possible to deliver directly from stock if $x_1 \geq 1$, it follows that $T_1^{di}(0) \geq 1$.

By interchanging the numbering of the stockpoints, the analogous results for stockpoint 2 directly follow. □

4.A.4 Proof of Theorem 4.4.5

PROOF. Analogously to the proof of Theorem 4.4.4, consider a demand at stockpoint 1, and define:

$$\Delta w_{x_2}(u, x_1, x_2) := w(u, x_1, x_2 + 1) - w(u, x_1, x_2),$$

where $w(u, x_1, x_2)$ is as defined in (4.A.1). Then for each $n \geq 0$, and for $x_1 > 0$:

$$\begin{aligned}&\Delta w_{x_2}(0, x_1, x_2) - \Delta w_{x_2}(1, x_1, x_2)\\&= V_n(x_1 - 1, x_2 + 1) - V_n(x_1 - 1, x_2) - V_n(x_1, x_2) + V_n(x_1, x_2 - 1) \geq 0,\end{aligned}$$

(as, by Theorem 4.4.3, V_n is SuperC(2, 1)), and:

$$\begin{aligned}&\Delta w_{x_2}(2, x_1, x_2) - \Delta w_{x_2}(0, x_1, x_2)\\&= V_n(x_1, x_2 + 1) - V_n(x_1, x_2) - V_n(x_1 - 1, x_2 + 1) + V_n(x_1 - 1, x_2) \geq 0,\end{aligned}$$

(as V_n is Supermod). Hence, for $x_1 > 0$:

$$\Delta w_{x_2}(2, x_1, x_2) \geq \Delta w_{x_2}(0, x_1, x_2) \geq \Delta w_{x_2}(1, x_1, x_2).$$

Analogously to the reasoning in the proof of Theorem 4.4.4, it now follows that, for n to infinity, there exist two thresholds $\hat{T}_2^{di}(x_1)$ and $\hat{T}_2^{lt}(x_1)$, where $\hat{T}_2^{di}(x_1) \leq \hat{T}_2^{lt}(x_1)$, such that from $\hat{T}_2^{lt}(x_1)$ lateral transshipments are optimal, from $\hat{T}_2^{di}(x_1)$ to $\hat{T}_2^{lt}(x_1) - 1$ direct delivering from stock is optimal, and from 0 to $\hat{T}_2^{di}(x_1) - 1$ emergency procedures are optimal.

For $x_1 = 0$, directly satisfying a demand from stock ($u = 0$) is not possible, and we have, for each $n \geq 0$:

$$\Delta w_{x_2}(2, x_1, x_2) - \Delta w_{x_2}(1, x_1, x_2) = V_n(x_1, x_2+1) - V_n(x_1, x_2) - V_n(x_1, x_2) + V_n(x_1, x_2-1) \geq 0,$$

(as V_n is Conv(2)). Hence $\Delta w_{x_1}(2, x_1, 0) \geq \Delta w_{x_1}(0, x_1, 0)$, and so, for the special case $x_1 = 0$, there exists only one threshold: $\hat{T}_2^{lt}(0)$ (which is equal to $\hat{T}_2^{di}(0)$). As it is only possible to apply a lateral transshipment if $x_2 \geq 1$, it follows that $\hat{T}_2^{lt}(0) \geq 1$.

By interchanging the numbering of the stockpoints, the analogous results for stockpoint 2 directly follow. □

4.A.5 Proof of Theorem 4.4.6

Proof. We prove part 1a). Part 1b) then directly follows by interchanging the stock-points, and 2) is a trivial consequence of 1a) and 1b).

For 1a), we prove that $a_1^*(1, x_2) = 0$ for all $x_2 \in \{0, 1, \ldots, S_2\}$, then it follows by Theorem 4.4.4 that $T_1^{di}(x_2) = 1$ for all x_2. It suffices to prove that, for all $n \geq 0$:

$$V_n(1, x_2) + P_{EP_1} \geq V_n(0, x_2), \quad \text{for } x_2 \in \{0, \ldots, S_2\}, \tag{4.A.2}$$

$$V_n(1, x_2 - 1) + P_{LT_1} \geq V_n(0, x_2), \quad \text{for } x_2 \in \{1, \ldots, S_2\}. \tag{4.A.3}$$

For $S_1 = 0$ trivially $T_1^{di}(x_2) = 1$ for all x_2, and for $S_1 > 0, S_2 = 0$ we only have to prove (4.A.2).

We prove the inequalities by induction, using that, by Theorem 4.4.3, V_n satisfies (4.4.1)–(4.4.7). For $V_0 \equiv 0$ both inequalities trivially hold. We first prove (i) the induction step of (4.A.2), then (ii) that of (4.A.3), both for $S_1 > 0$. All given inequalities hold by the induction hypothesis, unless stated otherwise.

(i) Assume that (4.A.2) holds for a given n (induction hypothesis), and let $S_1 > 0$. We consider the operators H_1, H_2, G_1 and G_2 separately.

For $x_2 = 0$:

$$\begin{aligned} H_1 V_n(1, 0) + P_{EP_1} &= \min\{P_{EP_1} + V_n(0, 0), 2\, P_{EP_1} + V_n(1, 0)\} \\ &\geq \min\{P_{EP_1} + V_n(0, 0), P_{EP_1} + V_n(0, 0)\} \\ &= P_{EP_1} + V_n(0, 0) = H_1 V_n(0, 0); \end{aligned}$$

and for $x_2 \in \{1, 2, \ldots, S_2\}$:

$$\begin{aligned} H_1 V_n(1, x_2) + P_{EP_1} &= \min\{P_{EP_1} + V_n(0, x_2), P_{EP_1} + P_{LT_1} + V_n(1, x_2 - 1), 2\, P_{EP_1} + V_n(1, x_2)\} \\ &\geq \min\{P_{LT_1} + V_n(0, x_2 - 1), P_{EP_1} + V_n(0, x_2)\} = H_1 V_n(0, x_2). \end{aligned}$$

For $x_2 = 0$:

$$\begin{aligned} H_2 V_n(1, 0) + P_{EP_1} &= \min\{P_{EP_1} + P_{LT_2} + V_n(0, 0), P_{EP_1} + P_{EP_2} + V_n(1, 0)\} \\ &\geq \min\{P_{EP_1} + P_{LT_2} - P_{EP_2} + H_2 V_n(0, 0), H_2 V_n(0, 0)\} \\ &= H_2 V_n(0, 0) + \min\{P_{EP_1} + P_{LT_2} - P_{EP_2}, 0\}; \end{aligned}$$

and for $x_2 \in \{1, 2, \ldots, S_2\}$:

$$\begin{aligned} H_2 V_n(1, x_2) + P_{EP_1} &= \min\{P_{EP_1} + V_n(1, x_2 - 1), P_{EP_1} + P_{LT_2} + V_n(0, x_2), P_{EP_1} + P_{EP_2} + V_n(1, x_2)\} \\ &\geq \min\{V_n(0, x_2 - 1), P_{EP_1} + P_{LT_2} + V_n(0, x_2), P_{EP_2} + V_n(0, x_2)\} \\ &\geq H_2 V_n(0, x_2) + \min\{P_{EP_1} + P_{LT_2} - P_{EP_2}, 0\}, \end{aligned}$$

as $H_2 V_n(0, x_2) = \min\{V_n(0, x_2 - 1), P_{EP_2} + V_n(0, x_2)\}$.

For the operator G_1 we obtain:

$$\begin{aligned} G_1 V_n(1, x_2) + (S_1 - 1) P_{EP_1} &= (S_1 - 1) V_n(2, x_2) + V_n(1, x_2) + (S_1 - 1) P_{EP_1} \\ &= (S_1 - 1)[V_n(2, x_2) - V_n(1, x_2)] + S_1 V_n(1, x_2) + (S_1 - 1) P_{EP_1} \\ &\geq (S_1 - 1)[V_n(1, x_2) - V_n(0, x_2)] + S_1 V_n(1, x_2) + (S_1 - 1) P_{EP_1} \\ &\geq S_1 V_n(0, 1) = G_1 V_n(0, x_2), \end{aligned}$$

where the first inequality holds as V_n is Conv(1) (cf. Theorem 4.4.3).

For $x_2 \in \{0, 1, \ldots, S_2 - 1\}$ we obtain:

$$\begin{aligned} G_2V_n(1,x_2) + S_2P_{EP_1} &= (S_2 - x_2)V_n(1,x_2+1) + x_2V_n(1,x_2) + S_2P_{EP_1} \\ &\geq (S_2 - x_2)V_n(0,x_2+1) + x_2V_n(0,x_2) = G_2V_n(0,x_2); \end{aligned}$$

and for $x_2 = S_2$ trivially:

$$G_2V_n(1,S_2) + S_2P_{EP_1} = S_2V_n(1,S_2) + S_2P_{EP_1} \geq S_2V_n(0,S_2) = G_2V_n(0,S_2).$$

Combining these give, for all x_2 (recall $\nu = \lambda_1 + \lambda_2 + \mu S_1 + \mu S_2$):

$$\begin{aligned} &\nu(V_{n+1}(1,x_2) + P_{EP_1}) \\ &= \lambda_1 H_1V_n(1,x_2) + \lambda_2 H_2V_n(1,x_2) + \mu G_1V_n(1,x_2) + \mu G_2V_n(1,x_2) + \nu\, P_{EP_1} \\ &= \lambda_1[H_1V_n(1,x_2) + P_{EP_1}] + \lambda_2[H_2V_n(1,x_2) + P_{EP_1}] + \mu[G_1V_n(1,x_2) + (S_1-1)P_{EP_1}] \\ &\qquad + \mu[G_2V_n(1,x_2) + S_2P_{EP_1}] + \mu P_{EP_1} \\ &\geq \lambda_1 H_1V_n(0,x_2) + \lambda_2[H_2V_n(0,x_2) + \min\{P_{EP_1} + P_{LT_2} - P_{EP_2}, 0\}] \qquad (4.A.4) \\ &\quad + \mu G_1V_n(0,x_2) + \mu G_2V_n(0,x_2) \\ &\geq \nu V_{n+1}(0,x_2), \qquad (4.A.5) \end{aligned}$$

where the last inequality holds by condition (4.4.9). This completes the induction step, and hence (4.A.2) holds for all $n \geq 0$.

(ii) Assume that (4.A.3) holds for a given n (induction hypothesis), and let $S_1, S_2 > 0$. We consider the operators H_1, H_2 and $G_1 + G_2$ separately:

For $x_2 \in \{2, \ldots, S_2\}$:

$$\begin{aligned} &H_1V_n(1,x_2-1) + P_{LT_1} \\ &= \min\{P_{LT_1} + V_n(0,x_2-1), 2P_{LT_1} + V_n(1,x_2-2), P_{LT_1} + P_{EP_1} + V_n(1,x_2-1)\} \\ &\geq \min\{P_{LT_1} + V_n(0,x_2-1), P_{EP_1} + V_n(0,x_2)\} = H_1V_n(0,x_2); \end{aligned}$$

and for $x_2 = 1$:

$$\begin{aligned} &H_1V_n(1,0) + P_{LT_1} \\ &= \min\{P_{LT_1} + V_n(0,0), P_{LT_1} + P_{EP_1} + V_n(1,0)\} \\ &\geq \min\{P_{LT_1} + V_n(0,0), P_{EP_1} + V_n(0,1)\} = H_1V_n(0,1). \end{aligned}$$

For $x_2 \in \{2, \ldots, S_2\}$:

$$\begin{aligned} &H_2V_n(1,x_2-1) + P_{LT_1} \\ &= \min\{P_{LT_1} + V_n(1,x_2-2), P_{LT_1} + P_{LT_2} + V_n(0,x_2-1), P_{LT_1} + P_{EP_2} + V_n(1,x_2-1)\} \\ &\geq \min\{V_n(0,x_2-1), P_{EP_2} + V_n(0,x_2)\} = H_2V_n(0,x_2); \end{aligned}$$

and for $x_2 = 1$:

$$\begin{aligned} &H_2V_n(1,0) + P_{LT_1} \\ &= \min\{P_{LT_1} + P_{LT_2} + V_n(0,0), P_{LT_1} + P_{EP_2} + V_n(1,0)\} \\ &\geq \min\{V_n(0,0), P_{EP_2} + V_n(0,1)\} = H_2V_n(0,1). \end{aligned}$$

For $x_2 \in \{1, \ldots, S_2 - 1\}$:

$$
\begin{aligned}
&(G_1+G_2)V_n(1,x_2-1)+(S_1+S_2)P_{LT_1}\\
&=(S_1-1)V_n(2,x_2-1)+V_n(1,x_2-1)\\
&\quad+(S_2-x_2+1)V_n(1,x_2)+(x_2-1)V_n(1,x_2-1)+(S_1+S_2)P_{LT_1}\\
&=(S_1-1)[V_n(2,x_2-1)-V_n(1,x_2)]+(S_1-1)V_n(1,x_2)+V_n(1,x_2-1)\\
&\quad+(S_2-x_2)[V_n(1,x_2)-V_n(0,x_2+1)]+V_n(1,x_2)+(S_2-x_2)V_n(0,x_2+1)\\
&\quad+x_2[V_n(1,x_2-1)-V_n(0,x_2)]-V_n(1,x_2-1)+x_2V_n(0,x_2)+(S_1+S_2)P_{LT_1}\\
&\geq(S_1-1)[V_n(1,x_2-1)-V_n(0,x_2)]+S_1V_n(1,x_2)\\
&\quad+(S_2-x_2)[V_n(1,x_2)-V_n(0,x_2+1)]+(S_2-x_2)V_n(0,x_2+1)\\
&\quad+x_2[V_n(1,x_2-1)-V_n(0,x_2)]+x_2V_n(0,x_2)+(S_1+S_2)P_{LT_1}\\
&\geq S_1V_n(1,x_2)+(S_2-x_2)V_n(0,x_2+1)+x_2V_n(0,x_2)+P_{LT_1}\\
&=(G_1+G_2)V_n(0,x_2)+P_{LT_1},
\end{aligned}
$$

where the first inequality holds as V_n is SuperC(1,2) (cf. Theorem 4.4.3). For $x_2 = S_2$:

$$
\begin{aligned}
&(G_1+G_2)V_n(1,S_2-1)+(S_1+S_2)P_{LT_1}\\
&=(S_1-1)V_n(2,S_2-1)+V_n(1,S_2-1)\\
&\quad+V_n(1,S_2)+(S_2-1)V_n(1,S_2-1)+(S_1+S_2)P_{LT_1}\\
&=(S_1-1)[V_n(2,S_2-1)-V_n(1,S_2)]+S_1V_n(1,S_2)\\
&\quad+S_2[V_n(1,S_2-1)-V_n(0,S_2)]+S_2V_n(0,S_2)+(S_1+S_2)P_{LT_1}\\
&\geq(S_1-1)[V_n(1,S_2-1)-V_n(0,S_2)]+S_1V_n(1,S_2)\\
&\quad+S_2[V_n(1,S_2-1)-V_n(0,S_2)]+S_2V_n(0,S_2)+(S_1+S_2)P_{LT_1}\\
&\geq S_1V_n(1,S_2)+S_2V_n(0,S_2)+P_{LT_1}=(G_1+G_2)V_n(0,S_2)+P_{LT_1},
\end{aligned}
$$

where the first inequality again holds as V_n is SuperC(1,2).

Combining these gives, for all $x_2 \in \{1, \ldots, S_2\}$:

$$
\begin{aligned}
&\nu(V_{n+1}(1,x_2-1)+P_{LT_1})\\
&=\lambda_1H_1V_n(1,x_2-1)+\lambda_2H_2V_n(1,x_2-1)+\mu(G_1+G_2)V_n(1,x_2-1)+\nu\,P_{LT_1}\\
&=\lambda_1[H_1V_n(1,x_2-1)+P_{LT_1}]+\lambda_2[H_2V_n(1,x_2-1)+P_{LT_1}]\\
&\qquad+\mu[(G_1+G_2)V_n(1,x_2-1)+(S_1+S_2)P_{LT_1}]\\
&\geq\lambda_1H_1V_n(0,x_2)+\lambda_2H_2V_n(0,x_2)+\mu(G_1+G_2)V_n(0,x_2)=\nu V_{n+1}(0,x_2),
\end{aligned}
$$

which completes the induction step, and hence (4.A.3) holds for all $n \geq 0$. □

4.A.6 Proof of Theorem 4.4.7

PROOF. We again prove only part 1a), as again part 1b) directly follows by interchanging the stockpoints, and 2) is a trivial consequence of 1a) and 1b).

For 1a), analogously to the proof of Theorem 4.4.6, we prove that $a_1^*(0,1) = 1$, then it follows by Theorem 4.4.5 that $\hat{T}_1^{lt}(0) = 1$. By induction, we prove that, for all $n \geq 0$:

$$V_n(0,1) + P_{EP_1} \geq V_n(0,0) + P_{LT_1}. \tag{4.A.6}$$

For $V_0 \equiv 0$ this trivially holds.

Assume that (4.A.6) holds for a given n (induction hypothesis), and we consider the operators H_1, H_2, G_1 and G_2 separately:

$$\begin{aligned} H_1 V_n(0,1) + P_{EP_1} &= \min\{P_{EP_1} + P_{LT_1} + V_n(0,0), 2\,P_{EP_1} + V_n(0,1)\} \\ &\geq P_{EP_1} + P_{LT_1} + V_n(0,0) = H_1 V_n(0,0) + P_{LT_1}; \end{aligned}$$

$$\begin{aligned} H_2 V_n(0,1) + P_{EP_1} &= \min\{P_{EP_1} + V_n(0,0), P_{EP_1} + P_{EP_2} + V_n(0,1)\} \\ &\geq \min\{P_{EP_1} - P_{EP_2} + P_{EP_2} + V_n(0,0), P_{EP_2} + V_n(0,0) + P_{LT_1}\} \\ &= H_2 V_n(0,0) + \min\{P_{EP_1} - P_{EP_2}, P_{LT_1}\}, \end{aligned}$$

as $H_2 V_n(0,0) = P_{EP_2} + V_n(0,0)$;

$$\begin{aligned} G_1 V_n(0,1) + S_1 P_{EP_1} &= S_1[V_n(1,1) - V_n(1,0) + V_n(1,0) + P_{EP_1}] \\ &\geq S_1[V_n(0,1) - V_n(0,0) + V_n(1,0) + P_{EP_1}] \\ &\geq S_1[V_n(1,0) + P_{LT_1}] = G_1 V_n(0,0) + S_1 P_{LT_1}, \end{aligned}$$

where the first inequality holds as V_n is Supermod;

$$\begin{aligned} G_2 V_n(0,1) + (S_2 - 1) P_{EP_1} &= (S_2 - 1)[V_n(0,2) - V_n(0,1) + P_{EP_1}] + S_2 V_n(0,1) \\ &\geq (S_2 - 1)[V_n(0,1) - V_n(0,0) + P_{EP_1}] + S_2 V_n(0,1) \\ &= S_2 V_n(0,1) + (S_2 - 1) P_{LT_1} = G_2 V_n(0,0) + (S_2 - 1) P_{LT_1}, \end{aligned}$$

where the first inequality holds as V_n is Conv(2).

Combining these, using condition (4.4.11), gives, analogously to (4.A.5), the induction step, and hence (4.A.6) holds for all $n \geq 0$. □

4.A.7 Proof of Lemma 4.5.1

PROOF. We give the proofs for the operator $\tilde{G}_1$. By interchanging the numbering of the locations, the results directly follow for the operator $\tilde{G}_2$ as well.

(i) It is straightforward to check that if f is Decr(1) (cf. (4.4.1)), then $\tilde{G}_1 f$ is Decr(1) as well, and if f is Decr(2) (cf. (4.4.2)), then $\tilde{G}_1 f$ is Decr(2) as well. Combining this proves that $\tilde{G}_1$ preserves Decr.

(ii) Assume that f is Conv(1) (cf. (4.4.3)), then we show that $\tilde{G}_1 f$ is Conv(1) as well. For $x_1 + 2 < S_1$ this is straightforward to check, for the case $x_1 + 2 = S_1$ we need Decr(1):

$$\begin{aligned} \tilde{G}_1(f(x_1, x_2) + f(x_1 + 2, x_2)) &= f(x_1 + 1, x_2) + f(x_1 + 2, x_2) \\ &\geq f(x_1 + 2, x_2) + f(x_1 + 2, x_2) = 2\,\tilde{G}_1 f(x_1 + 1, x_2). \end{aligned}$$

The preservation of Conv(2) (cf. (4.4.4)) is again straightforward to check, and hence $\tilde{G}_1$ preserves Conv.

(iii) It is straightforward to check that if f is Supermod (cf. (4.4.5)), then $\tilde{G}_1 f$ is Supermod as well, hence $\tilde{G}_1$ preserves Supermod.

(iv) It is straightforward to check that if f is SuperC(1,2) (cf. (4.4.6)), then $\tilde{G}_1 f$ is SuperC(1,2) as well, hence $\tilde{G}_1$ preserves SuperC(1,2). Assume that f is SuperC(2,1) (cf. (4.4.6)), then we show that $\tilde{G}_1 f$ is SuperC(2,1) as well. For $x_1 + 1 < S_1$ this is straightforward to check, for the case $x_1 + 1 = S_1$ we need Conv(2):

$$\begin{aligned}\tilde{G}_1(f(x_1, x_2+2) + f(x_1+1, x_2)) &= f(x_1+1, x_2+2) + f(x_1+1, x_2)\\ &\geq f(x_1+1, x_2+1) + f(x_1+1, x_2+1)\\ &= \tilde{G}_1(f(x_1, x_2+1) + f(x_1+1, x_2+1)).\end{aligned}$$

(v) By (4.4.8), this is a direct consequence of parts (iii) and (iv). □

4.A.8 Proof of Lemma 4.5.4

PROOF. The proof is cf. Koole [116], as F_i without the condition $x_j < S_j$ is a special case of $T_{CTD(j)}$, defined on [116, p.25]. Hence, by applying [116, Theorem 7.4], the result follows. It remains to check that Supermod, SuperC$(1,2)$ and SuperC$(2,1)$ also hold for the border $x_j = S_j$, which is straightforwardly checked to be the case. □

4.A.9 Proof of Theorem 4.5.5

PROOF. 1) By induction on n, as $V_0^F \equiv 0$ is MM, and using the result of Lemma 4.5.4.
2) Analogously to the proofs of Theorems 4.4.4 and 4.4.5, in particular using that V^F is SuperC$(1,2)$ and SuperC$(2,1)$ (which holds as V^F is MM). □

5

Approximate evaluation of multi-location models with hold back levels

In this chapter, we consider a continuous-time, single-echelon, multi-location inventory model with Poisson demand processes. In case of a stock-out at a local warehouse, a demand can be fulfilled via a lateral transshipment (LT). Each warehouse is assigned a pre-determined sequence of other warehouses where it will request for an LT. However, a warehouse can hold its last part(s) back from such a request. This is called a *hold back pooling policy*, where each warehouse has hold back levels determining whether a request for an LT by another warehouse is satisfied. We are interested in the fractions of demand satisfied from stock (fill rate), via an LT, and via an emergency procedure from an external source. From these, the average costs of a policy can be determined. We present a new approximation algorithm for the evaluation of a given policy, approximating the above mentioned fractions.

Whereas algorithms currently known in the literature approximate the stream of LT requests from a warehouse by a Poisson process, we use an *interrupted Poisson process*. This is a process that is turned alternatingly On and Off for exponentially distributed durations. This leads to the *On/Off overflow algorithm*. In a numerical study we show that this algorithm is significantly more accurate than the algorithm based on Poisson processes, although it requires a longer computation time. Furthermore, we show the benefits of hold back levels, and we illustrate how our algorithm can be used in a heuristic search for the setting of the hold back levels.

5.1 Introduction

Pooling of inventory has proven to be an interesting option for costs reductions and service level improvements. By sharing inventory between local warehouses such pooling benefits can be achieved. In case of a stock-out at one warehouse, a demand can be satisfied via a stock transfer from another warehouse. These stock transfers, which happen within the same echelon, are called *lateral transshipments (LTs)*. One possible strategy for this is *complete pooling*, under which all parts at all local warehouses may be used for an LT. Complete pooling however, is not always optimal. In this chapter we study the so-called *hold back pooling policy*. In Chapter 4, we have proven this policy to be optimal under certain conditions in a two location setting. The policy was introduced by Xu et al. [215] in a periodic review setting, and also arose to be optimal in a related two location problem by Archibald et al. [6]. Under a hold back pooling policy, a warehouse can hold back its last part(s) in stock from an LT request from another warehouse. The *hold back levels* of the warehouses determine how many parts are held back. For determining the optimal hold back levels, evaluation of the costs of a given setting is necessary. These costs can be calculated when one knows the fractions of the demand that are satisfied from stock (fill rate), satisfied via an LT, and satisfied via an emergency procedure. For this, we present a new approximate evaluation algorithm. Such an algorithm facilitates the search for optimal hold back and base stock levels. The distinct feature of our algorithm is that it approximates the LT requests between the warehouses more precisely than current algorithms. These are commonly approximated by Poisson processes (see Axsäter [10], Alfredsson and Verrijdt [2], Kukreja et al. [123], Kutanoglu [124], Kranenburg and Van Houtum [120], and Reijnen et al. [157]), where we use more accurate interrupted Poisson processes. This improves the accuracy of the results, which we show in an extensive numerical study. Moreover, compared to current algorithms given in the literature, our algorithm can deal with hold back levels. To make comparison of results possible, we extend an existing Poisson approximation algorithm for hold back levels.

In the present chapter, we consider an inventory model consisting of N local warehouses, with a given base stock policy. In case of positive on-hand stock at a particular warehouse, an incoming demand at that warehouse is directly satisfied. In case of a stock-out, the demand is satisfied via either an LT or an emergency procedure. A local warehouse is only willing to hand out a part to an LT if it has sufficient inventory, that is, if its on-hand stock level is above a certain threshold, called the *hold back level*. This hold back level can depend on the location the LT request originates from. Furthermore, each warehouse has a prescribed sequence of at most $N-1$ other warehouses that will be contacted for an LT.

Our model is motivated by spare parts inventory systems that serve installed bases of technically advanced machines. As downtimes of these machines are very expensive, spare parts stocks are needed to quickly respond to failures of machines. Further, backorders are not allowed, and thus LTs and emergency (repair) procedures are applied. Typically, demand rates are low and many spare parts are expensive. In such situations, the application of LTs may be very beneficial. Robinson [162] showed that substantial cost savings can be realized by the use of LTs, even when the transportation costs for LTs are high. Based on two case studies in the computer and automobile industry, Cohen and Lee [55] showed that stock pooling is an effective way to improve the service levels with even less on–hand inventory. A case study by Kranenburg and Van Houtum [120] at ASML has shown that using LTs at both the tactical and operational planning level leads to a 30% cost reduction in comparison to using LTs at the operational planning level only. Cohen et al. [53] point out that the pooling of spare parts is one of the best ways for companies to realize cost reductions.

As inventory pooling can simultaneously reduce costs and improve service levels, a lot of research has been devoted to the use of LTs, see Wong et al. [212] and Paterson et al. [153] for overviews. There are many options for the decisions on when to apply

LTs. Generally, we distinguish between *complete pooling* and *partial pooling*. For partial pooling all kinds of restrictions are possible, e.g. LTs can only take place between geographically nearby warehouses [124, 44, 125, 157], the LTs might be executed in only one way [12, 132, 151], not all inventory has to be shared [215, 193]. We take these restrictions into account in the following way. Firstly, each warehouse is assigned a sequence of warehouses it consults for an LT. Reijnen et al. [157] motivate this by a time constraint on the fulfillment of a demand: only warehouses close enough are consulted. Another motivation may be the transport facilities nearby some warehouses. In this way, also transshipments in one-direction only can be taken into account. Next to this, we use a *hold back policy*, cf. [215]. Note that by the combination of these options, we have a very general form of partial pooling. Complete pooling forms a special case, obtained when each warehouse can request for an LT from all other warehouses and all hold back levels are set to zero.

LTs limited by holding back inventory is mainly considered in decentralized inventory models, see e.g. [217, 218]. In such a setting, the local warehouses are independently owned and operated, which gives a game theoretical setting. Another reason for holding back parts is a periodic review setting. Based on the remaining time until a scheduled replenishment, the decision is taken if an LT takes places, as in Archibald et al. [6]. This also occurs in a periodic review setting when the replenishment lead times are non-zero, see Tagaras and Cohen [175]. We, however, concentrate on a continuous review model under centralized control.

Our incentive for the introduction of hold back levels are the results in Chapter 4 (see also Van Wijk et al. [193]) in which it is proven for two local warehouses that the optimal LT policy structure is a hold back policy, under two (sufficient) conditions on the cost parameters. These conditions are typically satisfied when the LT costs are non-negligible, and the emergency procedure costs at both locations are not too asymmetric. The setting assumed is identical to the setting as presented here. The benefit of holding back inventory occurs when a warehouse has only one or a few parts left in stock. When handing e.g. the last part out to an LT request, costs have to be made for this. This warehouse is stocked out until the next replenishment. When it faces a demand during this time, this demand has to be satisfied either via an LT or emergency procedure. In both cases, more costs have to be made than when the first LT request was refused.

From the results of Chapter 4 for two locations, we might expect such a policy to be optimal too, or at least to perform well, for a multi-location setting. Hence, in this work we assume a hold back policy. Although, under the given policy structure, exact evaluation and optimization is theoretically possible via Markov chain analysis, this is infeasible for large instances by the curse of dimensionality, and calculation times explode. Large problem instances require both accurate and fast approximate evaluation algorithms and effective and efficient heuristic optimization procedures (which are built on approximate evaluations). The focus of this chapter is on the development of evaluation algorithms. The development of heuristics requires a study in itself and, in essence, is left outside the scope of this chapter (we do touch this topic in Section 5.5, where we show benefits of hold back levels and we consider the heuristic optimization of hold back levels under given base stock levels).

The presented approximation algorithm is related to the approximate evaluation method as described in Axsäter [10]. He decomposes the network of local warehouses into individual local warehouses. The LTs between them are modeled as overflow demand streams, which are approximated by Poisson processes. In an iterative approach the rates of these Poisson streams and the performance characteristics of the individual local warehouses, are alternately updated. Similar algorithms are used in a.o. [2, 123, 124, 120, 157], each focusing on a different setting.

This chapter extends the above approximate evaluation methods in two directions. First, our model extends all earlier partial pooling models, as we allow for all earlier studied options for partial pooling. Second, our algorithm uses a more accurate approximation of the overflow demand streams. We approximate these streams by a Poisson process that can be turned On and Off, known as an *interrupted Poisson process* (IPP, cf. Kuczura [122]). When a warehouse has parts in stock, the demand overflow to another local warehouse where it requests for LTs, is turned Off. During a stock-out, the demand overflow is turned On and follows a Poisson process. When the consulted warehouse is that low on inventory that it does not fulfill LT requests, the overflow demand stream is routed to a next warehouse, etc. We approximate the durations of these On and Off times by exponential distributions. We call this the *On/Off overflow algorithm*. We refer to the algorithms using an ordinary Poisson overflow process, as the *Poisson overflow algorithm*. By an extensive numerical study, we show that the On/Off overflow algorithm is very accurate, while still being efficient in terms of computations time. We also show that it is significantly more accurate than the Poisson overflow algorithm.

An IPP is a special case of a Markov modulated Poisson process (MMPP, see Fischer and Meier-Hellstern [80] and the reference therein). MMPPs have often been used to describe the arrival process of data and telecommunication traffic, see e.g. [102]. An IPP is a 2-state MMPP where one arrival rate is zero, and arises as a natural approximation for the overflow process of (multi-server) queues, see [122]. E.g., Meier-Hellstern [138] uses IPPs for the approximative analysis of a queueing system consisting of multiple multi-server queues, in which overflows are rerouted to other queues.

The LT problem as studied here is also related to call center models (see Gans et al. [87] for an overview). In these systems, calling customers can be handled by different so-called operators or groups of operators. The inventory in our model is the equivalent of the operators in these call center models. Certain types of calls can only be handled by a subset of operators having appropriate skills. This leads to skill-based-routing of the customers, which gives a form of partial pooling. For structures with many alternative routings, hardly any fast evaluation methods seem to be available in the call center literature. Our type of approximation may be applied to such call centers as well. A complication that has to be incorporated is that in call centers a rerouted customer typically has a slower service rate.

Finally, the idea of hold back levels is related to so-called *critical levels* for stock rationing problems, see Topkis [183]. In these problems, a single warehouse faces multiple classes of customers, with different penalty costs (and hence priorities) for not satisfying a demand. The last part(s) are kept back from lower priority demands, reserving them for the higher priority demand classes. The optimal policy in this case is proven to be a critical level policy, see [143, 183], where the critical levels determine whether a demand is satisfied. Our model can be seen as an extension to a multi-location setting with one class of customers per location.

The outline of this chapter is as follows. We describe the model in more detail and introduce the notation in Section 5.2. In Section 5.3, we describe the *Poisson overflow algorithm* and present our *On/Off overflow algorithm*. In Section 5.4, a numerical study is conducted to test the performance of the On/Off overflow algorithm, using the Poisson overflow algorithm as the benchmark. In Section 5.5, we show the benefits of hold back levels and present a simple heuristic optimization procedure for hold back levels under given base stock levels. Finally, we conclude in Section 5.6. This chapter is based on [195].

5.2 Model and notation

We consider a spare parts inventory system with N local warehouses, numbered $i = 1, 2, \ldots, N$, which provide repairable spare parts for a single critical component of an advanced technical system. This initial stock level at local warehouse i, $i = 1, \ldots, N$, is denoted by $S_i \in \mathbb{N} \cup \{0\}$, which is referred to as the base stock level. The actual on-hand stock is denoted by $x_i \in \{0, 1, \ldots, S_i\}$. Define $\underline{x} = \{x_1, \ldots, x_N\}$. In case of a stock-out, a demand can be fulfilled from another warehouse. In this case, a part is transshipped from a warehouse with positive on-hand stock. This is called a *lateral transshipment* (LT). Each warehouse has a set of hold back levels, determining if LT requests from other warehouses will be accepted. Let $h_{i,j} \in \{0, \ldots, S_i\}$ be the hold back level at warehouse i for accepting an LT request from warehouse $j \neq i$. Only if $x_i > h_{i,j}$, the request is fulfilled. Define $h_{i,i} = 0$ and $\underline{h}_i = (h_{i,1}, \ldots, h_{i,N})$.

When a system breaks down, the failed part has to be replaced by a spare part. The failed part is returned to the stockpoint that supplied the requested spare part, repaired, and is added back to stock after repair. The repair process at local warehouse i is modeled as an ample server queue with exponential service times with mean $1/\mu_i$; i.e., the repair rate equals $(S_i - x_i)\mu_i$. This is equivalent to assuming that repair lead times at warehouse i are exponentially distributed and mutually independent. This assumption facilitates the analysis and is known to be justified, because the performance characteristics of the system as a whole are relatively insensitive to the lead time distributions for repairs, see [2, 157]. Note that only minor changes would have to be made in our analysis to deal with other settings for the repair process, such as a repair rate of μ_i, independently of the number of outstanding orders.

The demand at warehouse i is given by a Poisson process with rate $\lambda_i > 0$, $i = 1, \ldots, N$. We refer to this as class i demand. If there is on-hand stock at the warehouse where a demand arises, the demand is directly fulfilled. Otherwise, the warehouse requests for an LT at other warehouses. For this, each warehouse has a pre-specified order by which it will contact other warehouses. Denote this sequence for warehouse i by σ_i, whose entries are in $\{1, \ldots, N\} \backslash \{i\}$. If e.g. $\sigma_1 = \{2, 3\}$, then warehouse 1 will first consult 2 for an LT, then 3, and otherwise the demand is satisfied by an external source and thus lost for the local warehouses (see Figure 5.1). The latter is referred to as an emergency procedure. By $j \in \sigma_i$ we denote that j is an element of the sequence σ_i, and $\sigma_i(k)$ denotes the kth element of the sequence, $1 \leq k \leq |\sigma_i|$, where $|\sigma_i|$ denotes the number of elements in the sequence. As $i \notin \sigma_i$, we have $0 \leq |\sigma_i| \leq N - 1$. Location $j \neq i$ accepts warehouse i's LT request only if $x_j > h_{j,i}$. In this case, a part is taken from warehouse j's stock and used to fulfill warehouse i's demand. If $x_j \leq h_{j,i}$ for all $j \in \sigma_i$ (or if $\sigma_i = \emptyset$), then the demand is lost for the local warehouses. We do not allow for backorders, neither for rebalancing of stock. We assume all demand and repair processes to be mutually independent.

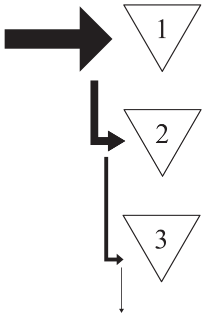


Figure 5.1: Demand overflow from warehouse 1 when $\sigma_1 = (2,3)$.

So, for a demand at warehouse i, there are three possibilities to fulfill it: directly from stock, via an LT, or via an emergency procedure. We are interested in the fractions of the demands that are fulfilled in either way, so we define for warehouse i:

- β_i: fraction of the class i demand that is fulfilled directly from stock;
- $\alpha_{i,j}$: fraction of the class i demand that is fulfilled via an LT from j ($j \neq i$), and by definition $\alpha_{i,j} = 0$ if $j \notin \sigma_i$;
- θ_i: fraction of the class i demand fulfilled via emergency procedure.

Furthermore, we define α_i to be the total fraction of the class i demand at warehouse i that is fulfilled via an LT:

$$\alpha_i = \sum_{j \in \sigma_i} \alpha_{i,j}, \tag{5.2.1}$$

where an empty sum equals zero. By definition, for all i it holds that:

$$\beta_i + \alpha_i + \theta_i = 1. \tag{5.2.2}$$

The model can be evaluated exactly as a Markov process for given hold back and base stock levels. The state of the system is given by the vector of on-hand stock levels $\underline{x} = \{x_1, \ldots, x_N\} \in \mathcal{S}$ where the state space $\mathcal{S}$ is given by $\mathcal{S} = \{0, 1, \ldots, S_1\} \times \ldots \times \{0, 1, \ldots, S_N\}$. For $N = 2$ the transition rates are shown in Figure 5.2. Denote by Q the transition rate matrix and by π the stationary probability distribution of $\underline{x}$. It is well known that π can be found by solving the following system:

$$\begin{cases} \pi Q = \underline{0}, \\ \displaystyle\sum_{\underline{x} \in \mathcal{S}} \pi(\underline{x}) = 1. \end{cases} \tag{5.2.3}$$

From π, the values of β_i, $\alpha_{i,j}$, and θ_i follow.

By the dimension of $\mathcal{S}$, which is $|\mathcal{S}| = \prod_{i=1}^{N} (S_i + 1)$, and hence the dimension of Q, evaluation of the stationary probability distribution by solving (5.2.3) is not feasible for larger values of N and S_i because of the curse of dimensionality. Hence, there is a need for fast and accurate approximations for the β_i's, $\alpha_{i,j}$'s, and θ_i's.

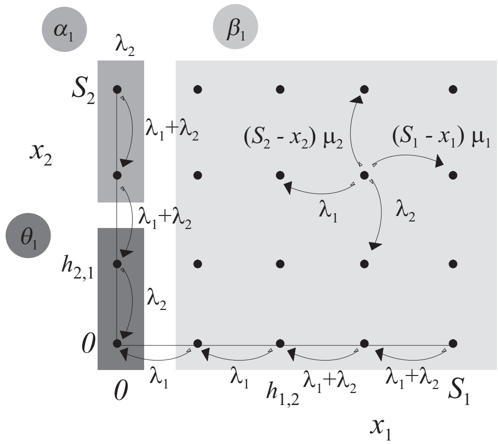


Figure 5.2: Diagram of the transition rates for $N = 2$, when $\sigma_1 = (2)$, $\sigma_2 = (1)$ and the hold back levels are given by $h_{1,2} = 2$ and $h_{2,1} = 1$. Note that when $x_1 = 0$ and $x_2 > h_{2,1} = 1$, demands at warehouse 1 are satisfied by an LT from warehouse 2, and vice versa when $x_2 = 0$ and $x_1 > h_{1,2} = 2$. Indicated are the states that contribute to β_1 (light gray), α_1 (=$\alpha_{1,2}$, gray), and θ_1 (dark gray).

5.3 Approximation algorithms

In this section we present the *Poisson overflow algorithm* and the *On/Off overflow algorithm*. Both approximate the β_i, $\alpha_{i,j}$, and θ_i for all i and j when the hold back and base stock levels are given. We first explain the general idea behind both algorithms. Then, for each one separately, we discussion the steps it consists of in more detail.

5.3.1 Idea behind approximations

Our approximation is based on a *decomposition* of the network of warehouses into individual warehouses, as in Axsäter [10]. In this way, we only deal with solving a Markov process per local warehouse, with states x_i, instead of solving the multi-dimensional Markov process with state $\underline{x}$. The LT requests are modeled as overflow demand streams. These constitute an additional demand stream at other warehouses. The stream from warehouse i at $j \in \sigma_i$ is referred to as the overflow stream (i, j). This is graphically depicted in Figure 5.1. Consequently, each warehouse can be evaluated individually as an Erlang loss system ($\cdot/M/S/S$ queue) with state-dependent arrival rates. In earlier approximations [10, 2, 123, 124, 120, 157], these overflow demand streams have been assumed to be Poisson processes. For the sake of selfcontainedness and in order to show how this approximation can be extended for hold back levels, we first present this algorithm, which we call the *Poisson overflow algorithm*. In our second and main algorithm, the On/Off algorithm, we approximate each overflow stream by an interrupted Poisson process, cf. [122], a Poisson process that is alternatingly turned On and Off.

We expect the approximation using the On/Off processes to be more precise, as it better follows the actual overflow demand streams. Overflow demands occur when the warehouse is stocked out. So, during a stock-out the overflow demand process is turned

On. The demands it is then facing, flow over to other warehouses as LT requests, following a Poisson process. On the other hand, when the warehouse has a positive stock level, there are no overflow demands. So, the overflow process is turned Off. Hence, it is one step more accurate than a Poisson process. The approximation here is that we *assume* the On and Off durations to be independently, exponentially distributed. We take the means of both to be equal to the actual means. It will turn out that this approximation performs very well.

Both algorithms consist of two main steps, which are alternately executed:

Step 1: Evaluation of the steady-state distribution (and hence performance characteristics) of the individual warehouses, given the overflow demand streams;

Step 2: Updating of the overflow demand streams, given the steady-state distribution of the individual warehouses.

The two steps are executed until the changes in consecutive iterations are smaller than some pre-specified, small value ε.

5.3.2 Poisson overflow algorithm

We approximate the overflow demand streams by Poisson processes. Let $\lambda_{i,j}$ be the rate of the overflow demand stream (i,j), $j \in \sigma_i$. Define $\lambda_{i,i} = \lambda_i$ and $\lambda_{i,j} = 0$, $j \notin \sigma_i$, $j \neq i$. The overflow demand rates $\lambda_{i,j}$ for all $j \in \sigma_i$ are calculated from the probabilities that a demand is satisfied either directly from stock or via an LT. Recall that an LT is carried out from warehouse j to warehouse i only if $x_j > h_{j,i}$. We denote the probability that the latter is true by $p_{i,j}$ for all i and $j \in \sigma_i \cup \{i\}$, defining $p_{i,i} = \mathbb{P}[x_i > 0]$. So, $p_{i,i}$ equals β_i, the probability that a demand is directly fulfilled from stock. The $p_{i,j}$ are derived when evaluating an individual warehouse. We initially assume that $\lambda_{i,j} = 0$ for all overflow demand streams (i,j).

Evaluation of individual warehouses

Given the overflow rates $\lambda_{i,j}$ for all i and $j \in \sigma_i$, we determine the probabilities that a demand can be fulfilled, either directly from stock, or via an LT. Each of the warehouses is evaluated individually. For warehouse i, we consider the Markov process, the state of which is given by its stock level x_i on the state space $\{0, 1, \ldots, S_i\}$. We have the following transitions. The repair rate is given by $(S_i - x_i)\mu_i$. Class i demands arise with rate λ_i, and are satisfied when $x_i > 0$. Moreover, the warehouse faces the demand overflow streams from the other warehouses, namely with rate $\lambda_{j,i}$ from warehouse j. Only when $x_i > h_{i,j}$ such a demand is satisfied. Therefore, the demand rate depends on the state of the system.

The arrival rate $\gamma_i(x_i)$ when the on-hand stock is x_i, equals $\gamma_i(x_i) = \sum_{j=1}^{N} \lambda_{j,i} \cdot 1\{x_i > h_{i,j}\}$. Note that $\gamma_i(0) = 0$, and that $\gamma_i(x_i)$ always includes $\lambda_i = \lambda_{i,i}$ for $x_i > 0$. Figure 5.3 shows an example of the transitions rates (where per state both the on-hand stock x_i and the number of outstanding orders $y = S_i - x_i$ is denoted).

Consequently, we can analyze warehouse i separately using the Erlang loss model. The steady-state behavior of the number of outstanding orders is identical to the steady-state behavior of the number of busy servers y in an Erlang loss system ($M/M/S_i/S_i$ queue) with S_i servers, a state-dependent arrival (i.e. demand) rate $\gamma_i(S_i - y)$, and mean

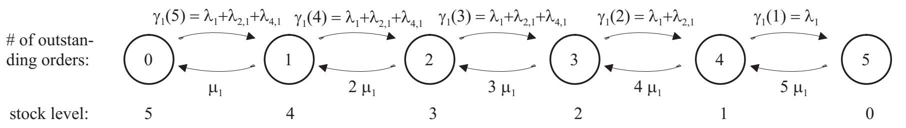


Figure 5.3: Example of the transitions rates at warehouse 1, where $N = 4$, $S_1 = 5$, and $\underline{h}_1 = (0,1,5,2)$.

repair lead time $1/\mu_i$ as mean service time. Define $\tilde{L}_i$ to be the stationary probability distribution of $y \in \{0,1,\dots,S_i\}$ at warehouse i, denoting by $\tilde{L}_i(y)$ its yth element:

$$\tilde{L}_i(y) = \frac{\dfrac{\prod_{j=0}^{y-1} \gamma_i(S_i - j)}{\mu_i^y \, y!}}{\sum_{n=0}^{S_i} \dfrac{\prod_{m=0}^{n-1} \gamma_i(S_i - m)}{\mu_i^n \, n!}}, \quad y = 0,\dots,S_i.$$

Recall that $\gamma_i(\cdot)$ is fully determined by the vectors $\underline{h}_i$ and $(\lambda_{1,i},\dots,\lambda_{N,i})$. From the stationary probability distribution $\tilde{L}_i(\cdot)$ the probabilities $p_{i,j}$ can be computed:

$$p_{i,j} = \mathbb{P}[x_j > h_{j,i}] = \sum_{y=0}^{S_j - h_{j,i} - 1} \tilde{L}_j(y), \text{ for all } i \text{ and } j \in \sigma_i, \tag{5.3.1}$$

and 0 otherwise. Note that $\tilde{L}_i(\cdot)$ depends on the overflow demand rates $\lambda_{i,j}$.

Updating overflow rates

Given the probabilities $p_{i,j}$, for all i, j, we derive the overflow demand rates $\lambda_{i,j}$. The fraction of demand that is directly satisfied from stock (the fill rate) is given by:

$$\beta_i = p_{i,i}. \tag{5.3.2}$$

Hence, the rate of overflow stream $(i, \sigma_i(1))$ is $\lambda_{i,\sigma_i(1)} = (1 - \beta_i)\lambda_i$ (assuming $\sigma_i \neq \emptyset$). Of this stream, a fraction $p_{i,\sigma_i(1)}$ is satisfied by warehouse $\sigma_i(1)$, and hence the remaining overflow to $\sigma_i(2)$ has rate $\lambda_{i,\sigma_i(2)} = (1 - p_{i,\sigma_i(1)})\lambda_{i,\sigma_i(1)}$. In general, defining $p_{i,\sigma_i(0)} = \beta_i$, the rate $\lambda_{i,\sigma_i(k)}$ is recursively given by $\lambda_{i,\sigma_i(k)} = (1 - p_{i,\sigma_i(k-1)})\lambda_{i,\sigma_i(k-1)}$. Hence, for $k = 1,\dots,|\sigma_i|$:

$$\lambda_{i,\sigma_i(k)} = \lambda_i \prod_{l=0}^{k-1} (1 - p_{i,\sigma_i(l)}). \tag{5.3.3}$$

Here we use of the assumption that the stock levels at the local warehouses are independent.

For the fraction of the demands that is satisfied by an LT from warehouse j, we have

$$\alpha_{i,j} = p_{i,j} \frac{\lambda_{i,j}}{\lambda_i}, \text{ for all } i \text{ and } j \in \sigma_i. \tag{5.3.4}$$

Algorithm 1: Poisson overflow algorithm	
	Input: $\lambda_i, \mu_i, S_i, \underline{h}_i, \sigma_i$, for $i = 1, \ldots, N$, ε;
	Output: β_i, $\alpha_{i,j}$ and so α_i, θ_i, for $i, j = 1, \ldots, N$, $i \neq j$;
Step 0:	Initialize for all i, $j \in \sigma_i$: $\lambda_{i,j} = 0$ and $\lambda_{i,i} = \lambda_i$.
Step 1:	Calculate for all i, $j \in \sigma_i \cup \{i\}$: $p_{i,j}$ using (5.3.1).
Step 2:	Calculate for all i, $j \in \sigma_i$: $\lambda_{i,j}$ using (5.3.3).
Step 3:	Repeat Steps 1 and 2 until the $p_{i,j}$'s do not change more than ε, for all i, j. Then return, for all $i, j \in \sigma_i$: β_i using (5.3.2), $\alpha_{i,j}$ using (5.3.4), α_i using (5.2.1), and θ_i using (5.2.2).

Here $\lambda_{i,j}/\lambda_i$ is the fraction of the demand of i that is offered to warehouse j, and of this, a fraction $p_{i,j}$ is satisfied. Next, the total fraction that is satisfied via LT, α_i, is given by (5.2.1). Given β_i and α_i, the fraction of the demand that is not satisfied by the local warehouses, θ_i, follows from (5.2.2). Algorithm 1 summarizes all steps.

5.3.3 On/Off overflow algorithm

We now develop a more precise approximation for the overflow demand processes. Namely, instead of a Poisson process, we use an *interrupted Poisson process*, cf. [122]: a Poisson process which is alternately turned On, for an exponentially distributed time, and then turned Off, for another (independent) exponentially distributed time. This process is described by three parameters. For the overflow demand stream (i, j), let $1/\phi_{j,i}$ be the mean Off duration, $1/\eta_{j,i}$ the mean On duration, and λ_i the rate while On. So, we have to estimate two out of three parameters of the interrupted Poisson process.

The idea is as follows. The overflow demand process of warehouse i at warehouse $\sigma_i(1)$ can be in two states, based on warehouse i's stock level. If warehouse i has on-hand stock, there is *no* overflow. So, the overflow process is turned Off. However, if warehouse i faces a stock-out, all demands flow over to warehouse $\sigma_i(1)$: the overflow process is turned On, and is given by a Poisson process with rate λ_i.

The same reasoning is applicable for the overflow demand of i at warehouse $\sigma_i(k)$ for $k = 2, \ldots, |\sigma_i|$. The overflow process is On at $\sigma_i(2)$ exactly when i is stocked out and warehouse $\sigma_i(1)$'s inventory level is below or at its hold back level for i, i.e., when $x_{\sigma_i(1)} \leq h_{\sigma_i(1),i}$. The overflow is turned Off otherwise.

When precisely overflows are turned On or Off, basically depends on the entire state space. The approximation we apply here, is to approximate the On and Off durations by exponential distributions. We choose the means of these to be equal to the estimated mean durations. These means are updated in each iteration of the algorithm.

The mean On duration at $\sigma_i(1)$ is the duration that the stock level x_i equals zero. It is exactly exponentially distributed, with mean

$$1/\eta_{\sigma_i(1),i} = 1/(S_i \, \mu_i). \tag{5.3.5}$$

In general, for the other On and Off durations, the exponential distribution is an approximation. Like in the Poisson overflow algorithm, we initialize by putting all overflow demand streams to zero. That is, all On/Off processes start being turned Off:

$$\phi_{j,i} = 0 \text{ for } j \in \sigma_i, \quad \eta_{j,i} = \infty \text{ for } j \in \sigma_i. \tag{5.3.6}$$

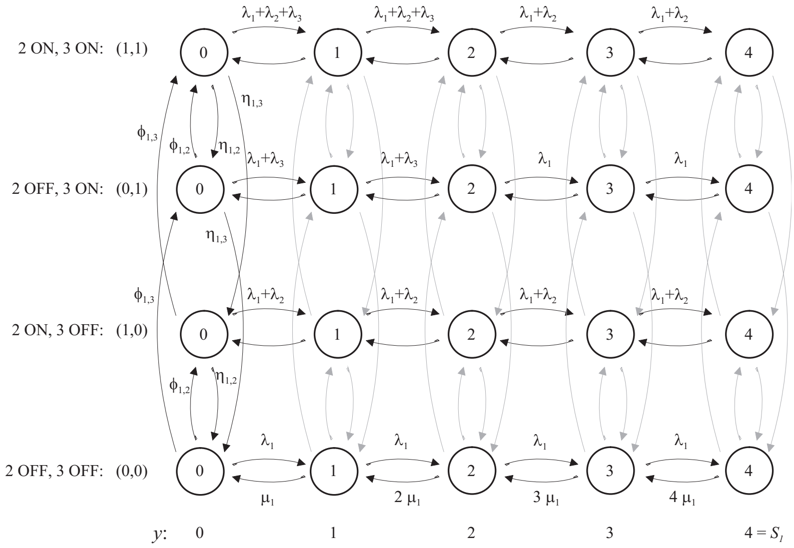


Figure 5.4: Example of the Markov processes of warehouse 1*, where* $N = 3$*,* $S_1 = 4$*,* $h_1 = (0, 0, 2)$*, and* $d_1 = \{2, 3\}$.

Evaluation of individual warehouses

Given the mean On and Off durations, we evaluate each of the individual warehouses. Consider warehouse i. Next to its own demand stream, it faces a number of overflow demands streams, namely from the warehouses j for which $i \in \sigma_j$. Denote this vector by $d_i = (j \mid i \in \sigma_j, j = 1, \ldots, N)$, and by $d_i(k)$ its kth element. As each of these overflow demand streams can either be turned On or Off, we have $2^{|d_i|}$ possible combinations, where $|d_i| \leq N-1$. Hence the joint process of the stock level and this state of the overflow demand streams, is a Markov process on a state space of dimension $2^{|d_i|}$ by $S_i + 1$. We encode the states by $(y, \underline{\delta})$ with $y = S_i - x_i$ the number of outstanding orders, $y \in \{0, 1, \ldots, S_i\}$, and $\underline{\delta} \in \{0, 1\}^{|d_i|}$, where the kth component of $\underline{\delta}$ equals 0 if the overflow stream from $d_i(k)$ is Off, and 1 if On. This state space is denoted by $\mathcal{S}_i$. Figure 5.4 shows an example.

The transition rates of this Markov process are as follows. When the process is in state $(y, \underline{\delta})$ three types of transitions can occur: a replenishment, an (overflow) demand, or a change in whether one of the processes is On or Off. With rate $y\,\mu_i$ replenishments take place, moving the process to state $(y-1, \underline{\delta})$. A demand, occurring with rate λ_i, moves the process to $(y+1, \underline{\delta})$ if $y < S_i$. An overflow demand, occurring at rate λ_k for all $k \in d_i$ when the overflow demand stream (k, i) is On, is only accepted if $S_i - y > h_{i,k}$. This also moves the process to state $(y+1, \underline{\delta})$. Finally, each of the overflow demand processes can switch from On to Off or vice versa. Transitions are only possible between states for which $\underline{\delta}$ differs in exactly one entry. With rate $\phi_{i,d_i(j)}$ the overflow from warehouse $d_i(j)$ is switched On, $j = 1, \ldots, |d_i|$, hence with this rate the process moves to state $(y, \underline{\delta} + \underline{e}_{d_i(j)})$, where $\underline{e}_k$ denotes the unit vector of appropriate length with an 1 at position k. Analogously, with rate $\eta_{i,d_i(j)}$ the process moves to state $(y, \underline{\delta} - \underline{e}_{d_i(j)})$. See again Figure 5.4 for an example.

Denote by Q_i the matrix of transitions rates and denote by $\pi_i(y,\underline{\delta})$ the stationary probability distribution of this process. Then π_i can be found by solving the system:

$$\begin{cases} \pi_i Q_i = \underline{0}, \\ \sum_{(y,\underline{\delta})\in\mathcal{S}_i} \pi_i(y,\underline{\delta}) = 1. \end{cases} \tag{5.3.7}$$

The dimension of Q_i is $(S_i+1)2^{|d_i|}$ by $(S_i+1)2^{|d_i|}$, where $|d_i| \le N-1$. We have to solve the system for all $i = 1,\dots,N$ in each iteration of the algorithm. Note that this is of a much smaller order than the original problem (5.2.3), where the dimension of Q is $\prod_{i=1}^{N}(S_i+1)$ by $\prod_{i=1}^{N}(S_i+1)$. We solve (5.3.7) as a regular system of equation using standard techniques. More tailored methods have been developed for solving steady-state distributions, see e.g. [50] for an efficient algorithm for solving (5.3.7).

Updating On and Off durations

Given the stationary probability distribution of each of the individual warehouses, we update the mean On and Off durations. Consider warehouse i and first concentrate on stream $(i,\sigma_i(1))$. Its mean On duration is given by (5.3.5). We show that, by using the stationary probability distribution π_i, we can directly find the mean Off duration $1/\phi_{\sigma_i(1),i}$.

The fraction of class i's demands that is satisfied from stock is β_i. Hence, by PASTA, β_i is also the fraction of time that class i's overflow is turned Off. And thus the fraction of time the overflow is turned On is $1-\beta_i$. The state space $\mathcal{S}_i$ can be split into two mutually exclusive subsets, say $\mathcal{S}_{i,\text{off}}$ (all states $(y,\underline{\delta}) \in \mathcal{S}_i$ for which $y < S_i$) and $\mathcal{S}_{i,\text{on}}$ (all states $(y,\underline{\delta}) \in \mathcal{S}_i$ for which $y = S_i$). Denote by $\mathbb{E}[\mathcal{S}_{i,\text{off}}]$ the expected duration the process is in subset $\mathcal{S}_{i,\text{off}}$, once first entered it, before leaving it again. Define $\mathbb{E}[\mathcal{S}_{i,\text{on}}]$ analogously. It holds that

$$\frac{\beta_i}{1-\beta_i} = \frac{\mathbb{E}[\mathcal{S}_{i,\text{off}}]}{\mathbb{E}[\mathcal{S}_{i,\text{on}}]}.$$

Using that $\phi_{\sigma_i(1),i} = 1/\mathbb{E}[\mathcal{S}_{i,\text{off}}]$ and $\eta_{\sigma_i(1),i} = 1/\mathbb{E}[\mathcal{S}_{i,\text{on}}] = S_i\,\mu_i$, it follows that:

$$\phi_{\sigma_i(1),i} = S_i\,\mu_i\,\frac{1-\beta_i}{\beta_i}. \tag{5.3.8}$$

The β_i follows from the stationary probability distribution π_i:

$$\beta_i = 1 - \sum_{(S_i,\underline{\delta})\in\mathcal{S}_i} \pi_i(S_i,\underline{\delta}).$$

For the On and Off durations of i at $\sigma_i(k)$ for $k = 2,\dots,|\sigma_i|$ some more work has to be done. Firstly, note that when the overflow stream $(i,\sigma_i(1))$ is turned On, there are periods that these demands are satisfied by warehouse $\sigma_i(1)$, and periods that this is not the case because $x_{\sigma_i(1)} \le h_{\sigma_i(1),i}$. During the latter periods, the overflow $(i,\sigma_i(2))$ is turned On. The duration of this On period, and that of the Off period as well, depends on the (Markov) process at warehouse $\sigma_i(1)$. Hence, in general, the durations of the On and Off periods of i at warehouse $\sigma_i(k)$ follow from evaluation of the Markov process at warehouse $\sigma_i(k-1)$, $k=2,\dots,|\sigma_i|$.

To find the expected On and Off durations of an overflow stream, we have to determine the expected time the Markov process is in a certain subset of states from the moment on the process enters it, before leaving it again. For this, we split the state space $\mathcal{S}_{\sigma_i(k-1)}$ into two mutually independent subsets, one consisting of the states in which the overflow of i to warehouse $\sigma_i(k)$ in turned On, and the other for which it is turned Off. To calculate the mean time spent in a certain subset, we view all states *except* the subset of interest as absorbing states, i.e. states that the Markov process cannot leave anymore once entered. Then we need to derive the mean time until absorption in these states. We first describe, following [91], how the mean time to absorption can be computed in general. Next we explain how this can be used to calculate the mean On and Off durations.

Consider a general, irreducible Markov process with state space $\mathcal{S}' = \mathcal{S}'_1 \cup \mathcal{S}'_2$, where $\mathcal{S}'_1 \cap \mathcal{S}'_2 = \emptyset$, with transition rates q_{ij} for $i, j \in \mathcal{S}'$. Let the matrix P be the transition probability matrix, given by $p_{ii} = (\nu_{\max} - |q_{ii}|)/\nu_{\max}$, and $p_{ij} = q_{ij}/\nu_{\max}$ for $i \neq j$, where $\nu_{\max} = \max_i |q_{ii}|$. Let $P_{\mathcal{S}'_1}$ be an $|\mathcal{S}'_1|$ by $|\mathcal{S}'_1|$ matrix with only the rows and columns of P that correspond to states in $\mathcal{S}'_1$. Its row sums are ≤ 1. Given that we start in a state $s \in \mathcal{S}'_1$, let t_s denote the expected number of steps to get absorbed in $\mathcal{S}'_2$. It is the unique solution of $(I - P_{\mathcal{S}'_1})\underline{t} = (1, 1 \ldots, 1)^T$, where $\underline{t} = (t_1, \ldots, t_{|\mathcal{S}'_1|})^T$ and I is the identity matrix of appropriate size. Then the vector with the mean times until absorption of the Markov process is $\underline{t}/\nu_{\max}$.

When we are given an initial distribution over the starting states in $\mathcal{S}'_1$, say $\underline{p} = (p_1, \ldots, p_{|\mathcal{S}'_1|})$, the mean time until absorption is

$$\mathbb{E}[T(\mathcal{S}'_1)] = \underline{p} \cdot \underline{t}/\nu_{\max}, \tag{5.3.9}$$

where the dot denotes the inner product of two vectors. When we want to find the mean duration the system is in $\mathcal{S}'_1$ before going to $\mathcal{S}'_2$, this initial distribution is given by the steady-state probability distribution that the first step out of $\mathcal{S}'_2$ is to $s \in \mathcal{S}'_1$, say p_s. Denote by $\pi = (\pi_{\mathcal{S}'_1}, \pi_{\mathcal{S}'_2})$ the stationary probability distribution. Let

$$\underline{\tilde{p}} = \pi_{\mathcal{S}'_2} \cdot P_{\mathcal{S}'_2, \mathcal{S}'_1},$$

where $P_{\mathcal{S}'_2, \mathcal{S}'_1}$ denotes the (non-square) matrix, which is the part of the transition probability matrix P of which the rows correspond to states in $\mathcal{S}'_2$, and the columns to states in $\mathcal{S}'_1$. We normalize $\underline{\tilde{p}}$ to find $\underline{p} = \{p_1, \ldots, p_{|\mathcal{S}'_1|}\}$:

$$\underline{p} = \underline{\tilde{p}} / \sum_{s \in \mathcal{S}'_1} \tilde{p}_s.$$

Hence, using (5.3.9) we have $\mathbb{E}[T(\mathcal{S}'_1)]$ and the rate at which the process jumps from subset $\mathcal{S}'_1$ to $\mathcal{S}'_2$ is $1/\mathbb{E}[T(\mathcal{S}'_1)]$.

In order to calculate the mean On and Off durations of the stream $(i, \sigma_i(k))$, we split the state space $\mathcal{S}_{\sigma_i(k-1)}$ into two mutually independent subsets. Let $j = \sigma_i(k-1)$. The overflow of i to $\sigma_i(k)$ is turned On when i's overflow to j is On but is not satisfied there, because the stock level $x_j \leq h_{j,i}$. Denote by $\mathcal{S}_j(i)$ the subset of states of $\mathcal{S}_j$ for which this holds. Recall that $y = S_i - x_i$, then this subset is given by:

$$\mathcal{S}_j(i) = \{(y, \underline{\delta}) \in \mathcal{S}_j \mid \underline{\delta}^{(i)} = 1 \text{ and } y > h_{j,i}\},$$

where $\underline{\delta}^{(i)}$ denotes the component of $\underline{\delta}$ corresponding to the overflow demand stream (i, j). Furthermore, define $\overline{\mathcal{S}}_j(i) = \mathcal{S}_j \backslash \mathcal{S}_j(i)$, that is, the complement of $\mathcal{S}_j(i)$ with respect to $\mathcal{S}_j$.

Algorithm 2: On/Off overflow algorithm	
	Input: $\lambda_i, \mu_i, S_i, \underline{h}_i, \sigma_i$, for $i = 1, \ldots, N$, ε;
	Output: β_i, $\alpha_{i,j}$ and so α_i, and θ_i, for $i, j = 1, \ldots, N$, $i \neq j$;
Step 0:	Initialize for all i, j: $\phi_{j,i}$ and $\eta_{j,i}$ using (5.3.6).
Step 1:	Solve for all warehouses i the stationary probability distribution π_i using (5.3.7).
Step 2:	Calculate for all $i, j \in \sigma_i$: $\eta_{j,i}$ and $\phi_{j,i}$ using (5.3.5), (5.3.8), and (5.3.10).
Step 3:	Repeat Steps 1 and 2 until the elements of π_i do not change more than ε, for all i. Then return, for $i, j \in \sigma_i$: β_i, $\alpha_{i,j}$, α_i, and θ_i using (5.3.11).

In the example of Figure 5.4, $S_1(2)$ consists of only two states: $(4,(1,0))$ and $(4,(1,1))$. Only when warehouse 1 is out-of-stock ($y = 4$), the overflow demand stream $(2, 1)$ is not satisfied when On. Hence, in these states, the overflow stream $(2, \sigma_2(k))$ is On, when $\sigma_2(k-1) = 1$. Analogously, $S_1(3)$ consists of six states, given by $S_1(3) = \{(y_1, \underline{\delta}) \in \mathcal{S}_1 \mid \underline{\delta} \in \{(0,1),(1,1)\}$ and $y_1 > 2 = h_{3,1}\}$. In these states the overflow stream $(3, \sigma_3(k))$ is On, when $\sigma_3(k-1) = 1$.

The mean On and Off durations of $(i, \sigma_i(k))$ now follow from the mean times spent in subset $\mathcal{S}_{\sigma_i(k-1)}(i)$, respectively subset $\overline{\mathcal{S}}_{\sigma_i(k-1)}(i)$, from the moment on the process enters the subset, before leaving it again. Hence, the rates $\eta_{j,i}$ and $\phi_{j,i}$ follow and are given by:

$$\eta_{\sigma_i(k),i} = 1/\mathbb{E}[T(\mathcal{S}_{\sigma_i(k-1)}(i))], \quad \phi_{\sigma_i(k),i} = 1/\mathbb{E}[T(\overline{\mathcal{S}}_{\sigma_i(k-1)}(i))], \tag{5.3.10}$$

for all i and $k = 2, \ldots, |\sigma_i|$. Here we define $1/0 := \infty$ and $1/\infty := 0$. If, for example, $\mathbb{E}[T(\mathcal{S}_{\sigma_i(k-1)}(i))] = 0$, then the overflow demand stream basically skips warehouse $\sigma_i(k-1)$ and is entirely routed to warehouse $\sigma_i(k)$. This overflow process then has the same mean On and Off durations. Furthermore, recall that when On, the demand rate of the overflow stream $(i, \sigma_i(k))$ equals λ_i.

Finalization

When the algorithm terminates, it remains to calculate the β_i, $\alpha_{i,j}$, α_i, and θ_i from the π_i's. This comes down to taking the summation of π_i over certain subsets of states:

$$\begin{aligned}
\beta_i &= 1 - \sum_{(S_i, \underline{\delta}) \in \mathcal{S}_i} \pi_i(S_i, \underline{\delta}), \\
\alpha_{i,j} &= \sum_{\substack{(y,\underline{\delta}) \in \mathcal{S}_j:\ \underline{\delta}^{(i)} = 1, \\ y \leq S_j - h_{j,i} - 1}} \pi_j(y, \underline{\delta}), \text{ for } j \in \sigma_i, \text{ and 0 otherwise,} \\
\alpha_i &= \sum_{j \in \sigma_i} \alpha_{i,j}, \\
\theta_i &= 1 - \beta_i - \alpha_i.
\end{aligned} \tag{5.3.11}$$

Algorithm 2 summarizes all steps. As an additional result, the steady-state distribution of the stock levels at each of the locations, are easily derived from the π_i's calculated by the algorithm.

5.4 Numerical study

In order to determine the performance of the two presented approximation algorithms, we execute a numerical study. We first focus on the case where all hold back levels are set to 0. For that the Poisson overflow algorithm boils down to the algorithm given in [157]. Hence we can test the performance gained by the use of the On/Off approximation compared to an algorithm currently known in the literature. Then we allow for hold back levels. In both cases, we compare both the performance of the algorithms with respect to the exact outcomes (via Markov analysis as described in Section 5.2), as well as the mutual performance of the algorithms.

Testbed
We consider $N = 2$, 3, and 5 local warehouses. We perform a factorial design of the test bed given in Table 5.1. In the table, the values of λ_i, μ_i, and σ_i are given, as well as f_i. The variable f_i is the minimum fill rate of warehouse i in isolation, i.e. in a situation without any LTs. From f_i the base stock level S_i follows:

$$S_i = \min\left\{S \in \mathbb{N} \cup \{0\} \mid L\left(S, \lambda_i/\mu_i\right) \leq 1 - f_i\right\}, \qquad (5.4.1)$$

where L is the Erlang loss function: $L\left(S, \rho\right) = \left(\rho^S/S!\right) \,/\, \left(\sum_{n=0}^{S} \rho^n/n!\right)$. With respect to the possibilities for LTs, we distinguish three types of pooling strategies:

- complete overflow: $\sigma_i = \{i+1, \ldots, i+N-1\} \mod N$ (is complete pooling when no hold back levels are set);
- one step: $\sigma_i = \{i+1 \mod N\}$;
- all to 1: $\sigma_i = \{1\}$ for $i \neq 1$ and $\sigma_1 = \emptyset$.

For $N = 2$, 3, and 5, we test 384, 1500, respectively 2500 instances *without* hold back levels, and 1000, 3000, respectively 5000 instances *with* hold back levels. That is, for $N = 2$ without hold back levels, we perform a full factorial design of the settings given in Table 5.1, and for the other cases we randomly select the indicated number of instances from the full factorial designs. We restrict our attention to a single hold back level per warehouse, i.e. $\underline{h}_i = (\bar{h}_i, \ldots, \bar{h}_i, 0, \bar{h}_i, \ldots, \bar{h}_i)$ with a 0 on the i-th position. For the instances with hold back levels, we take $\bar{h}_i \in \{0, 1, \ldots, S_i - 1\}$, excluding instances where $\{\bar{h}_1, \ldots, \bar{h}_N\} = \{0, \ldots, 0\}$. So, we can both have instances where all hold back levels are positive, as well as instances with combinations of zero and positive hold back levels. For all instances we run:

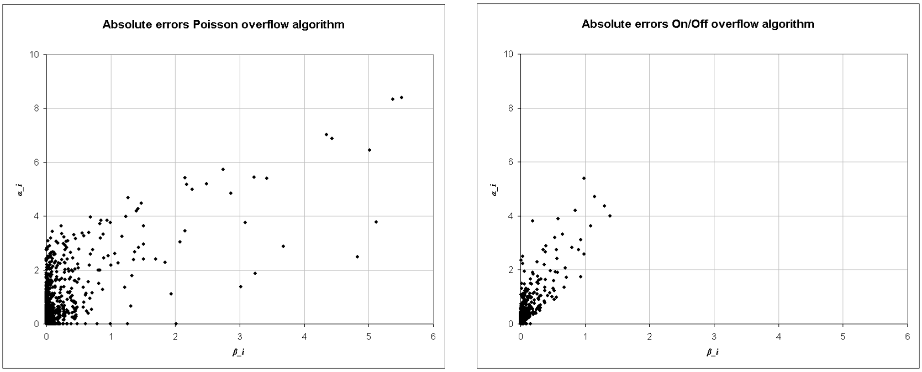
- Poisson overflow algorithm, using $\varepsilon = 10^{-10}$;
- On/Off overflow algorithm, using $\varepsilon = 10^{-10}$;
- Exact Markov analysis (see Section 5.2).

We concentrate on the average and maximum absolute errors in the β_i, α_i, and θ_i ($i = 1, \ldots, N$). Let

$$\Delta\beta_i = |\beta_{i,approx} - \beta_{i,exact}| * 100.$$

Parameter	Set of values
$\mu_1, \mu_2, \ldots, \mu_5$	1,
f_1, f_2, f_3, f_4, f_5	50%,70%,90%,95% $\rightarrow$ determines $S_1, \ldots, S_5$ using (5.4.1),
$\bar{h}_i$	$0, 1, \ldots, S_i - 1$ (for all i),
λ_1	$0.2, 0.5, 1, 2,$
λ_2, λ_3	$0.2, 0.5, 1,$
λ_4, λ_5	1,
$(\sigma_1, \ldots, \sigma_N)$	$\sigma_i = \{i+1, i+2, \ldots, i+N-1\}$ mod N, $i = 1, \ldots, N$; $\sigma_i = \{i+1\}$ mod N, $i = 1, \ldots, N$; $\sigma_1 = \emptyset$, $\sigma_i = \{1\}$, $i = 2, \ldots, N$.

Table 5.1: Test bed for numerical study: factorial design of the given possibilities.



(a) Poisson overflow algorithm *(b)* On/Off overflow algorithm

Figure 5.5: Scatter plot of the absolute relative errors in α_i versus β_i, for $N = 2$ ($i = 1, 2$, in each plot 2,768 data points).

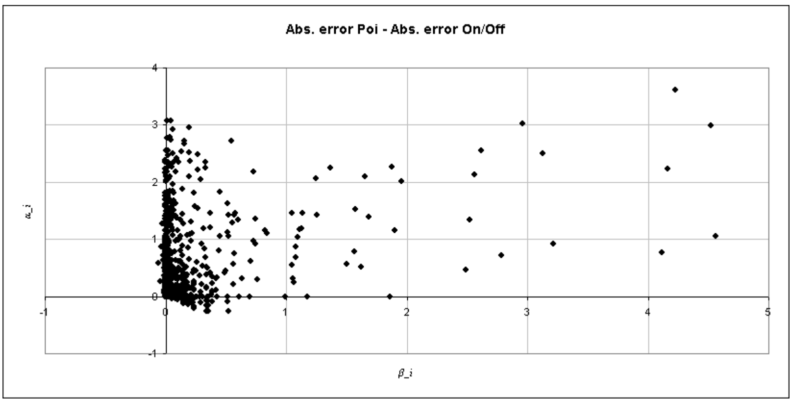


Figure 5.6: Scatter plot of the absolute relative errors, in the Poisson overflow algorithm minus those in the On/Off overflow algorithm, plotted for α_i versus β_i, for $N = 2$ ($i = 1, 2$, 2,768 data points). Explanation: each data point on the right of (above) 0 indicates a case where the On/Off overflow algorithm performs better for β_i (α_i); on the left (below) a case where the Poisson overflow algorithm performs better for β_i (α_i).

That is, we consider the error $\Delta\beta_i$ as the differences in the *percentages* $\beta_{i,approx}$ and $\beta_{i,exact}$. By 'av$\Delta\beta$' and 'max$\Delta\beta$' we denote the average respectively maximum over all $\Delta\beta_i$ (α and θ analogously).

The summary of all results is given in Table 5.2, which gives the average absolute errors and the maximal absolute errors for β_i, α_i, and θ_i for both the Poisson overflow algorithm as well as the On/Off overflow algorithm. From these results, it turns out that the On/Off algorithm clearly outperforms the Poisson algorithm. The average error for On/Off in β is about four times smaller. Also, the maximal errors are much smaller. Already from $N = 3$ on the errors in the On/Off algorithm are almost nil. In Figures 5.5 and 5.6 the errors are graphically represented. Again it becomes clear that the On/Off overflow algorithm outperforms the Poisson overflow algorithm. In the following, we further investigate the results.

Hold back levels

When we split out the results according to whether we include hold back levels or not, the results are given in Table 5.3. Both algorithms perform better for the case *with* hold back levels, about a factor two to three. Under the presence of hold back levels, a higher fraction of the demand is already satisfied at the own warehouse, and thus the overflow to the first candidate warehouse for an LT is smaller (the overflows to next warehouse may increase in case of a positive hold back level at this first candidate). As a result, the correlation in on-hand stocks at the local warehouses will reduce and the modeling of the overflow streams as Poisson/On-Off processes becomes more accurate. Both effects lead to an increased accuracy of the approximation algorithms.

Complete pooling

For the special case of complete pooling, we compare the results to those of the approximation algorithm given in [120]. Under complete pooling, all warehouses basically act as being one large warehouse. Hence, the fraction θ_i is the same for all warehouses and can be computed exactly (by the Erlang loss formula). This is exploited in their approximation. For the rest, Kranenburg and Van Houtum's algorithm is identical to our Poisson overflow algorithm. As θ_i is determined exactly, the absolute errors in β_i and α_i are always equal.

The results for complete pooling are given in Table 5.4. Both our On/Off algorithm and Kranenburg and Van Houtum's algorithm perform significantly better than the Poisson algorithm. Due to the exact calculation of the θ_i's, Kranenburg and Van Houtum's algorithm performs slightly better than our On/Off algorithm. However, the errors are of the same order, and thus we may conclude that the On/Off algorithm achieves (almost) the same accuracy as Kranenburg and Van Houtum's algorithm, without the use of the exact θ_i's.

Pooling

In the testbed, we distinguish three types of pooling strategies, which we have denoted as complete overflow, one step, and all to 1. Note that for $N = 2$ the first two strategies coincide. We split out the results according to these pooling strategies, see Table 5.5. Clearly, the results are best for the all-to-1 strategy, and remarkably of the same order for both the complete overflow and the one step strategies.

	Poisson overflow alg.						On Off overflow alg.					
N (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$
all N (13384)	0.016	5.510	0.070	8.390	0.061	3.406	0.004	1.384	0.025	5.396	0.021	4.406
$N=2$ (1384)	0.144	5.510	0.635	8.390	0.559	3.406	0.038	1.384	0.223	5.396	0.190	4.406
$N=3$ (4500)	0.001	0.062	0.005	0.098	0.005	0.054	0.000	0.040	0.002	0.088	0.002	0.054
$N=5$ (7500)	0.002	0.062	0.004	0.102	0.004	0.039	0.001	0.036	0.002	0.073	0.002	0.039

Table 5.2: Summary of numerical results: average and maximal absolute relative errors in β_i, α_i, and θ_i, for both the Poisson overflow algorithm and the On/Off overflow algorithm.

	Poisson overflow alg.						On Off overflow alg.					
N (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$
all N (4384)	0.025	5.510	0.089	8.390	0.074	3.406	0.008	1.384	0.040	5.396	0.033	4.406
$N=2$ (384)	0.267	5.510	0.961	8.390	0.805	3.406	0.087	1.384	0.440	5.396	0.362	4.406
$N=3$ (1500)	0.001	0.040	0.003	0.088	0.003	0.054	0.001	0.040	0.003	0.088	0.003	0.054
$N=5$ (2500)	0.003	0.062	0.006	0.102	0.005	0.039	0.001	0.036	0.001	0.073	0.001	0.039

(a) Zero hold back levels

	Poisson overflow alg.						On Off overflow alg.					
N (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$
all N (9000)	0.012	5.113	0.061	6.441	0.055	3.339	0.003	0.851	0.018	4.201	0.016	3.350
$N=2$ (1000)	0.097	5.113	0.509	6.441	0.465	3.339	0.019	0.851	0.139	4.201	0.125	3.350
$N=3$ (3000)	0.001	0.062	0.006	0.098	0.006	0.045	0.000	0.025	0.002	0.067	0.002	0.046
$N=5$ (5000)	0.001	0.022	0.003	0.053	0.003	0.033	0.001	0.016	0.003	0.031	0.002	0.030

(b) With hold back levels

Table 5.3: Summary of numerical results: average absolute relative errors and maximal absolute errors, for β_i, α_i, and θ_i, $i = 1,\dots,N$.

Compl. pool.	Poisson overflow alg.						On Off overflow alg.						Kranenburg & Van Houtum			
N (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$
all N (1539)	0.060	5.510	0.215	8.390	0.165	3.406	0.023	1.384	0.112	5.396	0.091	4.406	0.026	1.622	0.026	1.622
$N = 2$ (192)	0.461	5.510	1.681	8.390	1.296	3.406	0.170	1.384	0.868	5.396	0.706	4.406	0.179	1.622	0.179	1.622
$N = 3$ (501)	0.002	0.040	0.008	0.088	0.007	0.054	0.002	0.040	0.008	0.088	0.007	0.054	0.003	0.038	0.003	0.038
$N = 5$ (846)	0.004	0.062	0.005	0.102	0.002	0.039	0.002	0.036	0.003	0.073	0.002	0.039	0.004	0.051	0.004	0.051

Table 5.4: Complete pooling: summary of numerical results. Results of algorithm Kranenburg & Van Houtum added for comparison (for which by construction $\Delta\theta_i \equiv 0$).

Pooling strategy	Poisson overflow alg.						On Off overflow alg.					
strategy (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$
compl. overflow (4694)	0.039	5.510	0.162	8.390	0.133	3.406	0.012	1.384	0.066	5.396	0.056	4.406
one step (4749)	0.037	5.510	0.161	8.390	0.133	3.406	0.011	1.384	0.065	5.396	0.055	4.406
all to 1 (4633)	0.006	2.017	0.032	2.420	0.038	2.420	0.001	0.155	0.003	0.191	0.004	0.191

Table 5.5: Results of Table 5.2 split out to pooling strategy. Note: for $N = 2$ 'complete overflow' and 'one step' coincide.

Fill rate at 1	Poisson overflow alg.						On Off overflow alg.					
fill rate (# inst.)	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$	*av* $\Delta\beta$	*max* $\Delta\beta$	*av* $\Delta\alpha$	*max* $\Delta\alpha$	*av* $\Delta\theta$	*max* $\Delta\theta$
$f_1 = 50\%$ (2072)	0.038	5.510	0.134	8.390	0.111	3.406	0.011	1.305	0.051	5.396	0.041	4.406
$f_1 = 70\%$ (2737)	0.025	5.374	0.102	8.328	0.087	2.993	0.007	1.384	0.038	4.720	0.032	3.570
$f_1 = 90\%$ (3802)	0.011	3.418	0.060	5.438	0.055	2.418	0.003	0.989	0.021	3.313	0.018	2.659
$f_1 = 95\%$ (4773)	0.005	1.505	0.031	2.838	0.031	2.418	0.001	0.523	0.010	1.664	0.009	1.349

Table 5.6: Results of Table 5.2 split out to fill rate at location 1.

Fill rates
We specified four fill rates in the testbed: 50%, 70%, 90%, and 95%. In Table 5.6 we split out the results according to the fill rate at location 1. The accuracy of both algorithms increases as the fill rate increases. This is due to similar effects as when one goes from zero to positive hold back levels. Note that the number of instance increases in f_1, as the higher f_1 the higher S_1, allowing for more possibilities for $\bar{h}_1$.

Calculation time
The calculation time of the Poisson algorithm is extremely fast, it needs merely a fraction of a second to run. In every iteration, for all N warehouses a Markov process on $S_i + 1$ states has to be solved. In the numerical study, on average the algorithm converged in 8.4 iterations, where the On/Off algorithm used on average 9.4 iteration before convergence. The On/Off algorithm requires also more calculation time in every iteration, but is still reasonably fast. In every iteration a Markov process with $(S_i + 1) \cdot 2^{|d_i|}$ states has to be solved, for all N warehouses. Hence, its speed depends on the pooling strategy chosen: complete overflow is slower than 'one step' and 'all to 1', as in the first case the Markov processes contains more states than in the latter two cases. However, there is a large gain compared to exact evaluation, which requires solving the steady-state distribution of a Markov process with $\prod_{i=1}^{N}(S_i + 1)$ states. Furthermore, in the cases with zero hold back levels, both the Poisson and the On/Off algorithm needs on average two extra steps to converge, compared to the case hold back levels are set. This is due to the fact that with hold back levels, more demands are directly satisfied at the warehouse itself, and hence the overflow stream will be smaller.

5.5 Advantage of hold back levels

In this section we illustrate the advantage of hold back levels and we introduce a simple and effective heuristic for the setting of hold back levels under given base stock levels. We assume the following cost structure. Location i incurs holding costs c_i per item per time unit, where holding costs are also charged for items in repair. A demand at location i satisfied by an LT incurs costs p_i^{lt}, regardless of the location from which the part is transshipped. When the demand is satisfied by an emergency procedure, the costs are p_i^{ep} ($\geq p_i^{lt}$). We focus on the long run average costs for all locations together, denoted by C:

$$C = \sum_{i=1}^{N} \big(c_i\, S_i + \lambda_i\, \alpha_i\, p_i^{lt} + \lambda_i\, \theta_i\, p_i^{ep} \big).$$

We propose the following *greedy approach* as a heuristic to set the hold back levels. We start by setting all hold back levels to 0. Then, the $h_{i,j}$ which decreases the average costs most, is increased by 1 (ties are broken with equal probabilities). We repeat this until there is no cost reduction possible any more. For the evaluations of given policies that are needed in this heuristic, we use exact evaluations. Notice, that the Poisson or On/Off overflow algorithm could be used as well and would lead to much faster calculation times. However, in this section we use the exact evaluation procedure to exclude the effect of inaccuracies within the evaluations.

We consider a series of symmetric instances, with $N = 3$ location, and $\lambda_i = 2$ and $\mu_i = 1$ for all i. Furthermore, $\sigma_i = \{i + 1, i + 2\} \bmod 3$ for all i, and $c_i = 1$ for all i.

setting		$S_i = 2$: opt. vs.			$S_i = 3$: opt. vs.			$S_i = 4$: opt. vs.			$S_i = 5$: opt. vs.		
$p_i^{lt}:p_i^{ep}$	p_i^{ep}	np	cp	hr	np	cp	hr	np	cp	hr	np	cp	hr
1:1	5	0.0	22.4	0.0*	0.0	16.5	0.0*	0.0	6.1	0.0*	0.0	1.1	0.0*
	10	0.0	25.7	0.0*	0.0	21.8	0.0*	0.0	9.8	0.0*	0.0	2.1	0.0*
	25	0.0	28.2	0.0*	0.0	27.1	0.0*	0.0	15.5	0.0*	0.0	4.4	0.0*
	50	0.0	29.2	0.0*	0.0	29.5	0.0*	0.0	19.2	0.0*	0.0	6.7	0.0*
	100	0.0	29.7	0.0*	0.0	30.9	0.0*	0.0	21.8	0.0*	0.0	9.2	0.0*
1:2	5	1.4	4.4	0.0*	5.0	1.7	0.0*	5.7	0.5	0.0*	2.9	0.1	0.0*
	10	1.7	5.3	0.0*	7.1	2.4	0.0*	9.5	0.8	0.0*	5.4	0.2	0.0*
	25	2.0	6.0	0.0*	9.5	3.3	0.0*	16.0	1.5	0.0*	11.3	0.5	0.0*
	50	2.1	6.3	0.0*	10.7	3.7	0.0*	20.8	2.0	0.0*	17.8	0.9	0.0*
	100	2.1	6.4	0.0*	11.4	4.0	0.0*	24.4	2.5	0.0*	25.0	1.3	0.0*
1:3	5	5.4	0.0	0.0*	11.1	0.0	0.0*	9.2	0.0	0.0	4.1	0.1	0.0*
	10	6.5	0.0	0.0*	15.7	0.0	0.0*	15.4	0.1	0.1	7.7	0.1	0.0*
	25	7.4	0.0	0.0*	20.9	0.0	0.0*	25.9	0.2	0.2	16.1	0.2	0.0*
	50	7.7	0.0	0.0*	23.6	0.0	0.0*	33.6	0.2	0.2	25.4	0.4	0.0*
	100	7.9	0.0	0.0*	25.1	0.0	0.0*	39.4	0.3	0.3	35.7	0.6	0.0*
1:4	5	9.7	0.0	0.0*	15.0	0.0	0.0*	11.1	0.0	0.0*	4.7	0.0	0.0
	10	11.6	0.0	0.0*	21.2	0.0	0.0*	18.6	0.0	0.0*	8.9	0.0	0.0
	25	13.2	0.0	0.0*	28.2	0.0	0.0*	31.3	0.0	0.0*	18.6	0.1	0.1
	50	13.8	0.0	0.0*	31.7	0.0	0.0*	40.6	0.0	0.0*	29.3	0.1	0.1
	100	14.2	0.0	0.0*	33.9	0.0	0.0*	47.6	0.0	0.0*	41.1	0.2	0.2

Table 5.7: Results for the test bed of Section 5.5: relative decrease in costs when using the optimal hold back policy instead of respectively no pooling (np), complete pooling (cp), and a hold back policy where the hold back levels are determined by the greedy heuristic (hr, where 0.0^* *denotes that the heuristic and optimal hold back levels coincide).*

For base stock levels, we take $S_i \in \{2, 3, 4, 5\}$. This corresponds to service percentages of 70%, 90%, 95%, respectively 99% for satisfying the demands either directly from stock or via an LT, under the assumption of complete pooling. We also vary the costs for satisfying demands by emergency procedures ($p_i^{ep} \in \{5, 10, 25, 50, 100\}$) and the ratio between the costs for an emergency procedure and an LT ($p_i^{lt} : p_i^{ep} \in \{1{:}1, 1{:}2, 1{:}3, 1{:}4\}$).

To find the optimal hold back policy, we conduct an exhaustive search over all $h_{i,j} \in \{0, 1, \ldots, S_i\}$. Next, we consider the following policies: (i) no pooling (i.e., hold back levels equal to the base stock levels); (ii) complete pooling (i.e., zero hold back levels); (iii) a hold back policy where the hold back levels are determined via the greedy approach. For each of these policies, we determine the relative decrease in costs when switching from this policy to the optimal policy. The comparison with no pooling and complete pooling shows the beneficial effect of hold back levels. The comparison with the third policy shows the effectiveness of the greedy approach for the hold back levels. The results are listed in Table 5.7.

The results show that the optimal hold back policy is better than the no pooling policy in all instances with $p_i^{ep} > p_i^{lt}$, and in many instances the improvement is large. The optimal hold back policy is considerably better than complete pooling in those instances where p_i^{ep}/p_i^{lt} is not too large. In fact, especially at medium levels of p_i^{ep}/p_i^{lt}, we see that the optimal hold back policy may be significantly better than both the no pooling and complete pooling policy.

The comparison of the optimal hold back policy to the policy obtained via the greedy approach shows that the greedy approach finds the optimal hold back levels in 70 out of 80 instances. In the other 10 instances, the optimal policy is at most 0.3 % more expensive than the policy obtained by the greedy approach. Hence, we may conclude that the greedy heuristic performs very well (i.e., within our small test bed), and it may be a good basis for the development of heuristics for the determination of both base stock and hold back levels.

5.6 Conclusion and further research

In this chapter, we introduced the On/Off overflow algorithm for the approximation of the performance characteristics of a multi-location inventory model with both LTs and hold back levels. In this algorithm we approximate overflow demand streams more accurately than currently done in the literature, by using interrupted Poisson processes. We compared the performance of this algorithm to an extended version of the Poisson overflow algorithm (extended compared to Reijnen et al. [157], the extension is made for the hold back levels). In an extensive numerical study, the On/Off algorithm turned out to be very accurate, and considerably more accurate than the Poisson overflow algorithm.

The approximate evaluation algorithms can be used to optimize hold back and base stock levels, where heuristic optimization procedures are needed for large-size instances. The development of such heuristics requires further research. As an illustration, we showed that, under given base stock levels, a greedy heuristic works well for the hold back levels. We also showed that hold back levels may reduce costs significantly compared to no pooling and complete pooling.

We assumed a pre-specified static order per location by which it will contact other warehouses for an LT request. Instead, one could consider dynamic orders, which take actual on-hand stocks into account. Recently, this has been studied by Tiemessen et al. [181]. They show that in multi-location networks with delivery time constraints, dynamic policies instead of static policies (with zero hold back levels) reduces costs by 5-10% for many instances in their test bed. In their study, base stock levels are taken as given and hold back levels are implicit (whether the last part(s) are taken away from a local warehouse is also determined dynamically). Obviously, the above cost difference would be smaller when hold back levels are incorporated in the static policies. What is most convenient in applications in practice, static or dynamic policies, depends on multiple factors, such as requirements with respect to information systems and the people who are involved at the tactical and operational decision levels.

6

STOCK RATIONING IN A SYSTEM WITH BACKORDERS AND LOST SALES

We study a single-location stock rationing problem with two demand classes. Demands can be (i) satisfied directly from stock, (ii) backordered, or (iii) satisfied via an emergency procedure. We derive the optimal policy structure which minimizes the long-run average costs, where the costs consist of inventory holding and backordering costs per time-unit, and one-time penalty costs per backorder and per emergency procedure. We show that a high priority demand is always satisfied from stock, if on-hand stock is available. Furthermore, we address the similarity between this model and the two-location lateral transshipment problem of Chapter 4.

6.1 Introduction

In this chapter we return our attention to a stock rationing problem, where a single stockpoint faces two types of customer demands. However, we relax our assumption of Chapter 2 concerning the way a demand can be fulfilled. In Chapter 2 we had two possibilities: a demand is satisfied either directly from stock or via an emergency procedure. Now we add the possibility of backordering a demand. Each demand class has its own one-time backorder costs and one-time emergency costs, and costs per outstanding backorder per time unit are incurred. We want to optimize the decision which of the three options is taken, in case of a demand from either of the two classes. We show that the optimal policy is described by state-dependent threshold levels, i.e. by monotone switching curves. Note that a distinctive feature of this model is that there can be outstanding backorders *and* on-hand stock at the same time, e.g. because demands of lower priority are backordered, holding back on-hand stock for higher priority demands. We show the remarkable similarity between this model and the two-location lateral transshipment problem of Chapter 4. We again use the terminology of a spare parts provisioning system, see Remark 1.2.1. Note that essentially an emergency procedure can be interpreted as a lost sales, as from a modeling point of view, these coincide. Hence, we refer to the problem as a stock rationing problem with backorders and lost sales.

The initial motivation for this model comes from a stock rationing problem with two demand classes (high and low priority), where the high priority demands are always satisfied from stock, if possible, and lost otherwise, whereas low priority demands can be backordered. Enders et al. [69] study this model, where a single, static *critical level* determines whether a low priority demand is backordered or satisfied from stock, and whether an arriving replenished part (i.e. repair completion) is used to increase the stock level or to decrease the number of outstanding backorders. They provide an exact evaluation procedure and an extensive performance evaluation, comparing their critical level policy to other policies.

In [69], three application areas for the model under consideration are discussed. The first one occurs when the stockpoint faces demands from both loyal, long term customers with high service level requirements, and occasional walk-in customers. Here, inventory can be held back from the occasional customers by backlogging their demands, to be able to satisfy the loyal customers from stock. In line with our spare parts provisioning motivation (see Section 1.2.2), is the application of an OEM, which is both operating a central warehouse as well as fulfilling emergency demands from a set of local warehouses. Such an emergency demand has a higher priority than a replenishment order of a local warehouse, which might be backlogged when on-hand inventory is low. Another application of a model with both backorders and lost sales is a combination of a physical store and an on line shop. The customers in the physical shop are typically satisfied from the on-hand stock, if possible, whereas the on line customers can be backlogged when the on-hand stock is low, as they anticipate some lead time anyways.

This model fits into the stream of literature considering single-location, stock rationing models with multiple customer classes (see Section 2.1 for a discussion of the literature on these models). In that literature, however, typically models with either purely backorders are studied, or models with only lost sales (emergency procedures). Below we discuss the literature that includes both of these options, as in the model of this chapter. We start by discussing the literature on models with an optimal dynamic policy, and then turn our attention to those with static decision rules.

The optimal policy structures for a model similar to ours, with a single production (i.e. repair) server, are derived in Benjaafar et al. [21]. Our type of results strongly resemble these. However, we consider a different setting. The main difference is that we study a model with ample repair (i.e. production) capacity under a base stock policy, whereas in [21] a model with a single production server is studied, with a state-dependent order-up-to level. As we motivate our study with a closed-loop system providing repairable spare parts (see Remark 1.2.1), a base stock policy is a natural choice here.

Benjaafar and ElHafsi [20] study is a model with a patient class of customers, whose demands can be backordered, and an impatient class, of which the demands must be satisfied directly or are lost otherwise. The way in which the demands are handled is similar to that of [69]. They derive the optimal dynamic policy structures for this model. Also Ha [95] studies a model with both on-hand stock and backorders at the same time, for a problem with two customer classes. He proves that a single switching curve both determines the optimal decision for whether to satisfy or backlog a class 2 demand, as well as for whether to increase the on-hand stock level or decrease the number of outstanding class 2 backorders in case of a production completion. He, however, does not include the option of emergency procedures.

In Cattani and Souza [46] and Alvarez et al. [4] backorder-lost sales models with a static policy are studied. In Cattani and Souza [46] six different policies are compared for a single-location stock rationing problem with two demand classes. They either use lost sales, backlogging or a combination of both, where for each two policies are considered: either no stock rationing is applied, or rationing using a static critical level. Their focus, however, is on shipment flexibility. They conclude that inventory rationing and shipment flexibility are possibilities to increase profitability. They show the benefit of backlogging low priority demands before being stocked-out, and rejecting these demands when inventory is depleted.

Alvarez et al. [4] allow backorders and lost sales only in case of a stock-out. They assume that the decision is only based on the demand class of a customer. More specifically, they partition the set of demand classes into two subsets, where for one subset emergency procedures are used, while the demands of the other subset are backlogged. Although addressing the issue that a dynamic policy would lead to cost reductions, they do not take this kind of policies into account in their study. Moreover, they impose a static, critical level on the on-hand stock, which determines whether the on-hand stock is increased (if the on-hand stock is below the critical level), or a backorder is cleared. In an extensive numerical experiment they show the combination of both policies to outperform so-called 'one-size-fits-all' strategies by 14%.

Models with both backorders and lost sales have also been studied for a periodic review setting, assuming zero lead times. In Tang et al. [178] the optimality of a so-called base stock rationing policy is proven, which consists of the combination of a base stock level and a (single) rationing level in each period. In a numerical study, they compare the optimal policy to two heuristics, and find that the benefits of the optimal control policy can be significant. As in [69], they differentiate the use of backorders and lost sales based on the demand class. A similar model is studied in Zhou and Zhao [220, 221] for multiple demand classes, and in Sobel and Zhang [170] for a combination of deterministic and random demands. Finally, Rabinowitz et al. [156] limit the maximum number of accumulated backorders in one period, where this number is used as a control variable.

The contribution of this chapter is as follows. We study a new model with both backorders and lost sales for both demand classes and ample repair capacity. We prove the optimal policy structure, which is a state-dependent threshold type policy, and determine when the optimal policy simplifies. Moreover, we identify the relation between this backorder-lost sales model and the two-location lateral transshipment model.

The outline of this chapter is as follows. We start by introducing the model in more detail and the notation in Section 6.2. In this section, we also present the dynamic programming formulation and make the connection with the model of Chapter 4. Then, in Section 6.3, we derive the structural results, from which we derive the structure of the optimal policy. We also show when the optimal policy simplifies. We outline a model variation in Section 6.4, and end with conclusions in Section 6.5. All proofs are given in Appendix 6.A.

6.2 Model and notation

In Section 6.2.1 we introduce the problem, followed by its dynamic programming formulation in Section 6.2.2. We introduce the *value function* (the n-period minimal cost function) and the event operators. In Section 6.2.3 we discuss the connection with the two-location lateral transshipment problem of Chapter 4.

6.2.1 Problem description

We consider a single stockpoint, keeping repairable spare parts of a single type on stock for technically advanced machines. Upon a breakdown of a machine, it demands a spare part. A demand can be (i) satisfied directly from stock, (ii) backordered, or (iii) satisfied via an emergency procedure. When the demand is directly fulfilled from stock, the ready-for-use spare part is installed in the machine. The failed part is brought back to the stockpoint, where it is repaired and added to stock again. In this way, the down time of the machine is reduced to a minimum. In case the demand is not directly satisfied from stock (which might be the case even if on-hand stock is available), there are two remaining options. The demand can be satisfied via an expensive emergency repair procedure. Then, the failed part is repaired in a fast repair procedure, e.g. on-site, after which the machine is working properly again. Otherwise, the demand can be backlogged. As the machine is down and hence not making any revenue, backlog cost per outstanding order per time unit are incurred. Also in this case, the failed part is brought back to the stockpoint. There it is repaired, and used to satisfy an outstanding backorder. Backlogs are satisfied in order of arrival.

Initially, there are $S \in \mathbb{N}_0$ parts on stock. Holding costs are incurred at rate $\tilde{h}(\cdot)$ per time unit, as a function of the number of parts on stock, where $\tilde{h}(\cdot)$ is non-negative, non-decreasing, and convex. There are $J = 2$ demand classes. We refer to these as class j demands, more specifically as high ($j = 1$) and low ($j = 2$) priority demands. Hence, the set of technical systems served using the spare parts of this stock point, can be divided into two subsets, of which the one is of higher importance than the other. Each demand consists of the request for one part, and for each class, the demand arrivals form a Poisson process with rate λ_j, $j = 1, 2$. Upon a demand request, a decision has to be taken on how to fulfill it. For this, both the direct costs, as well as the future costs should be taken into account.

We set the costs for satisfying a demand directly from stock to 0, as these costs are made anyways, and hence would only add a constant to the cost function. This option is only possible if on-hand stock is available. The costs for fulfilling a demand by an emergency procedure are denoted by p_j, $j = 1, 2$. When a demand is backordered, we charge a one-time penalty cost b_j, and costs $\tilde{b}(\cdot)$ per time unit as a function of the total number of outstanding backorders. We assume $0 \leq b_j \leq p_j$ for $j = 1, 2$, and $p_1 \geq p_2$, $b_1 \geq b_2$. Hence, not satisfying a class 1 customers' demand is more expensive, and hence this demand class is of higher priority than class 2.

Note that the backlog costs per time unit are equal for both demand classes. If these were different, say $\tilde{b}_j(\cdot)$ for class j, one would need a three-dimensional model in order to keep track of the on-hand stock, the number of outstanding class 1 backorders, and the number of outstanding class 2 backorders. By assuming them equal, we obtain a two-dimensional model. This model is, from a mathematical point of view, almost identical to the model of Chapter 4. Hence, all structural results readily follow. Moreover, we prove that it is always suboptimal to not satisfy a class 1 demand directly from stock, when on-hand stock is available. We can generalize this result to the case where $\tilde{b}_1(i) \geq \tilde{b}_2(i)$. Hence, when we allow class 1 demands to be backlogged or satisfied by emergency procedures only when out-of-stock, this does not affect the model. We then interpreted the backorders of class 1 as negative on-hand stock, and hence have a two-dimensional model again. However, we need to make a small adjustment to the way in which repairs are used, in order for this to work. Therefore, we discuss this setting, with class dependent backlog costs per time unit, as a model variation in Section 6.4.

For the purpose of analyzing the model, we limit the maximum number of outstanding backorders to be B. This, however, is not restrictive, as B can always be chosen that large that it does not influence the costs nor the optimal policy.

When a demand is satisfied from stock or the demand is backlogged, the failed part is brought back to the stockpoint. This part is repaired and, after repair completion, either added to stock or used to satisfy a backorder. We assume that there are two repair shops: one for the repairs of parts to clear the backlog and one for the repair of parts that will be added to stock. The first mentioned are referred to as *backlog repairs*, whereas the latter are referred to as *stock repairs*. All repair lead times are exponentially distributed with rate $\mu > 0$, where we assume that ample repair capacity for both repair shops. That is, the repair lead time of each part is exponentially distributed with mean $1/\mu$, independently of the number of parts in repair. Hence, the rate of repair completions is linear in the number of outstanding repairs at a repair shop. We assume all arrival and repair processes to be mutually independent.

6.2.2 Dynamic programming formulation

Let the state of the system be given by (x, y), where x is the on-hand inventory, $x \in \{0, 1, \ldots, S\}$, and $B - y$ is the number of outstanding backorders, $y \in \{0, 1, \ldots, B\}$. By letting $B - y$ (instead of y) be the number of outstanding backorders, all the transition rates and operators turn out to be identical to those in the model of Chapter 4 (see Section 6.2.3). Now, y is the maximum number of outstanding backorders B, minus the actual number of outstanding backorders. The state space $\mathcal{S}$ is given by

$$\mathcal{S} = \{(x, y) \mid x \in \{0, 1, \ldots, S\}, y \in \{0, 1, \ldots, B\}\}.$$

As the interarrival times of demands as well as the repair lead times are independent, exponentially distributed random variables, we can apply uniformization (see [131]) to convert the semi-Markov decision problem into an equivalent Markov decision problem (MDP). For this, we use our technical assumption that there is a maximum number of outstanding backorders, and that hence the maximum rate out of a state is finite. The existence of a stationary average costs optimal policy is guaranteed by [155, Theorem 8.4.5a], as the state space and action space for every state are finite, the costs are bounded and the model is both unichain and aperiodic.

Let $V_n : \mathcal{S} \mapsto \mathbb{R}$ be the value function, given by

$$V_{n+1}(x, y) = C\Big(U\big(H_1 V_n(x, y), H_2 V_n(x, y), G_S V_n(x, y), G_B V_n(x, y)\big)\Big)$$

starting with $V_0 \equiv 0$, where C is the costs operator, U the uniformization operator, H_j the demand operator for class j customers, $j = 1, 2$, and G_S and G_B are the repair operators corresponding to a stock repair, respectively a backorder repair. All operators are defined below. Let $\nu = \lambda_1 + \lambda_2 + (S + B)\mu$ be the uniformization rate.

The cost operator C is defined by

$$Cf(x, y) = h(x) + b(B - y) + f(x, y),$$

where $h(x) = \tilde{h}(x)/\nu$ and $b(B - y) = \tilde{b}(B - y)/\nu$.

The uniformization operator U is, for this model, defined by:

$$U(f_1, f_2, f_3, f_4) = \frac{1}{\nu}\Big(\lambda_1 f_1 + \lambda_2 f_2 + \mu f_3 + \mu f_4\Big). \tag{6.2.1}$$

The operator H_j models the demand of a class j customer, and is given by:

$$H_j f(x,y) = \begin{cases} \min\{f(x-1,y), b_j + f(x,y-1), p_j + f(x,y)\} & \text{if } x>0, y>0, \\ \min\{b_j + f(x,y-1), p_j + f(x,y)\} & \text{if } x=0, y>0, \\ \min\{f(x-1,y), p_j + f(x,y)\} & \text{if } x>0, y=0, \\ p_j + f(x,y) & \text{if } x=0, y=0. \end{cases} \tag{6.2.2}$$

Basically, there are three options to fulfill a demand in state (x, y): directly from stock (possible when $x > 0$, decreasing the on-hand stock level x by one), by backordering it (possible when $y > 0$, at direct costs b_j, decreasing y by one), or via an emergency shipment (at direct costs p_j).

The operator G_S models the (potential) stock repairs and is given by:

$$G_S f(x,y) = (S-x)f(x+1,y) + xf(x,y). \tag{6.2.3}$$

In state (x, y) there are $S-x$ outstanding stock orders. Hence, with this rate the on-hand stock is increased by one. The term $xf(x,y)$ represents *fictitious* transitions, hence assuring that the total rate at which μG_S occurs is equal to μS. Analogously, the operator G_B models the (potential) backorder repairs and is given by:

$$G_B f(x,y) = (B-y)f(x,y+1) + yf(x,y). \tag{6.2.4}$$

REMARK 6.2.1. In the setting with linear backlog costs, when one assumes *no maximum level* on the number of outstanding backorders, the system can be reduced to a one-dimensional model. For this, only keep track of the on-hand stock level x, and note that each backlogged demand has essentially its own repair process (as we assumed ample repair capacity), with mean duration $1/\mu$, independently of the number of other outstanding backorders and on-hand stock level. Hence upon a demand arrival, the decision can be made only based on the on-hand stock level.
However, the current model allows for non-linear backlog costs, as well as variations of the assumptions for the repair process, such as state-dependent repair rates, and e.g. a model with a single repair server for the stock orders and one for the backorders.
The two-location lateral transshipment problem does not reduce to a one-dimensional model, as the number of parts on-hand at location 2 is limited (compare: there is a maximum level on the number of outstanding backorders). The reasoning above can only be applied to this model when one assumes unlimited on-hand stock at location 2.

REMARK 6.2.2. The extension to multiple customers classes, say $j = 1, 2, \ldots, J$ is easily made, by use of the operator H_j. Instead of the two terms in the value function involving H_1 and H_2, one would get $\sum_{j=1}^{J} \lambda_j H_j V_n(x,y)$, and $\sum_{j=1}^{J} \lambda_j$ in the uniformization rate ν (i.e. in the denominator in (6.2.1)). The values of b_j and p_j can be set differently for each of the classes.

6.2.3 Connection with two-location lateral transshipment problem

The model of this chapter, and in particular that of this section, is closely related to the two-location inventory model with lateral transshipments of Chapter 4, see Figure 6.1. Essentially, the mathematical model is the same, where we give another interpretation to the variables and events. In Chapter 4 we deal with two stockpoints, each having their own inventory, demand streams and repairs. Here we have one inventory point (stockpoint 1), where stockpoint 2 resembles the outstanding backorders. Hence, the base stock levels S_1 and S_2 of Chapter 4, are equivalent to the base stock level S respectively maximum number of outstanding backorders B in this chapter.

We now only have stock at a single point, say at location 1. The demand operator H_j in (6.2.2) then is the same as H_1 (!) as defined in (4.3.3), for both $j = 1, 2$ (with $P_{LT_1} = b_j$ and $P_{EP_1} = p_j$). Note that H_2 in (6.2.2) does *not* coincide with H_2 (as defined analogously to (4.3.3)), because one has a different cost structure for the possible transitions.

Because of this, we *cannot* reuse the conditions (4.4.9)–(4.4.12) of Theorems 4.4.6 and 4.4.7 to simplify the structure of the optimal policy. However, we derive (other) conditions in Section 6.3.3, which also take into account the fact that we have holding and backlog costs.

In Figure 6.1 the Markov processes of both problems are shown next to each other. For the two-location lateral transshipment problem, demands and repairs at both locations are depicted. The rates and costs are indicated. For the stock rationing problem with backorders and lost sales, demands of class 1 and 2, a stock repair, and a backorder repair are indicated. There is pairwise correspondence between these four events. The main difference, however, is the ordering of the costs, for a demand at location 2, i.e., for a class 2 demand. In the two-location problem, the costs for the transition from (x, y) to $(x, y-1)$ are smaller than those for the transition to $(x-1, y)$. Contrary, in the backorder–lost sales problem, the opposite is true for a class 2 demand. The other difference between the two models, is that we added backlog and holding costs for the backorder–lost sales problem. When one assumes that holding costs are also charged for parts in repair, as in Chapter 4, the holding costs are a constant, and hence can be left out of the model. For generality, we have added them here.

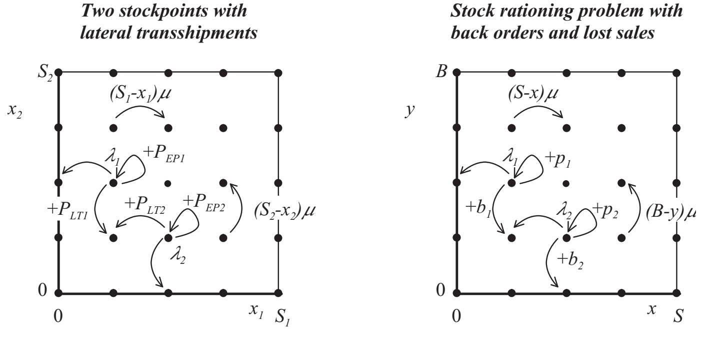


Figure 6.1: The Markov processes for the two-location lateral transshipment problem (Chapter 4), and that of the backorder-lost sales stock rationing problem of this chapter.

6.3 Structural results

In this section we prove our main result: the structure of the optimal policy. For this we first prove that the value function V_n is multimodular. From this we derive the structure of the optimal lateral transshipment policy, which is a threshold type policy. We illustrate this by an example. Finally, we give conditions under which the optimal policy structure simplifies.

6.3.1 Properties of operators and value function

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all (x, y) such that the states appearing in the right-hand and left-hand side of the inequalities exist in $\mathcal{S}$:

$$\begin{aligned}
\text{BFOD}(1, p_1): &\quad f(x+1, y) + p_1 \geq f(x, y), && (6.3.1)\\
\text{BFODD}(1, 2, b_1): &\quad f(x+1, y) + b_1 \geq f(x, y+1), && (6.3.2)\\
\text{Decr}(2): &\quad f(x, y) \geq f(x, y+1), && (6.3.3)\\
\text{Conv}(1): &\quad f(x, y) + f(x+2, y) \geq 2f(x+1, y), && (6.3.4)\\
\text{Conv}(2): &\quad f(x, y) + f(x, y+2) \geq 2f(x, y+1), && (6.3.5)\\
\text{Supermod}: &\quad f(x, y) + f(x+1, y+1) \geq f(x+1, y) + f(x, y+1), && (6.3.6)\\
\text{SuperC}(1, 2): &\quad f(x+2, y) + f(x, y+1) \geq f(x+1, y) + f(x+1, y+1), && (6.3.7)\\
\text{SuperC}(2, 1): &\quad f(x, y+2) + f(x+1, y) \geq f(x, y+1) + f(x+1, y+1), && (6.3.8)\\
\text{MM}: &\quad \text{Supermod} \cap \text{SuperC}(1, 2) \cap \text{SuperC}(2, 1). && (6.3.9)
\end{aligned}$$

In the proof of the following lemma, parts *a)* and *b)* reuse results of Chapter 4 for the propagation of MM.

LEMMA 6.3.1. *a) Operator H_j, $j = 1, 2$, preserves Decr(2) and MM.*
b) The sum of the operators $G_S + G_B$ preserves Decr(2) and MM.
c) $C\big(U\big(H_1, H_2, G_S, G_B\big)\big)$ preserves (i) BFOD$(1, p_1)$ and (ii) BFODD$(1, 2, b_1)$.

Parts *a)* and *b)* are a direct consequence of Lemmas 4.4.1 and 4.4.2, respectively. Part *c)* is proven in the Appendix. It uses the assumptions $p_1 \geq p_2$ and $b_1 \geq b_2$ for part *(i)* and part *(ii)*, respectively. By induction on n, the following result directly follows from Lemma 6.3.1.

THEOREM 6.3.2. *V_n satisfies properties* (6.3.1)–(6.3.9) *for all $n \geq 0$.*

6.3.2 Structure of optimal policy

We now characterize the structure of the optimal policy. For this, we note that the results in Theorems 4.4.4 and 4.4.5 hold as well, providing the structure of the optimal policy, i.e. the existence of a monotone switching curve (dynamic threshold level) for the optimal decision when to satisfy a class j demand from stock, when to backorder it, and when to satisfy it by an emergency procedure. The optimal policy is graphically depicted in Figure 6.2 (compare to Figure 4.1 for the relation with the two-location lateral transshipment problem).

Denote by $a_j^*(x, y)$ the optimal decision for a class j demand when in state (x, y).

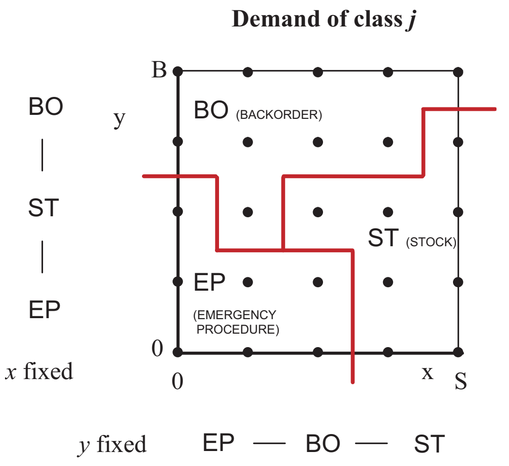


Figure 6.2: Structure of the optimal policy for fulfilling a class j demand.

THEOREM 6.3.3. *a) The optimal policy for fulfilling a demand from class j for fixed y is a threshold type policy: for each* $y \in \{0,1,\ldots,B\}$*, there exist thresholds* $T_j^{bo}(y) \in \{0,1,\ldots,S+1\}$ *and* $T_j^{st}(y) \in \{1,\ldots,S+1\}$*, with* $T_j^{bo}(y) \le T_j^{st}(y)$*, such that:*

$$a_j^*(x,y) = 2 \text{ (emergency procedure), for } 0 \le x \le T_j^{bo}(y) - 1;$$
$$a_j^*(x,y) = 1 \text{ (backorder), for } T_j^{bo}(y) \le x \le T_j^{st}(y) - 1;$$
$$a_j^*(x,y) = 0 \text{ (directly from stock), for } T_j^{st}(y) \le x \le S,$$

where $T_j^{bo}(0) = T_j^{st}(0) \ge 1$.
b) $T_1^{st}(y) = 1$ *for all* $y \in \{0,1,\ldots,B\}$*, that is*

$$a_1^*(x,y) = 0 \text{ (directly from stock), for all } x \ge 1, y \ge 0.$$

This structure is graphically represented below the horizontal axis in Figure 6.2. If $y = 0$, only one threshold describes the optimal policy for when to fulfill the demand from stock, and when to apply an emergency procedure.

The intuition behind this theorem is as follows. If the stock level x is high, one is willing to take a part from stock as there are still plenty left afterwards. But if the stock level is low, one might, depending on the cost parameters, decide to hold some parts back for future higher priority demands. By part *b)*, high priority demands are always satisfied from stock, if on-hand stock is available.

This is intuitively correct, as there are no incentives for holding back parts for customers from this demand class.

A similar characterization of the optimal policy can be made for fixed x, which is given in the following theorem.

THEOREM 6.3.4. *For the optimal policy for fulfilling a demand of class j for fixed* $x \in \{0,1,\ldots,S\}$*, there exist* $\hat{T}_j^{st}(x) \in \{0,1,\ldots,B+1\}$ *and* $\hat{T}_j^{bo}(x) \in \{1,\ldots,B+1\}$*, with*

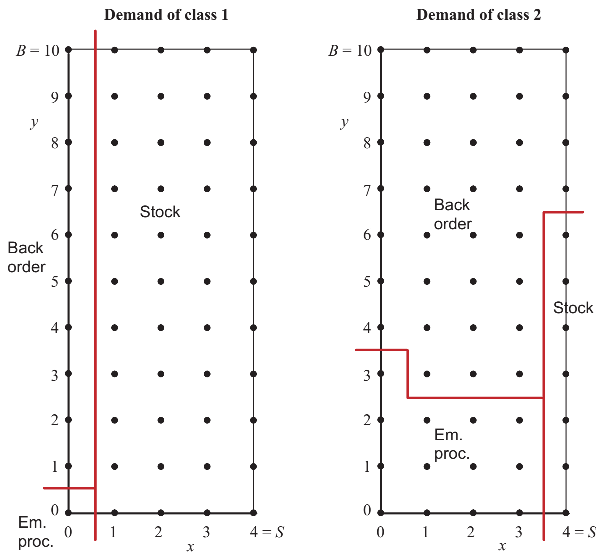


Figure 6.3: Optimal policy for Example 6.3.1.

$\hat{T}_j^{st}(x) \leq \hat{T}_j^{bo}(x)$, *such that:*

$$a_j^*(x) = 2 \text{ (emergency procedure), for } 0 \leq y \leq \hat{T}_j^{st}(x) - 1;$$
$$a_j^*(x) = 0 \text{ (direct from stock), for } \hat{T}_j^{st}(x) \leq y \leq \hat{T}_j^{bo}(x) - 1;$$
$$a_j^*(x) = 1 \text{ (backorder), for } \hat{T}_j^{bo}(x) \leq y \leq B,$$

where $\hat{T}_j^{st}(0) = \hat{T}_j^{bo}(0) \geq 1$.

This structure is graphically represented next to the vertical axis in Figure 6.2. If $x = 0$, a single threshold describes the optimal policy, for either backordering or applying an emergency procedure.

Note that if $b_j = p_j$, one never backorders a demand. This follows from the minimization in the definition of H_j, as V_n is Decr(2), cf. (6.3.3).

As in Chapter 4, combining Theorems 6.3.3 and 6.3.4 restricts the possibilities for the optimal policy significantly, see the discussion in Section 4.4.2.

Example 6.3.1. Let $\lambda_1 = 3$, $\lambda_2 = 10$, $\mu = 1$, $S = 4$, $B = 10$ and $h(x) = 0.1x$, $b(y) = \ln(11/(11-j))$. Furthermore, let $p_1 = 100$, $b_1 = 50$ and $p_2 = 15$, $b_2 = 2$. Then the optimal policy is as given in Figure 6.3. Note that it satisfies the optimal policy structure described by Theorems 6.3.3 and 6.3.4, which is depicted in Figure 6.2.

This optimal policy structure for satisfying the demands is in line with the optimal policy structure of [21]. In Enders et al. [69] a static, critical level is assumed which determines whether a class 2 demand is satisfied from stock, or backlogged. Although this policy satisfies the optimal policy structure, it typically is sub-optimal. In [69] it is argued that the main advantage of such a single, static critical level is that it is easily explained to practitioners and that it is easy to implement, as it does not depend on other factors, such as the repair pipeline. In a testbed over 1,232 instance, they show an

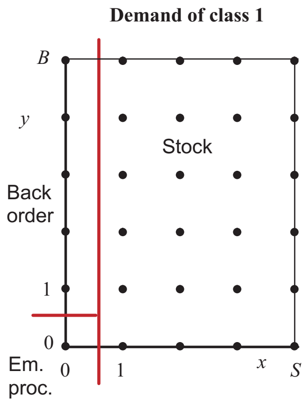


Figure 6.4: Simplified optimal policy structure when condition (6.3.10) *is satisfied.*

average relative difference of 2.1%, comparing the costs of the optimal state-dependent policy, versus the optimal critical level policy.

6.3.3 Conditions simplifying the optimal policy

Under a sufficient condition on the cost parameters, the structure of the optimal policy is simplified.

THEOREM 6.3.5. *If*

$$\frac{\lambda_2}{\nu} \min\{(p_1 - b_1) - (p_2 - b_2), 0\} \geq b(B) - b(B-1), \tag{6.3.10}$$

then $\hat{T}_1^{bo}(0) = 1$.
Also, if

$$\frac{\lambda_1}{\nu} \min\{(p_2 - b_2) - (p_1 - b_1), 0\} \geq b(B) - b(B-1), \tag{6.3.11}$$

then $\hat{T}_2^{bo}(0) = 1$.

That is, the optimal action is to always backorder a demand in case of a stock-out (unless the maximum number of backorders is already reached).

Hence, under condition (6.3.10), the optimal policy for a class 1 demand is a greedy policy: use a part from stock, if possible, otherwise backlog the demand, if possible, and only when out-of-stock and already fully backordered, apply an emergency procedure (see Figure 6.4). This is the equivalent of a complete pooling strategy at location 1 for the two-location model of Chapter 4.

In Example 6.3.1, condition (6.3.10) is clearly satisfied, hence the optimal policy found for class 1 demands in this example (see Figure 6.3, left) coincides with the simplified optimal policy structure of Figure 6.4.

REMARK 6.3.6. In Enders et al. [69] a slightly different version of the options for satisfying the demands is studied. Namely, a class 1 demand is directly satisfied from stock, if possible, and lost otherwise; and a class 2 demand may be directly satisfied from stock, or can be backordered. In [69] a static threshold is used to determine which option is used for a class 2 demand. The optimization, however, is a special case of our setting. To achieve this, set $b_1 = p_1$ in (6.2.2), then the option to backorder a class 1 demand is always suboptimal. Moreover, by (6.3.1), a demand is always satisfied from stock, if possible. For a class 2 demand, if one lets $p_2 \to \infty$, the option of an emergency procedure is always suboptimal. One should take B large enough, to avoid boundary effects influencing the optimal policy.

REMARK 6.3.7. With the approach followed in this chapter, one is not able to optimize the decision whether a repair completion is used to increase the stock level or to decrease the number of outstanding backorders. The operator that would be used to optimize this decision does not (necessarily) propagate the property Supermod, cf. (6.3.6). As a consequence, also MM (6.3.9) is not propagated. Hence, the value function does not (necessarily) satisfy the properties (6.3.4)–(6.3.9), and the structural results derived in Section 6.3 does not have to hold.
Although examples for which the value function does not satisfy (6.3.6) are easily constructed, we did not succeed in finding a counter example violating the optimal policy structure of Theorems 6.3.3 and 6.3.4. In such an example, the optimal policy is not described by a switching curve. This means that there is no monotonicity in the optimal actions when either x or y is fixed.
This problem arises in models that have (a finite or infinite number of) parallel severs. For a model with a single repair server (for both backorders and stock) the optimization is possible. Such an assumption is made in Benjaafar et al. [21] for a model similar to ours (see also Ha [95]). In this case, one is then also able to prove the optimality of a state-dependent base stock level. That is, one optimizes the decision whether to repair a part for stock, clear a backorder, or do neither of the two.
Next to the optimized decisions, the following (heuristic) policies for using repair completions also lead to value functions that do not necessarily satisfy properties (6.3.4)–(6.3.9) on $\mathcal{S}$:

- Use all repair completions to increase the stock level until $x = S$, then use them to decrease the number of outstanding backorders (until $y = B$);
- Use all repair completions to decrease the number of outstanding backorders until $y = B$, then use them to increase the stock level (until $x = S$);
- Use a fixed threshold, as in e.g. [4, 69].

6.4 Model variation

In this section we outline a model variation, allowing asymmetric backlog cost per time unit for class 1 and class 2 demands. In this way, the model becomes similar to the model studied in Benjaafar et al. [21] (with a slightly different assumption on the repair rates), to which we refer for the analysis.

We have assumed that the backlog cost per time unit are equal for both class 1 and class 2 demands (see Section 6.2.1). In this way, the obtained model is two-dimensional. For this model, we proved that a class 1 demand is always satisfied from stock, if possible (see Section 6.3.2). Hence, one can restrict the options for how a class 1 demand is satisfied, without changing the model, by only allowing the options of backlogging and emergency procedures when out-of-stock. One now counts class 1 backlogs as negative on-hand stock (one can do so, as class 1 backlogs only occur when $x = 0$). Only the class 2 backorders are now counted in the direction of y. In this way, the class 1 and class 2 backorders are separated. Hence, one can now allow for different backlog cost per time unit, while keeping a two-dimensional model. There is an extra assumption necessary for this to work, namely that class 1 backorders are first cleared, before parts are added to stock. Moreover, one cannot charge one-time backlog cost for class 1 demands: as these are non-convex in x, the resulting value function would not necessarily be convex in x, and the propagation of the structural properties cannot be guaranteed.

Let $b_j(\cdot)$ be the backlog costs per time unit as a function of the number of outstanding class j backlogs, $j = 1, 2$, assuming $b_j(\cdot)$ to be non-negative, non-decreasing, and convex for $j = 1, 2$, such that $b_1(i) \geq b_2(i)$ for all i. Let B_j be the maximum number of outstanding backorders for class j. The state of the system is still given by (x, y), where x is now the stock level, such that $x^+ = \max\{x, 0\}$ is the on-hand inventory, and $x^- = -\min\{x, 0\}$ is the number of outstanding class 1 backorders. Hence, $x \in \{-B_1, \ldots, 0, 1, \ldots, S\}$, and $B_2 - y$ is the number of outstanding class 2 backorders, $y \in \{0, 1, \ldots, B_2\}$. The state space $\tilde{\mathcal{S}}$ is given by

$$\tilde{\mathcal{S}} = \{(x, y) \mid x \in \{-B_1, \ldots, 0, 1, \ldots, S\}, y \in \{0, 1, \ldots, B_2\}\}.$$

One has to adapt the demand operators H_1 and H_2:

$$\tilde{H}_1 f(x, y) = \begin{cases} \min\{f(x-1, y), p_j + f(x, y)\} & \text{if } x > -B_1, \\ p_j + f(x, y) & \text{if } x = -B_1, \end{cases}$$

and

$$\tilde{H}_2 f(x, y) = \begin{cases} \min\{f(x-1, y), b_2 + f(x, y-1), p_2 + f(x, y)\} & \text{if } x > 0, y > 0, \\ \min\{b_2 + f(x, y-1), p_2 + f(x, y)\} & \text{if } x \leq 0, y > 0, \\ \min\{f(x-1, y), p_2 + f(x, y)\} & \text{if } x > 0, y = 0, \\ p_2 + f(x, y) & \text{if } x \leq 0, y = 0. \end{cases}$$

One has to adapt the other operators to fit the state space $\tilde{\mathcal{S}}$.

The resulting model is similar to the model of [21]. They assume a single production server for the replenishments (where we assume ample repair capacity), and hence they are also able to optimize the production decision (see Remark 6.3.7). Also, we bounded the state space for computational purposes by the use of maximum numbers of outstanding backorders, which is not a restriction, as these can be taken arbitrarily large. Hence, the analysis follows along the same lines as in [21], where is it proven that the optimal policy structure for satisfying the demands is again given by a state-dependent threshold policy for both classes.

6.5 Conclusion

In this chapter we studied a stock rationing problem which combines the options of backorders and emergency procedures. We derived the optimal policy structure, minimizing the long-run average costs, for a problem with two demand classes. Furthermore, we derived conditions under which the optimal policy simplified, and outlined a model variation with different backlog costs per time unit for the two demand classes.

For further research, it would be interesting to investigate whether one can derive conditions such that value function remains Supermod, when the use of repairs is optimized (see Remark 6.3.7). This also is an open question for the heuristic rules for the use of the repairs. Both need further study.

6.A Appendix: Proofs

6.A.1 Proof of Lemma 6.3.1

PROOF. *a)* By Lemma 4.4.2, parts *(i)* and *(ii)*, as the definitions of H_j in (6.2.2) and H_1 as in (4.3.3) coincide, for $j = 1, 2$.
b) By Lemma 4.4.1, parts *b), (i)* and *(v)*, as one can write $G = G_S + G_B$, where the definitions of G_S and G_1 as in (4.3.2) coincide, and analogously the definitions of G_B and G_2 coincide.
c) (i) Assume that f satisfies (6.3.1). Then, for $y > 0$:

$$
\begin{aligned}
&H_j f(x+1,y) + p_1 \\
&= \min \begin{cases} f(x,y) + p_1 & \geq f(x,y) + p_j \\ b_j + f(x+1,y-1) + p_1 & \geq b_j + f(x,y-1) \\ p_j + f(x+1,y) + p_1 & \geq p_j + f(x,y) \end{cases} \\
&\geq \min\{f(x-1,y),\ b_j + f(x,y-1),\ p_j + f(x,y)\} = H_j f(x,y),
\end{aligned}
$$

for $j = 1, 2$, since f satisfies property (6.3.1), and $p_1 \geq p_j$ for all j. The case $y = 0$ proceeds along the same lines.
Also, for $x < S$, $y < B$, writing $G = G_S + G_B$:

$$
\begin{aligned}
&Gf(x+1,y) + (S+B)p_1 \\
&= (S-x-1)f(x+2,y) + (B-y)f(x+1,y+1) + (x+y+1)f(x+1,y) + (S+B)p_1 \\
&= (S-x-1)\big(f(x+2,y)+p_1\big) + (B-y)\big(f(x+1,y+1)+p_1\big) \\
&\qquad + (x+y)\big(f(x+1,y)+p_1\big) + f(x+1,y) + p_1 \\
&\geq (S-x-1)f(x+1,y) + (B-y)f(x,y+1) + (x+y)f(x,y) + f(x+1,y) + p_1 \\
&= (S-x)f(x+1,y) + (B-y)f(x,y+1) + (x+y)f(x,y) + p_1 \\
&\geq (S-x)f(x+1,y) + (B-y)f(x,y+1) + (x+y)f(x,y) = Gf(x,y),
\end{aligned}
$$

since f satisfies property (6.3.1) (used in the first inequality), and $p_1 \geq 0$ (second inequality). The cases $x = S$, $y < B$, and $x < S$, $y = B$, and $x = S$, $y = B$ proceed along the same lines. Then $U\big(H_1, H_2, G_S, G_B\big) f$ satisfies property (6.3.1) as well:

$$\begin{aligned}
&U\big(H_1, H_2, G_S, G_B\big) f(x+1, y) + p_1 \\
&= \frac{1}{\lambda_1 + \lambda_2 + (S+B)\mu} \Big(\lambda_1 H_1 + \lambda_2 H_2 + \mu G\Big) f(x+1,y) + p_1 \\
&= \frac{1}{\lambda_1 + \lambda_2 + (S+B)\mu} \Big(\lambda_1 (H_1 f(x+1,y) + p_1) + \lambda_2 (H_2 f(x+1,y) + p_1) \\
&\qquad\qquad\qquad\qquad\qquad\qquad + \mu (G f(x+1,y) + (S+B) p_1)\Big) \qquad (6.A.1) \\
&\geq \frac{1}{\lambda_1 + \lambda_2 + (S+B)\mu} \Big(\lambda_1 H_1 f(x,y) + \lambda_2 H_2 f(x,y) + \mu G f(x,y)\Big) \\
&= \frac{1}{\lambda_1 + \lambda_2 + (S+B)\mu} \Big(\lambda_1 H_1 + \lambda_2 H_2 + \mu G\Big) f(x,y) \\
&= U\big(H_1, H_2, G_S, G_B\big) f(x,y).
\end{aligned}$$

This leads to

$$\begin{aligned}
&C\big(U\big(H_1, H_2, G_S, G_B\big)\big) f(x+1, y) + p_1 \\
&= h(x+1) + b(B-y) + U\big(H_1, H_2, G_S, G_B\big) f(x+1,y) + p_1 \\
&\geq h(x+1) + b(B-y) + U\big(H_1, H_2, G_S, G_B\big) f(x,y) \qquad (6.A.2) \\
&\geq h(x) + b(B-y) + U\big(H_1, H_2, G_S, G_B\big) f(x,y) \\
&= C\big(U\big(H_1, H_2, G_S, G_B\big)\big) f(x,y),
\end{aligned}$$

where the first inequality holds by (6.A.1), and the second as $h(x)$ is non-decreasing in x.
(ii) Assume that f satisfies (6.3.2). Then, for $x > 0$, $y > 0$:

$$\begin{aligned}
&H_j f(x+1, y) + p_1 \\
&= \min \begin{cases} f(x,y) + b_1 & \geq f(x,y) + b_j \\ b_j + f(x+1, y-1) + b_1 & \geq b_j + f(x,y) \\ p_j + f(x+1,y) + b_1 & \geq p_j + f(x, y+1) \end{cases} \\
&\geq \min\{f(x-1, y+1),\ b_j + f(x,y),\ p_j + f(x,y+1)\} = H_j f(x, y+1),
\end{aligned}$$

for $j = 1, 2$, since f satisfies property (6.3.2), and $b_1 \geq b_j$ for all j. The cases $x > 0$, $y = 0$, and $x = 0$, $y > 0$, and $x = 0$, $y = 0$ proceed along the same lines.
Also, for $x < S$, $y < B$, writing $G = G_S + G_B$:

$$\begin{aligned}
&G f(x+1, y) + (S+B) b_1 \\
&= (S-x-1) f(x+2, y) + (B-y) f(x+1, y+1) + (x+y+1) f(x+1, y) + (S+B) b_1 \\
&= (S-x-1)\big(f(x+2,y) + b_1\big) + (B-y)\big(f(x+1,y+1) + b_1\big) \\
&\qquad + (x+y+1)\big(f(x+1,y) + b_1\big) \\
&\geq (S-x-1) f(x+1, y+1) + (B-y)\big(f(x+1,y+1) + b_1\big) + (x+y+1) f(x, y+1) \\
&= (S-x) f(x+1, y+1) + (B-y-1)\big(f(x+1,y+1) + b_1\big) + (x+y+1) f(x, y+1) + b_1 \\
&\geq (S-x) f(x+1, y+1) + (B-y-1) f(x, y+2) + (x+y+1) f(x, y+1) + b_1 \\
&\geq (S-x) f(x+1, y+1) + (B-y-1) f(x, y+2) + (x+y+1) f(x, y+1) = G f(x, y+1),
\end{aligned}$$

since f satisfies property (6.3.1) (used in the first two inequalities), and $b_1 \geq 0$ (third inequality). The cases $x = S$, $y < B$, and $x < S$, $y = B$, and $x = S$, $y = B$ proceed along the same lines.Then $U\big(H_1, H_2, G_S, G_B\big)f$ satisfies property (6.3.2) as well:

$$
\begin{aligned}
&U\big(H_1,H_2,G_S,G_B\big)f(x+1,y)+b_1\\
&= \frac{1}{\lambda_1+\lambda_2+(S+B)\mu}\Big(\lambda_1 H_1+\lambda_2 H_2+\mu G\Big)f(x+1,y)+b_1\\
&= \frac{1}{\lambda_1+\lambda_2+(S+B)\mu}\Big(\lambda_1 H_1(f(x+1,y)+b_1)+\lambda_2(H_2 f(x+1,y)+b_1)\\
&\qquad\qquad +\mu(Gf(x+1,y)+(S+B)b_1)\Big) \qquad (6.A.3)\\
&\geq \frac{1}{\lambda_1+\lambda_2+(S+B)\mu}\Big(\lambda_1 H_1 f(x,y+1)+\lambda_2 H_2 f(x,y+1)+\mu G f(x,y+1)\Big)\\
&= \frac{1}{\lambda_1+\lambda_2+(S+B)\mu}\Big(\lambda_1 H_1+\lambda_2 H_2+\mu G\Big)f(x,y+1)\\
&= U\big(H_1,H_2,G_S,G_B\big)f(x,y+1).
\end{aligned}
$$

Analogously to (6.A.2), now using inequality (6.A.3), this leads to

$$C\big(U\big(H_1,H_2,G_S,G_B\big)\big)f(x+1,y)+b_1 \geq C\big(U\big(H_1,H_2,G_S,G_B\big)\big)f(x,y+1). \qquad \square$$

6.A.2 Proof of Theorem 6.3.3

PROOF. a) The proof is along the same lines as the proof of Theorem 4.4.4.
b) By inequality (6.3.1) it directly follows that $T_1^{st}(0) = 1$. By inequality (6.3.2) it follows that $T_1^{st}(y) = 1$ for all $y \geq 1$. $\square$

6.A.3 Proof of Theorem 6.3.5

PROOF. We prove the result for class 1 demands, as the result for class 2 demands follows along the same lines, by interchanging the demand classes. We prove that $a_1^*(0,1) = 1$ (backorder), then it follows by Theorem 6.3.4 that $\hat{T}_1^{bo}(0) = 1$. It suffices to prove that, for all $n \geq 0$:

$$V_n(0,1) + p_1 - b_1 \geq V_n(0,0). \qquad (6.A.4)$$

We prove this inequality by induction. For $V_0 \equiv 0$, (6.A.4) trivially holds, as by assumption $p_1 \geq b_1$. Assume that (6.A.4) holds for a given n (induction hypothesis). We consider the operators H_1, H_2, and $G_S + G_B$ separately. All given inequalities hold by the induction hypothesis, unless stated otherwise.

$$
\begin{aligned}
&H_1 V_n(0,1) + p_1 - b_1 - H_1 V_n(0,0)\\
&= \min\{V_n(0,0)+p_1, V_n(0,1)+2p_1-b_1\} - \big(V_n(0,0)+p_1\big)\\
&\geq V_n(0,0)+p_1-\big(V_n(0,0)+p_1\big) \geq 0.
\end{aligned}
$$

$$H_2 V_n(0,1) + p_1 - b_1 - H_2 V_n(0,0)$$

$$
\begin{aligned}
&= \min\{V_n(0,0)+b_2+p_1-b_1, V_n(0,1)+p_2+p_1-b_1\} - \big(V_n(0,0)+p_2\big)\\
&\geq \min\{V_n(0,0)+b_2+p_1-b_1, V_n(0,0)+p_2\} - \big(V_n(0,0)+p_2\big)\\
&\geq \min\{(p_1-b_1)-(p_2-b_2), 0\}.
\end{aligned}
$$

$$
\begin{aligned}
&(G_S+G_B)V_n(0,1)+(S+B)(p_1-b_1)-(G_S+G_B)V_n(0,0)\\
&= SV_n(1,1)+(B-1)V_n(0,2)+V_n(0,1)+(S+B)(p_1-b_1)-SV_n(1,0)-BV_n(0,1)\\
&= S\big(V_n(1,1)+p_1-b_1-V_n(1,0)\big)+(B-1)\big(V_n(0,2)+p_1-b_1-V_n(0,1)\big)\\
&\qquad + V_n(0,1)-V_n(0,1)+p_1-b_1\\
&\geq p_1-b_1 \geq 0,
\end{aligned}
$$

where the last inequality holds by assumption. Combining these give

$$
\begin{aligned}
&V_{n+1}(0,1)+p_1-b_1-V_{n+1}(0,0) = h(0)+b(B-1)-h(0)-b(B)\\
&\qquad + \frac{1}{\nu}\Bigg(\lambda_1\Big(H_1V_n(0,1)+p_1-b_1-H_1V_n(0,0)\Big)+\lambda_2\Big(H_2V_n(0,1)+p_1-b_1-H_2V_n(0,0)\Big)\\
&\qquad + \mu\Big((G_S+G_B)V_n(0,1)+(S+B)(p_1-b_1)-(G_S+G_B)V_n(0,0)\Big)\Bigg)\\
&\geq b(B-1)-b(B)+\lambda_2\min\{(p_1-b_1)-(p_2-b_2),0\}/\nu \geq 0,
\end{aligned}
$$

where the last inequality holds by condition (6.3.10). This completes the induction step, and hence (6.A.4) holds for all $n \geq 0$. □

Pooling of server capacity

7

Optimal control of a server farm

A server farm consists of ample servers that serve a stream of arriving customers. Upon a service completion, a server can be turned off. This might be beneficial to save power, and hence costs. However, for shutting down and starting up a server, extra power (i.e. costs) is incurred. Thus, there is a trade-off between the savings by turning servers off, and the extra costs made for the start-ups and shut-downs. We consider a model where arriving customers are taken into service directly. For this, we study the optimal control of such a server farm, that is, we derive the optimal dynamic control policy deciding when a server should be turned off after a demand completion, minimizing the expected discounted long-run costs. We prove this policy to be a state-dependent threshold type policy.

7.1 Introduction

A server farm consists of an unlimited amount of servers, that serve an arriving stream of customers. Each server in the system can be in one of the following three states: busy, idle, or off. Busy servers consume power, idle servers consume less power, and off servers consume no power at all. Hence, by turning an idle server off, power, and hence costs, can be saved. However, for starting-up and shutting-down a server, extra power (i.e. costs) is incurred. In this way, there is a trade-off between the potential costs savings, and the extra costs incurred. Thus the question is, when servers should be turned off, and when they should be idled. In this chapter we derive the optimal control policy for such a server farm, answering this question.

Bell [18] studies a server farm consisting of only two servers (i.e., an $M/M/2$ system), for which he characterizes the optimal policy by four parameters. Lu and Serfozo [133] consider an $M/M/1$, where they choose the arrival and departure rate from a finite set of possibilities. Szarkowicz and Knowles [173] allow to turn on or off an arbitrary number of servers at decision epoch, i.e. they take the decision to have $u \in \{0, 1, \dots, S\}$ servers on until the next decision epoch. Artalejo et al. [8] derive the steady-state behavior of a system without control. That is, servers are turned on or off when a customer arrives respectively departs. Moreover, they assume an exponentially distributed setup time for turning a server on. In Feinberg and Zhang [77] an $M/M/\infty$ system is studied, where all servers can be turned on and off at once. An exponential setup time is required when all are turned on. They prove that the optimal policy can be described by two thresholds on the number of customers in the system.

By controlling the servers in a server farm, the energy consumption can be decreased, see e.g. [85, 86]. There is a variety of systems referred to as server farms, with different options for control. Our focus is on a server farm consisting of an unlimited number of servers, where costs are incurred for turning servers on or off, and per time unit a server is on. We assume that customers arrive according to a Poisson process, each requiring an exponentially distributed service time. An arriving customer has to be taken into service directly, either by occupying an idle server, or by switching on a server. We want to minimize the expected long-run discounted costs. When a server completes service to a customer, we can decide whether to turn this server off, or let it idle. For this system, we derive the optimal control policy, which we prove to be a dynamic state-dependent threshold type policy.

The outline of this chapter is as follows. We start by describing the model and introducing the notation in Section 7.2, where we also formulate the problem as an MDP and introduce the value function. In Section 7.3 we show that the value function satisfies certain structural results, from which we derive the optimal policy structure. We end by an example and possibilities for further research. All proofs are in the Appendix 7.A. This chapter is based on [1].

7.2 Model and notation

7.2.1 Problem description

We study a server farm consisting of an unlimited number of servers. Each server can be in one of three states: busy, idle, or off. Customers are arriving to the system according to a Poisson process with rate $\lambda > 0$. Each customer requests an i.i.d. exponentially distributed service time with mean $1/\mu$, for $\mu > 0$. If an incoming customer finds a server in idle state, he starts getting served immediately by that server, and that server state instantaneously changes to busy. If there are no idle servers, the customer occupies one of the servers turned off and starts getting served. The state of that server instantaneously changes to busy. As there is an unlimited number of servers, there is always an off server available. When a service completes at a server, we have an option of switching the server state to idle or off instantaneously. We do not allow turning an idle server off.

Costs are incurred for keeping servers on, and turning them on and off. As we require all jobs to be taken into service directly, the costs for busy servers add up to a constant, for all policies. Hence, we do not take those costs into account, since they do not influence the optimal decisions, and we only charge costs $\tilde{c}(i)$ per unit time to keep i servers idle. We assume that $\tilde{c}(i)$ is convex in i. It costs K^{on} to turn a server from off to on (i.e. busy), while it costs K^{off} to switch a server off after a service completion. We assume that the state change of a server happens immediately. When j servers are busy, the next service completion occurs at rate $\mu(j)$, which is an increasing, bounded function, with $\mu(0) = 0$ and a finite upper bound $\bar{\mu}$. This upper bound is required to be able to apply uniformization. In this way, the infinite server case $\mu(j) = j\,\mu$ is not included, however, the many server case $\mu(j) = \min\{j, M\}\mu$, for some finite M, is. We want to derive an optimal policy that minimizes the expected long-run discounted cost, with continuous discount factor $\alpha > 0$.

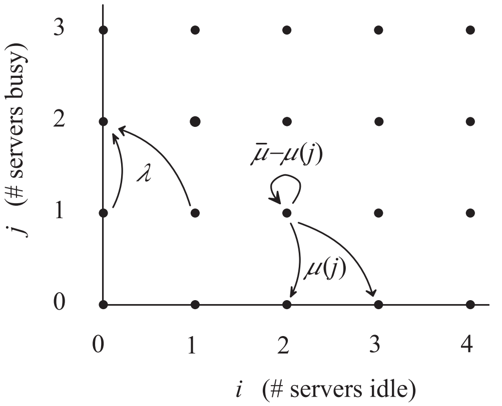


Figure 7.1: Transition rates for the server farm model.

7.2.2 Dynamic programming formulation

We model the problem as a Markov decision process (MDP, cf. [155]). Denote by i the number of servers in idle state and by j the number of servers in busy state. Then the state space $\mathcal{S}$ is given by

$$\mathcal{S} = \{(i,j) \mid i \geq 0, j \geq 0\}.$$

There are two types of events that can occur: customer arrivals and service completions. At rate λ customers arrive to the system. When it finds a server idle ($i > 0$), it directly occupies this server (at no extra costs), changing the server's state to busy. If $i = 0$, an off server has to be turned on, at costs K^{on}, and the customer directly occupies this server, changing the server's state state to busy as well. When service to a customer is completed (at rate $\mu(j)$), we have two option to choose from. Either, one can leave the server idle (at no direct extra costs), or one can decide to switch off the server, incurring costs K^{off}. Note that lower costs per time unit are made in the latter case, as one pays per time unit for each server being idle.

As the interarrival times of demands as well as the service times are independent exponentially distributed random variables, we can apply uniformization (cf. [131]) to convert the semi-Markov decision problem into an equivalent Markov decision problem (MDP). For this, we add fictitious transitions, to let the service completion event occur at rate $\bar{\mu}$. The existence of a stationary optimal policy is guaranteed by [155, Theorem 11.5.3].

Let $V_n : \mathcal{S} \mapsto \mathbb{R}$ be the *value function*, the minimum cost function when there are n events (customer arrivals or service completions) left. It is given by:

$$V_{n+1}(i,j) = c(i) + \frac{1}{\nu}\Big(\lambda T_{\text{arr}} V_n(i,j) + T_{\text{dep}} V_n(i,j)\Big), \tag{7.2.1}$$

starting with $V_0 \equiv 0$, where $\nu = \lambda + \bar{\mu} + \alpha$ is the uniformization rate, and $c(i) = \tilde{c}(i)/\nu$. The operators T_{arr} (customer arrival) and T_{dep} (service completion, i.e., departure) are defined below. The costs $c(i)$ represent the costs for keeping i servers idle during one time unit $1/\nu$. Recall that $\tilde{c}(\cdot)$, and hence $c(\cdot)$, is assumed to be convex.

The operator T_{arr} models the customer arrivals, and is defined by

$$T_{\text{arr}}f(i,j) = \begin{cases} f(i-1,j+1) & \text{if } i > 0; \\ f(i,j+1) + K^{\text{on}} & \text{if } i = 0. \end{cases}$$

When there or no idle servers ($i = 0$), a server has to be turned on (at costs K^{on}) to serve the arriving customer. When there is at least one server idle ($i > 0$), the arriving customer occupies one of these servers. Hence one server switches from idle to busy.

The operator T_{dep} models (potential) service completions (i.e., departures), and is defined by

$$T_{\text{dep}}f(i,j) = \mu(j)\min\{f(i+1,j-1),\ f(i,j-1)+K^{\text{off}}\} + (\bar{\mu}-\mu(j))f(i,j).$$

When there are j servers busy, a service completion occurs with rate $\mu(j)$. The operator T_{dep} takes the cost minimizing decision to either keep the vacant server idle, or turn it off (at costs K^{off}). The part $(\bar{\mu}-\mu(j))f(i,j)$ is a fictitious transition, to assure that the rate at which T_{dep} occurs is always equal to $\bar{\mu}$.

In Figure 7.1 the transition rates are graphically depicted.

7.3 Structural results

In this section we prove our main result: the structure of the optimal control policy for a server farm. For this, we first introduce a number of structural properties. Each of the operators in the value function preserves these properties, hence the value function satisfies them. From this the optimal policy structure is derived.

7.3.1 Properties of operators and value function

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all $i, j \geq 0$:

$$\begin{aligned} \text{Conv}(i): &\quad f(i,j)+f(i+2,j) \geq 2f(i+1,j), && (7.3.1) \\ \text{Supermod}(i,j): &\quad f(i,j)+f(i+1,j+1) \geq f(i+1,j)+f(i,j+1), && (7.3.2) \\ \text{SuperC}(i,j): &\quad f(i,j+1)+f(i+2,j) \geq f(i+1,j)+f(i+1,j+1), && (7.3.3) \\ \text{BFOD}(i,K^{\text{on}}): &\quad f(i+1,j)+K^{\text{on}} \geq f(i,j). && (7.3.4) \end{aligned}$$

Note that we prove $\text{BFOD}(i,K^{\text{on}})$ for all $i \geq 0$, although we only need it for $i = 0$.

The next two lemmas show that the operators T_{arr} and T_{dep} preserve properties (7.3.1)–(7.3.3). Furthermore, the third lemma shows that (7.3.4) is preserved as well. The proofs are given in the Appendix 7.A. Recall that $X: P_1,\dots,P_N \rightarrow P_1$ denotes that when a function f satisfies properties $P_1,\dots,P_N$, then Xf satisfies property P_1, for operator X.

LEMMA 7.3.1. *For all $i, j \geq 0$:*

1) T_{arr} : *Conv(i), BFOD(i,K^{on})* $\rightarrow$ *Conv(i)*,
2) T_{arr} : *Supermod(i,j)* $\rightarrow$ *Supermod(i,j)*,
3) T_{arr} : *SuperC(i,j), BFOD(i,K^{on})* $\rightarrow$ *SuperC(i,j)*.

	leave **idle**	turn **off**	struc. prop.
i)	$\leftarrow$: $(i-1, j)$	$\rightarrow$: $(i+1, j)$	Conv(i)
ii)	$\downarrow$: $(i, j-1)$	$\uparrow$: $(i, j+1)$	Supermod(i, j)
iii)	$\nwarrow$: $(i-1, j+1)$	$\searrow$: $(i+1, j-1)$	SuperC(i, j)

Table 7.1: If in state (i, j) the optimal action is to leave the server idle *(to turn the server* off*), then the left (middle) column indicates the states in which that is the optimal action as well (see Figure 7.2). The right column indicates the structural properties of the value function needed for that result.*

LEMMA 7.3.2. *For all $i, j \geq 0$:*

$$\begin{aligned}
&1)\ T_{dep} : Conv(i) \ \rightarrow\ Conv(i),\\
&2)\ T_{dep} : Supermod(i,j),\ SuperC(i,j) \ \rightarrow\ Supermod(i,j),\\
&3)\ T_{dep} : SuperC(i,j),\ Supermod(i,j) \ \rightarrow\ SuperC(i,j).
\end{aligned}$$

LEMMA 7.3.3. *For all $i, j \geq 0$:*

$$\frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda T_{arr} + T_{dep}\Big) : BFOD(i, K^{on}) \ \rightarrow\ BFOD(i, K^{on}).$$

REMARK 7.3.4. In both Lemma 7.3.1 and 7.3.2 it suffices to prove parts 2) and 3), since they imply 1). For completeness, we also give the proofs for 1).

Since $V_0(i, j) = 0$ for all (i, j), V_0 satisfies the properties (7.3.1)–(7.3.4). We assumed that the cost function $\tilde{c}(i)$ is convex. Moreover, taking a linear combination preserves these properties. Then, by induction on n, Lemmas 7.3.1, 7.3.2, and 7.3.3 lead to the following result:

THEOREM 7.3.5. *V_n satisfies properties* (7.3.1)–(7.3.4) *for all $n \geq 0$.*

7.3.2 Structure of optimal policy

Using the result of Theorem 7.3.5, we derive the structure of the optimal policy.

THEOREM 7.3.6. *The optimal policy is described by a switching curve $T(i)$, such that if service completes in state (i, j) it is optimal to leave the server idle if $j \leq T(i)$, and turn the server off if $j > T(i)$. Furthermore, $T(i)$ is* strictly *decreasing in i, i.e. $T(i) > T(i+1)$, for all $i \geq 0$ (until it reaches 0).*

The optimal policy structure is shown in Figure 7.2. Suppose that in state (i, j) the optimal action is to leave the server *idle*, then this is also the optimal action in states $(i-1, j)$, $(i, j-1)$, and $(i-1, j+1)$. When in state (i, j) the optimal action is to turn the server *off*, then this is also the optimal action in states $(i+1, j)$, $(i, j+1)$, and $(i+1, j-1)$ (for both, see again Figure 7.2). Table 7.1 summarizes this, and also indicates the structural properties of the value function needed for each of the statements.

We end by an example and suggestions for model variations.

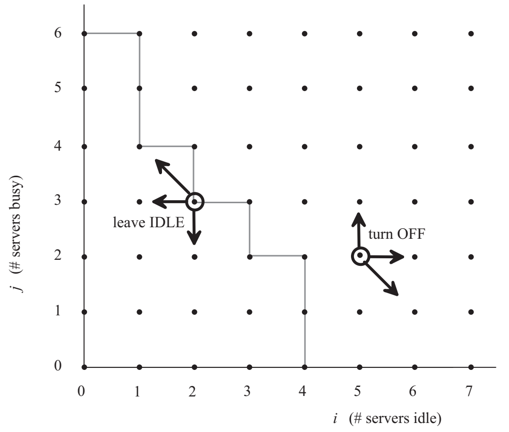


Figure 7.2: Structure of the optimal policy.

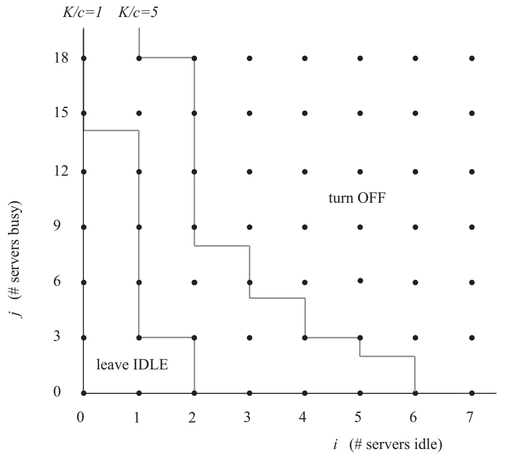


Figure 7.3: Optimal policy for Examples 7.3.1(a) and 7.3.1(b).

Example 7.3.1. Consider the following example, with $\lambda = 5$, $\mu = 1$, for $\alpha \downarrow 0$, where $K^{\text{off}} = 0$.
(a) When $K^{\text{on}}/c = 1$, the optimal policy that minimizes the long run cost per unit time is given by the following switching curve: $T(1) = 14$, $T(2) = 3$ and $T(i) = 0$ for $i \geq 3$ (see Figure 7.3), with long run cost $2.49K^{\text{on}}$.
(b) When $K^{\text{on}}/c = 5$, the optimal policy is given by $T(1) \geq 50$, $T(2) = 18$, $T(3) = 8$, $T(4) = 5$, $T(5) = 3$, $T(6) = 2$, and $T(i) = 0$ for $i \geq 7$ (see again Figure 7.3), with long run cost $.95K^{\text{on}}$.

REMARK 7.3.7. Possible extensions for the model include (i) adding the option of turning off an idle server at any time (i.e. not triggered by a service completion), (ii) limiting the number of servers, (iii) considering an unbounded rate for the total server rate, (iv) investigating the long-run average costs case, (v) adding a positive start-up and/or shut down time when turning a server on or off, e.g. following an exponential distribution, and (vi) allowing customers to queue, charging holding costs for waiting customers.

7.A Appendix: Proofs

7.A.1 Proof of Lemma 7.3.1

PROOF. 1) Suppose that f is Conv(i) and BFOD(i, K^{on}), then we show that $T_{\text{arr}}f$ is Conv(i). For $i > 0$:

$$\begin{aligned} T_{\text{arr}}f(i,j) + T_{\text{arr}}f(i+2,j) &= f(i-1,j+1) + f(i+1,j+1) \\ &\geq 2f(i,j+1) = 2T_{\text{arr}}f(i+1,j), \end{aligned}$$

since f is Conv(i), and for $i = 0$:

$$\begin{aligned} T_{\text{arr}}f(0,j) + T_{\text{arr}}f(2,j) &= f(0,j+1) + K^{\text{on}} + f(1,j+1) \\ &\geq 2f(0,j+1) = 2T_{\text{arr}}f(1,j), \end{aligned}$$

since f is BFOD(i, K^{on}).

2) Suppose that f is Supermod(i,j), then we show that $T_{\text{arr}}f$ is Supermod(i,j). For $i > 0$:

$$\begin{aligned} T_{\text{arr}}f(i,j) + T_{\text{arr}}f(i+1,j+1) &= f(i-1,j+1) + f(i,j+2) \\ &\geq f(i,j+1) + f(i-1,j+2) = T_{\text{arr}}f(i+1,j) + T_{\text{arr}}f(i,j+1), \end{aligned}$$

since f is Supermod(i,j), and for $i = 0$:

$$\begin{aligned} T_{\text{arr}}f(0,j) + T_{\text{arr}}f(1,j+1) &= f(0,j+1) + K^{\text{on}} + f(0,j+2) \\ &= T_{\text{arr}}f(1,j) + T_{\text{arr}}f(0,j+1). \end{aligned}$$

3) Suppose that f is SuperC(i,j) and BFOD(i, K^{on}), then we show that $T_{\text{arr}}f$ is SuperC(i,j). For $i > 0$:

$$\begin{aligned} T_{\text{arr}}f(i,j+1) + T_{\text{arr}}f(i+2,j) &= f(i-1,j+2) + f(i+1,j+1) \\ &\geq f(i,j+1) + f(i,j+2) = T_{\text{arr}}f(i+1,j) + T_{\text{arr}}f(i+1,j+1), \end{aligned}$$

since f is SuperC(i,j), and for $i = 0$:

$$\begin{aligned} T_{\text{arr}}f(0,j+1) + T_{\text{arr}}f(2,j) &= f(0,j+2) + K^{\text{on}} + f(1,j+1) \\ &\geq f(0,j+1) + f(0,j+2) = T_{\text{arr}}f(1,j) + T_{\text{arr}}f(1,j+1), \end{aligned}$$

since f is BFOD(i, K^{on}). □

7.A.2 Proof of Lemma 7.3.2

PROOF. The proofs of parts 1)-3) come down to case checking: applying T_{dep} to f introduces a minimization over two terms, so the sum of two gives a total of four possibilities, which we each check separately. For this we use the trivial results: $a \geq \min\{a,b\}$, and hence $a+b \geq 2\min\{a,b\}$, $\forall a,b \in \mathbb{R}$. Furthermore, we use that $\mu(j)$ is increasing in j, hence $\mu(j+1) - \mu(j) \geq 0$, and we use that $\mu(j) \leq \bar{\mu}$ for all j, hence $\bar{\mu} - \mu(j) \geq 0$ for all j. Moreover, for a few cases we use:

$$\mu(j)\,a + \big(\mu(j+1) - \mu(j)\big)\,b \geq \mu(j)\min\{a,b\} + \big(\mu(j+1) - \mu(j)\big)\min\{a,b\} = \mu(j+1)\min\{a,b\}, \tag{7.A.1}$$

for all $a, b \in \mathbb{R}$.

1) Suppose that f is Conv(i), then we show that $T_{\text{dep}}f$ is Conv(i). For $j > 0$:

$$
\begin{aligned}
&T_{\text{dep}}f(i,j) + T_{\text{dep}}f(i+2,j)\\
&= \big(\bar\mu - \mu(j)\big)\big(f(i,j) + f(i+2,j)\big)\\
&\qquad + \mu(j)\min\begin{cases}
① \ f(i+1,j-1) + f(i+3,j-1) \geq 2f(i+2,j-1)\\
② \ f(i+1,j-1) + f(i+2,j-1) + K^{\text{off}}\\
③ \ f(i,j-1) + K^{\text{off}} + f(i+3,j-1)\\
\qquad\qquad \geq 2f(i+1,j-1) + f(i+3,j-1) - f(i+2,j-1) + K^{\text{off}}\\
\qquad\qquad \geq f(i+1,j-1) + f(i+2,j-1) + K^{\text{off}}\\
④ \ f(i,j-1) + K^{\text{off}} + f(i+2,j-1) + K^{\text{off}} \geq 2f(i+1,j-1) + 2K^{\text{off}}
\end{cases}\\
&\geq 2\,\big(\bar\mu - \mu(j)\big)f(i+1,j) + 2\mu(j)\min\{f(i+2,j-1),\ f(i+1,j-1) + K^{\text{off}}\} = 2\,T_{\text{dep}}f(i+1,j),
\end{aligned}
$$

where all inequalities hold since f is Conv(i), and for $j = 0$:

$$
T_{\text{dep}}f(i,0) + T_{\text{dep}}f(i+2,0) = \bar\mu\big(f(i,0) + f(i+2,0)\big) \geq 2\,\bar\mu f(i+1,0) = 2\,T_{\text{dep}}f(i+1,0),
$$

since f is Conv(i).

2) Suppose that f is Supermod(i,j) and SuperC(i,j) (and hence Conv(i)), then we show that $T_{\text{dep}}f$ is Supermod(i,j). For $j > 0$:

$$
\begin{aligned}
&T_{\text{dep}}f(i,j) + T_{\text{dep}}f(i+1,j+1)\\
&= \big(\bar\mu - \mu(j)\big)f(i,j) + \big(\bar\mu - \mu(j+1)\big)f(i+1,j+1)\\
&\qquad + \min\begin{cases}
① \ \mu(j)f(i+1,j-1) + \mu(j+1)f(i+2,j)\\
② \ \mu(j)f(i+1,j-1) + \mu(j+1)\big(f(i+1,j) + K^{\text{off}}\big)\\
③ \ \mu(j)\big(f(i,j-1) + K^{\text{off}}\big) + \mu(j+1)f(i+2,j)\\
④ \ \mu(j)\big(f(i,j-1) + K^{\text{off}}\big) + \mu(j+1)\big(f(i+1,j) + K^{\text{off}}\big)
\end{cases}\\
&\geq \big(\bar\mu - \mu(j)\big)f(i+1,j) + \big(\bar\mu - \mu(j+1)\big)f(i,j+1) + \mu(j)\min\{f(i+2,j-1),\ f(i+1,j-1) + K^{\text{off}}\}\\
&\qquad + \mu(j+1)\min\{f(i+1,j),\ f(i,j) + K^{\text{off}}\}\\
&= T_{\text{dep}}f(i+1,j) + T_{\text{dep}}f(i,j+1).
\end{aligned}
$$

Here ① holds because:

$$
\begin{aligned}
&\big(\bar\mu - \mu(j)\big)f(i,j) + \big(\bar\mu - \mu(j+1)\big)f(i+1,j+1) + \mu(j)f(i+1,j-1) + \mu(j+1)f(i+2,j)\\
&= \big(\bar\mu - \mu(j)\big)\big(f(i,j) + f(i+1,j+1)\big) + \mu(j)\big(f(i+1,j-1) + f(i+2,j)\big)\\
&\qquad\qquad + \big(\mu(j+1) - \mu(j)\big)\big(f(i+2,j) - f(i+1,j+1)\big)\\
&\geq \big(\bar\mu - \mu(j)\big)\big(f(i+1,j) + f(i,j+1)\big) + \mu(j)\big(f(i+2,j-1) + f(i+1,j)\big)\\
&\qquad\qquad + \big(\mu(j+1) - \mu(j)\big)\big(f(i+2,j) - f(i+1,j+1)\big)\\
&= \big(\bar\mu - \mu(j)\big)f(i+1,j) + \big(\bar\mu - \mu(j+1)\big)f(i,j+1) + \mu(j)f(i+2,j-1) + \mu(j+1)f(i+1,j)\\
&\qquad + \big(\mu(j+1) - \mu(j)\big)\big(f(i+2,j) - f(i+1,j+1) - f(i+1,j) + f(i,j+1)\big)\\
&\geq \big(\bar\mu - \mu(j)\big)f(i+1,j) + \big(\bar\mu - \mu(j+1)\big)f(i,j+1) + \mu(j)f(i+2,j-1) + \mu(j+1)f(i+1,j),
\end{aligned}
$$

since f is Supermod(i,j), respectively SuperC(i,j).

And ② holds because:

$$\begin{aligned}
&\big(\bar\mu-\mu(j)\big)f(i,j)+\big(\bar\mu-\mu(j+1)\big)f(i+1,j+1)+\mu(j)f(i+1,j-1)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&=\big(\bar\mu-\mu(j+1)\big)\big(f(i,j)+f(i+1,j+1)\big)+\mu(j)f(i+1,j-1)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)f(i,j)\\
&\geq\big(\bar\mu-\mu(j+1)\big)\big(f(i+1,j)+f(i,j+1)\big)+\mu(j)f(i+1,j-1)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)f(i,j)\\
&=\big(\bar\mu-\mu(j)\big)f(i+1,j)+\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)+\mu(j+1)f(i+1,j)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}-f(i+1,j)\big)\\
&=\big(\bar\mu-\mu(j)\big)f(i+1,j)+\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)+\mu(j)f(i+1,j)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}\big),
\end{aligned}$$

since f is Supermod(i,j) and since by (7.A.1):

$$\mu(j)f(i+1,j)+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}\big)\geq\mu(j+1)\min\{f(i+1,j),\ f(i,j)+K^{\text{off}}\}.$$

For ③ first note that

$$\begin{aligned}
f(i,j-1)+f(i+2,j)&=f(i,j-1)+f(i+2,j)+f(i+1,j)-f(i+1,j)\\
&\geq f(i+1,j-1)+f(i,j)+f(i+2,j)-f(i+1,j)\\
&\geq 2f(i+1,j)+f(i+1,j-1)-f(i+1,j)\\
&=f(i+1,j)+f(i+1,j-1),
\end{aligned}\tag{7.A.2}$$

since f is Supermod(i,j), respectively Conv(i). Now ③ holds because:

$$\begin{aligned}
&\big(\bar\mu-\mu(j)\big)f(i,j)+\big(\bar\mu-\mu(j+1)\big)f(i+1,j+1)+\mu(j)\big(f(i,j-1)+K^{\text{off}}\big)+\mu(j+1)f(i+2,j)\\
&=\big(\bar\mu-\mu(j)\big)\big(f(i,j)+f(i+1,j+1)\big)+\mu(j)\big(f(i,j-1)+K^{\text{off}}+f(i+2,j)\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+1,j+1)\big)\\
&\geq\big(\bar\mu-\mu(j)\big)\big(f(i+1,j)+f(i,j+1)\big)+\mu(j)\big(f(i+1,j)+K^{\text{off}}+f(i+1,j-1)\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+1,j+1)\big)\\
&=\big(\bar\mu-\mu(j)\big)f(i+1,j)+\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)+\mu(j+1)f(i+1,j)\\
&\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+1,j+1)+f(i,j+1)-f(i+1,j)\big)\\
&\geq\big(\bar\mu-\mu(j)\big)f(i+1,j)+\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)+\mu(j+1)f(i+1,j),
\end{aligned}$$

where the first inequality holds since f is Supermod(i,j) (part with $\bar\mu-\mu(j)$) and by (7.A.2) (part with $\mu(j)$), and the second inequality holds since f is SuperC(i,j).
Finally, ④ holds because:

$$\begin{aligned}
&\big(\bar\mu-\mu(j)\big)f(i,j)+\big(\bar\mu-\mu(j+1)\big)f(i+1,j+1)+\mu(j)\big(f(i,j-1)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&=\big(\bar\mu-\mu(j+1)\big)\big(f(i,j)+f(i+1,j+1)\big)+\mu(j)\big(f(i,j-1)+K^{\text{off}}+f(i+1,j)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+f(i+1,j)+K^{\text{off}}\big)\\
&\geq\big(\bar\mu-\mu(j+1)\big)\big(f(i+1,j)+f(i,j+1)\big)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}+f(i,j)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+f(i+1,j)+K^{\text{off}}\big)\\
&=\big(\bar\mu-\mu(j)\big)f(i+1,j)+\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)\\
&\qquad\qquad\qquad\qquad+\mu(j+1)\big(f(i,j)+K^{\text{off}}\big),
\end{aligned}$$

since f is Supermod(i,j).

Furthermore, for $j = 0$:

$$\begin{aligned}
&T_{\text{dep}}f(i,0) + T_{\text{dep}}f(i+1,1) \\
&= \bar{\mu}f(i,0) + \mu(1)\min\begin{cases} ① \ f(i+2,0) \\ ② \ f(i+1,0) + K^{\text{off}} \end{cases} + \big(\bar{\mu} - \mu(1)\big)f(i+1,1) \\
&\geq \bar{\mu}f(i+1,0) + \big(\bar{\mu} - \mu(1)\big)f(i,1) + \mu(1)\min\{f(i+1,0), f(i,0) + K^{\text{off}}\} \\
&= T_{\text{dep}}f(i+1,0) + T_{\text{dep}}f(i,1),
\end{aligned}$$

where ① holds because:

$$\begin{aligned}
&\bar{\mu}f(i,0) + \mu(1)f(i+2,0) + \big(\bar{\mu} - \mu(1)\big)f(i+1,1) \\
&= \bar{\mu}\big(f(i,0) + f(i+1,1)\big) + \mu(1)\big(f(i+2,0) - f(i+1,1)\big) \\
&\geq \bar{\mu}\big(f(i+1,0) + f(i,1)\big) + \mu(1)\big(f(i+2,0) - f(i+1,1)\big) \\
&= \bar{\mu}f(i+1,0) + \big(\bar{\mu} - \mu(1)\big)f(i,1) + \mu(1)\big(f(i+2,0) - f(i+1,1) + f(i,1)\big) \\
&\geq \bar{\mu}f(i+1,0) + \big(\bar{\mu} - \mu(1)\big)f(i,1) + \mu(1)f(i+1,0),
\end{aligned}$$

since f is Supermod(i,j), respectively SuperC(i,j), and ② holds because:

$$\begin{aligned}
&\bar{\mu}f(i,0) + \mu(1)\big(f(i+1,0) + K^{\text{off}}\big) + \big(\bar{\mu} - \mu(1)\big)f(i+1,1) \\
&= \big(\bar{\mu} - \mu(1)\big)\big(f(i,0) + f(i+1,1)\big) + \mu(1)\big(f(i+1,0) + K^{\text{off}} + f(i,0)\big) \\
&\geq \big(\bar{\mu} - \mu(1)\big)\big(f(i+1,0) + f(i,1)\big) + \mu(1)\big(f(i+1,0) + K^{\text{off}} + f(i,0)\big) \\
&= \bar{\mu}f(i+1,0) + \big(\bar{\mu} - \mu(1)\big)f(i,1) + \mu(1)\big(f(i,0) + K^{\text{off}}\big),
\end{aligned}$$

since f is Supermod(i,j).

3) Suppose that f is SuperC(i,j) and Supermod(i,j), then we show that $T_{\text{dep}}f$ is SuperC(i,j). For $j > 0$:

$$\begin{aligned}
&T_{\text{dep}}f(i,j+1) + T_{\text{dep}}f(i+2,j) \\
&= \big(\bar{\mu} - \mu(j+1)\big)f(i,j+1) + \big(\bar{\mu} - \mu(j)\big)f(i+2,j) \\
&\qquad + \min\begin{cases} ① \ \mu(j+1)f(i+1,j) + \mu(j)f(i+3,j-1) \\ ② \ \mu(j+1)f(i+1,j) + \mu(j)\big(f(i+2,j-1) + K^{\text{off}}\big) \\ ③ \ \mu(j+1)\big(f(i,j) + K^{\text{off}}\big) + \mu(j)f(i+3,j-1) \\ ④ \ \mu(j+1)\big(f(i,j) + K^{\text{off}}\big) + \mu(j)\big(f(i+2,j-1) + K^{\text{off}}\big) \end{cases} \\
&\geq \big(\bar{\mu} - \mu(j)\big)f(i+1,j) + \big(\bar{\mu} - \mu(j+1)\big)f(i+1,j+1) \\
&\qquad + \mu(j)\min\{f(i+2,j-1),\ f(i+1,j-1) + K^{\text{off}}\} + \mu(j+1)\min\{f(i+2,j),\ f(i+1,j) + K^{\text{off}}\} \\
&= T_{\text{dep}}f(i+1,j) + T_{\text{dep}}f(i+1,j+1).
\end{aligned}$$

Here ① holds because:

$$\begin{aligned}
&\big(\bar{\mu} - \mu(j+1)\big)f(i,j+1) + \big(\bar{\mu} - \mu(j)\big)f(i+2,j) + \mu(j+1)f(i+1,j) + \mu(j)f(i+3,j-1) \\
&= \big(\bar{\mu} - \mu(j+1)\big)\big(f(i,j+1) + f(i+2,j)\big) + \mu(j)\big(f(i+1,j) + f(i+3,j-1)\big) \\
&\qquad\qquad + \big(\mu(j+1) - \mu(j)\big)\big(f(i+1,j) + f(i+2,j)\big) \\
&\geq \big(\bar{\mu} - \mu(j+1)\big)\big(f(i+1,j) + f(i+1,j+1)\big) + \mu(j)\big(f(i+2,j) + f(i+2,j-1)\big) \\
&\qquad\qquad + \big(\mu(j+1) - \mu(j)\big)\big(f(i+1,j) + f(i+2,j)\big) \\
&= \big(\bar{\mu} - \mu(j)\big)f(i+1,j) + \big(\bar{\mu} - \mu(j+1)\big)f(i+1,j+1) + \mu(j+1)f(i+2,j) + \mu(j)f(i+2,j-1),
\end{aligned}$$

since f is SuperC(i,j).
And ② holds because:

$$
\begin{aligned}
&\big(\bar{\mu}-\mu(j+1)\big)f(i,j+1)+\big(\bar{\mu}-\mu(j)\big)f(i+2,j)+\mu(j+1)f(i+1,j)+\mu(j)\big(f(i+2,j-1)+K^{\text{off}}\big)\\
&=\big(\bar{\mu}-\mu(j+1)\big)\big(f(i,j+1)+f(i+2,j)\big)+\mu(j+1)f(i+1,j)+\mu(j)\big(f(i+2,j-1)+K^{\text{off}}\big)\\
&\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)f(i+2,j)\\
&\geq\big(\bar{\mu}-\mu(j+1)\big)\big(f(i+1,j)+f(i+1,j+1)\big)+\mu(j+1)f(i+1,j)+\mu(j)\big(f(i+2,j-1)+K^{\text{off}}\big)\\
&\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)f(i+2,j)\\
&=\big(\bar{\mu}-\mu(j)\big)f(i+1,j)+\big(\bar{\mu}-\mu(j+1)\big)f(i+1,j+1)+\mu(j)\big(f(i+1,j)+K^{\text{off}}\big)+\mu(j)f(i+2,j-1)\\
&\qquad\qquad+\big(\mu(j+1)-\mu(j)\big)f(i+2,j),
\end{aligned}
$$

since f is SuperC(i,j), and since by (7.A.1):

$$\mu(j)\big(f(i+1,j)+K^{\text{off}}\big)+(\mu(j+1)-\mu(j))(f(i+2,j))\geq\mu(j+1)\min\{f(i+2,j),f(i+1,j)+K^{\text{off}}\}.$$

For ③ first note that

$$
\begin{aligned}
f(i,j)+f(i+3,j-1)&=f(i,j)+f(i+3,j-1)+f(i+1,j)-f(i+1,j)\\
&\geq f(i,j)+f(i+2,j)+f(i+2,j-1)-f(i+1,j)\\
&\geq f(i+1,j-1)+f(i+1,j)+f(i+2,j)-f(i+1,j)\\
&=f(i+1,j-1)+f(i+2,j),
\end{aligned}
\tag{7.A.3}
$$

since f is SuperC(i,j) (used twice). Now ③ holds because:

$$
\begin{aligned}
&\big(\bar{\mu}-\mu(j+1)\big)f(i,j+1)+\big(\bar{\mu}-\mu(j)\big)f(i+2,j)+\mu(j+1)\big(f(i,j)+K^{\text{off}}\big)+\mu(j)f(i+3,j-1)\\
&=\big(\bar{\mu}-\mu(j)\big)\big(f(i,j+1)+f(i+2,j)\big)+\mu(j)\big(f(i,j)+K^{\text{off}}+f(i+3,j-1)\big)\\
&\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}-f(i,j+1)\big)\\
&\geq\big(\bar{\mu}-\mu(j)\big)\big(f(i+1,j)+f(i+1,j+1)\big)+\mu(j)\big(f(i+1,j-1)+f(i+2,j)+K^{\text{off}}\big)\\
&\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}-f(i,j+1)\big)\\
&=\big(\bar{\mu}-\mu(j)\big)f(i+1,j)+\big(\bar{\mu}-\mu(j+1)\big)f(i+1,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)\\
&\qquad+\mu(j)f(i+2,j)+\big(\mu(j+1)-\mu(j)\big)\big(f(i,j)+K^{\text{off}}+f(i+1,j+1)-f(i,j+1)\big)\\
&\geq\big(\bar{\mu}-\mu(j)\big)f(i+1,j)+\big(\bar{\mu}-\mu(j+1)\big)f(i+1,j+1)+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)\\
&\qquad+\mu(j)f(i+2,j)+\big(\mu(j+1)-\mu(j)\big)\big(f(i+1,j)+K^{\text{off}}\big),
\end{aligned}
$$

where the first inequality holds since f is SuperC(i,j) (part with $\bar{\mu}-\mu(j)$) and by (7.A.3) (part with $\mu(j)$), and the second inequality holds since f is Supermod(i,j), and using that by (7.A.1):

$$\mu(j)f(i+2,j)+\big(\mu(j+1)-\mu(j)\big)\big(f(i+1,j)+K^{\text{off}}\big)\geq\mu(j+1)\min\{f(i+2,j),\ f(i+1,j)+K^{\text{off}}\}.$$

Finally, ④ holds because:

$$
\begin{aligned}
&\big(\bar\mu-\mu(j+1)\big)f(i,j+1)+\big(\bar\mu-\mu(j)\big)f(i+2,j)+\mu(j+1)\big(f(i,j)+K^{\text{off}}\big)\\
&\qquad+\mu(j)\big(f(i+2,j-1)+K^{\text{off}}\big)\\
&=\big(\bar\mu-\mu(j+1)\big)\big(f(i,j+1)+f(i+2,j)\big)+\mu(j+1)\big(f(i,j)+K^{\text{off}}+f(i+2,j-1)+K^{\text{off}}\big)\\
&\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+2,j-1)-K^{\text{off}}\big)\\
&\geq\big(\bar\mu-\mu(j+1)\big)\big(f(i+1,j+1)+f(i+1,j)\big)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}+f(i+1,j-1)+K^{\text{off}}\big)\\
&\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+2,j-1)-K^{\text{off}}\big)\\
&=\big(\bar\mu-\mu(j+1)\big)f(i+1,j+1)+\big(\bar\mu-\mu(j)\big)f(i+1,j)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&\qquad+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big)\\
&\qquad+\big(\mu(j+1)-\mu(j)\big)\big(f(i+2,j)-f(i+2,j-1)-f(i+1,j)+f(i+1,j-1)\big)\\
&\geq\big(\bar\mu-\mu(j+1)\big)f(i+1,j+1)+\big(\bar\mu-\mu(j)\big)f(i+1,j)+\mu(j+1)\big(f(i+1,j)+K^{\text{off}}\big)\\
&\qquad+\mu(j)\big(f(i+1,j-1)+K^{\text{off}}\big),
\end{aligned}
$$

since f is SuperC(i,j), respectively Supermod(i,j).

Furthermore, for $j=0$:

$$
\begin{aligned}
&T_{\text{dep}}f(i,1)+T_{\text{dep}}f(i+2,0)\\
&=\big(\bar\mu-\mu(1)\big)f(i,1)+\mu(1)\min\begin{Bmatrix}① \ f(i+1,0)\\ ② \ f(i,0)+K^{\text{off}}\end{Bmatrix}+\bar\mu f(i+2,0)\\
&\geq\bar\mu f(i+1,0)+\big(\bar\mu-\mu(1)\big)f(i+1,1)+\mu(1)\min\{f(i+2,0),f(i+1,0)+K^{\text{off}}\}\\
&=T_{\text{dep}}f(i+1,0)+T_{\text{dep}}f(i+1,1),
\end{aligned}
$$

where ① holds because:

$$
\begin{aligned}
&\big(\bar\mu-\mu(1)\big)f(i,1)+\mu(1)f(i+1,0)+\bar\mu f(i+2,0)\\
&=\big(\bar\mu-\mu(1)\big)\big(f(i,1)+f(i+2,0)\big)+\mu(1)\big(f(i+1,0)+f(i+2,0)\big)\\
&\geq\big(\bar\mu-\mu(1)\big)\big(f(i+1,0)+f(i+1,1)\big)+\mu(1)\big(f(i+1,0)+f(i+2,0)\big)\\
&=\bar\mu f(i+1,0)+\big(\bar\mu-\mu(1)\big)f(i+1,1)+\mu(1)f(i+2,0),
\end{aligned}
$$

since f is SuperC(i,j), and ② holds because:

$$
\begin{aligned}
&\big(\bar\mu-\mu(1)\big)f(i,1)+\mu(1)\big(f(i,0)+K^{\text{off}}\big)+\bar\mu f(i+2,0)\\
&=\bar\mu\big(f(i,1)+f(i+2,0)\big)+\mu(1)\big(f(i,0)+K^{\text{off}}-f(i,1)\big)\\
&\geq\bar\mu\big(f(i+1,0)+f(i+1,1)\big)+\mu(1)\big(f(i,0)+K^{\text{off}}-f(i,1)\big)\\
&=\bar\mu f(i+1,0)+\big(\bar\mu-\mu(1)\big)f(i+1,1)+\mu(1)\big(f(i,0)+K^{\text{off}}-f(i,1)+f(i+1,1)\big)\\
&\geq\bar\mu f(i+1,0)+\big(\bar\mu-\mu(1)\big)f(i+1,1)+\mu(1)\big(f(i+1,0)+K^{\text{off}}\big),
\end{aligned}
$$

since f is SuperC(i,j), respectively Supermod(i,j). □

7.A.3 Proof of Lemma 7.3.3

PROOF. Suppose that f is BFOD(i, K^{on}), then we show that $\frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda T_{\text{arr}} + T_{\text{dep}}\Big) f$ is BFOD(i, K^{on}) as well. For $i > 0$, respectively $i = 0$:

$$\begin{aligned}
&\frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda T_{\text{arr}} f(i+1,j) + T_{\text{dep}} f(i+1,j)\Big) + K^{\text{on}} \\
&= \frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda f(i,j+1) + \mu(j)\min\{f(i+2,j-1),\ f(i+1,j-1)+K^{\text{off}}\} \\
&\qquad\qquad + (\bar{\mu}-\mu(j)) f(i+1,j) + (\lambda+\bar{\mu}+\alpha)K^{\text{on}}\Big) \\
&= \frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda\left(f(i,j+1)+K^{\text{on}}\right) + \mu(j)\min\{f(i+2,j-1)+K^{\text{on}},\ f(i+1,j-1)+K^{\text{off}}+K^{\text{on}}\} \\
&\qquad\qquad + (\bar{\mu}-\mu(j))\left(f(i+1,j)+K^{\text{on}}\right) + \alpha K^{\text{on}}\Big) \\
&\geq \frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda\left\{\begin{array}{ll} i>0: & f(i-1,j+1) \\ i=0: & f(i,j+1+K^{\text{on}}) \end{array}\right\} + \mu(j)\min\{f(i+1,j-1),\ f(i,j-1)+K^{\text{off}}\} \\
&\qquad\qquad + (\bar{\mu}-\mu(j)) f(i,j)\Big) + \frac{\alpha}{\lambda+\bar{\mu}+\alpha}K^{\text{on}} \\
&\geq \frac{1}{\lambda+\bar{\mu}+\alpha}\Big(\lambda T_{\text{arr}} f(i,j) + T_{\text{dep}} f(i,j)\Big),
\end{aligned}$$

since f is BFOD(i, K^{on}) and $\alpha/(\lambda+\bar{\mu}+\alpha)K^{\text{on}} \geq 0$. □

7.A.4 Proof of Theorem 7.3.6

PROOF. Define, for all $i \geq 0$, $j > 0$, and $n \geq 0$:

$$\begin{aligned}
w^{(n)}(\text{on}, i, j) &:= V_n(i+1, j-1), \\
w^{(n)}(\text{off}, i, j) &:= V_n(i, j-1) + K^{\text{off}}.
\end{aligned}$$

Hence

$$T_{\text{dep}} V_n(i,j) = \mu(j) \min_{u\in\{on,off\}} w^{(n)}(u,i,j) + \big(\bar{\mu}-\mu(j)\big) V_n(i,j).$$

i) Define $\Delta w_i^{(n)}(u,i,j) := w^{(n)}(u,i+1,j) - w^{(n)}(u,i,j)$. Then

$$\begin{aligned}
&\Delta w_i^{(n)}(\text{on}, i, j+1) - \Delta w_i^{(n)}(\text{off}, i, j+1) \\
&= w^{(n)}(\text{on}, i+1, j+1) - w^{(n)}(\text{on}, i, j+1) - w^{(n)}(\text{off}, i+1, j+1) + w^{(n)}(\text{off}, i, j+1) \\
&= V_n(i+2, j) - V_n(i+1, j) - V_n(i+1, j) + V_n(i, j) \geq 0
\end{aligned}$$

as, by Corollary 7.3.5, V_n is Conv(i). Hence, if the optimal action in state (i, j) is to keep the server on upon a service completion, then this is the optimal action as well in state $(i-1, j)$. Also, if the optimal action in state (i, j) is to turn the server off upon a service completion, then this is the optimal action as well in state $(i+1, j)$.

ii) Define $\Delta w_j^{(n)}(u,i,j) := w^{(n)}(u,i,j+1) - w^{(n)}(u,i,j)$. Then

$$\begin{aligned}
&\Delta w_j^{(n)}(\text{on}, i, j+1) - \Delta w_j^{(n)}(\text{off}, i, j+1) \\
&= w^{(n)}(\text{on}, i, j+2) - w^{(n)}(\text{on}, i, j+1) - w^{(n)}(\text{off}, i, j+1) + w^{(n)}(\text{off}, i, j+2) \\
&= V_n(i+1, j+1) - V_n(i+1, j) - V_n(i, j+1) + V_n(i, j) \geq 0
\end{aligned}$$

as, by Corollary 7.3.5, V_n is Supermod(i, j). Hence, if the optimal action in state (i, j) is to keep the server on upon a service completion, then this is the optimal action as well in state $(i, j-1)$. Also, if the optimal action in state (i, j) is to turn the server off upon a service completion, then this is the optimal action as well in state $(i, j+1)$.

iii) Define $\Delta w^{(n)}_{j-i}(u, i, j) := w^{(n)}(u, i-1, j+1) - w^{(n)}(u, i, j)$. Then

$$\begin{aligned}
&\Delta w^{(n)}_{j-i}(\text{on}, i, j+1) - \Delta w^{(n)}_{j-i}(\text{off}, i, j+1) \\
&= w^{(n)}(\text{on}, i, j+2) - w^{(n)}(\text{on}, i, j+1) - w^{(n)}(\text{off}, i, j+1) + w^{(n)}(\text{off}, i, j+2) \\
&= V_n(i, j+1) - V_n(i+1, j) - V_n(i-1, j+1) + V_n(i, j) \geq 0
\end{aligned}$$

as, by Corollary 7.3.5, V_n is SuperC(i, j). Hence, if the optimal action in state (i, j) is to keep the server on upon a service completion, then this is the optimal action as well in state $(i-1, j+1)$. Also, if the optimal action in state (i, j) is to turn the server off upon a service completion, then this is the optimal action as well in state $(i+1, j-1)$.

This proves that the described optimal policy structure holds for an arbitrary $n \geq 0$. As $V_n \to V$ as $n \to \infty$, it follows that V satisfies properties (7.3.1)–(7.3.4) (cf. Lemma 7.3.1), and hence satisfies the above results as well. Then, from (7.2.1) (for $n \to \infty$) it follows that this is also the optimal policy structure in the limit as $n \to \infty$. $\square$

8

Optimal control of a head-of-line processor sharing model

Motivated by a workload control setting, we study a model where two types of customers are served by a single server according to the head-of-line processor sharing discipline. Regular customers and opportunity customers are arriving to the system according to two independent Poisson processes, each requiring an exponentially distributed service time. The regular customers will queue, incurring some holding costs. On contrary, an opportunity customer has to be taken into service directly, or is lost otherwise. There can be at most one opportunity customer in the system. The server can work on both one regular and one opportunity customer at the same time, where one can decide on how the server speed is split out. Moreover, one can decide whether to accept or reject an opportunity customer, incurring penalty costs for the latter. In this way, one has partial control about the workload in the system. We formulate the model as a Markov decision problem. We prove that the optimal policy, minimizing the expected discounted long-run cost, has a monotone structure in the number of regular customers. That is, the optimal policy for accepting an opportunity customer is described be a threshold, and the fraction of the server's attention devoted to the opportunity customer is a monotone decreasing function.

8.1 Introduction

Motived by a workload control setting, we study a model where two types of customers are served by a single server. These two types are *regular customers* and *opportunity customers*. The regular customers are willing to wait before taken into service, whereas the opportunity customers have to be taken into service directly, or are lost otherwise. We allow at most one such a customer in the system. When, for example, the workload of the regular customers is low, an opportunity customer provides the opportunity for some extra revenue. Hence, we have control over the workload by deciding whether to take such a customer into service. For this system, we derive the optimal control policy in this chapter.

We model this problem as a single server queueing model servicing the two types of customers. The regular customers form a queue upon arrival to the system. The server can work on both one regular and one opportunity customer at the same time. The (total) service rate of the server is given and fixed, but one can adjust how it is split out among these two customers. This is known as the head-of-line processor sharing discipline, with adjustable service rate. Moreover, one can decide whether to accept or reject an opportunity customer, incurring penalty costs for the latter. Hence, we have to balance the server speed and acceptance decision. If the service rate of the opportunity customer is set too low, we face the risk of the next opportunity customer already arriving before service completion (having to reject it). On the other hand, a higher server speed will let all regular customers queue for a longer time, incurring longer and higher holding costs.

We formulate the model as Markov decision problem. Based on the number of regular customers in the system, a decision has to be taken whether to accept or reject an arriving opportunity customer. Moreover, one has to decide on the server speed for an opportunity customer, if such a customer is present in the system. Using event-based dynamic programming, we prove that the value function is increasing and multimodular. From that, the structure of the optimal policy follows. This optimal policy is a threshold policy for admitting an opportunity customer, and a monotone decreasing function for the server rate assigned to the opportunity customer.

The optimal control of a head-of-line processor sharing is to the best of our knowledge an open problem. We mention the following related results. Konheim et al. [113] give a complete analysis of a system with two parallel queueing lines, served by a single server, but assume that each is served with half of the service rate. Fayolle et al. [76] study a more general framework than [113], where the fraction of the attention to each queue is flexible. However, they assume that from a given number of customers on, the service rates are independent. Wasserman and Bambos [202] study the dynamic allocation of a single server to parallel queues with finite-capacity buffers, characterizing the allocation policy that stochastically minimizes the number of customers lost due to buffer overflows. A similar problem is studied in Towsley et al. [184]. Stidham [171] focuses on the optimal control of admission to queueing systems, and uses dynamic-programming to show that an optimal control is monotonic or characterized by one or more critical levels. Weber and Stidham [203] study the optimal control of service rates in a network of queues. The optimal control of limited processor sharing is studied in Van der Weij et al. [188]. They dynamically adjust the number of servers in a queue with processor sharing, where every customer in service receives a proportional fraction of the processing time. They use the same kind of techniques and derive the same kind of structural results as we do, namely monotonicity properties and optimal dynamic policies using dynamic programming.

The outline of this chapter is as follows. We start by introducing the model and notation in Section 8.2. We describe the problem in more detail and give the dynamic programming formulation. In Section 8.3 we present the optimal policy structure, and show some examples. We derive the steady-state probability distribution in Section 8.4. In Section 8.5 we consider two model generalizations, which we show to fit in the same framework. Combining them leads to a general two queue head-of-line processor sharing model. We conclude in Section 8.6. All proofs are in Appendix 8.A, and the details of the derivation of the steady-state probability distribution are given in Appendix 8.B. This chapter is based on [191].

8.2 Model and notation

In this section we describe the model in more detail and introduce the notation used. We then formulate the model as a Markov decision problem (MDP). We also present the value function and the event operators.

8.2.1 Problem description

We consider the following queuing model, with two types of customers served by a single server. Regular customers arrive according to a Poisson process with rate λ_{reg}, and form a queue; opportunity customers arrive according to a Poisson process with rate λ_{opp}. An opportunity customer has to be taken into service directly, or is lost otherwise, at penalty cost $C_{opp} \geq 0$. Holding costs are charged for both regular and opportunity customers in the system: $\tilde{h}_{reg}(\cdot)$ and $\tilde{h}_{opp}(\cdot)$ respectively, per time unit as a function of the number of regular, respectively opportunity customers in the system. We assume $\tilde{h}_{reg}(\cdot)$ and $\tilde{h}_{opp}(\cdot)$ to be non-negative, increasing and convex. By incorporating holding costs for an opportunity customer in service, we try to prevent the model from choosing a very low service rate for this customer.

A single server servers both queues, applying the head-of-line processor sharing strategy with adjustable weights. That is, the server can simultaneously serve an opportunity customer and the first in line regular customer. The total service rate of the server is fixed, say $\bar{\mu}$, but it can be decided how this is split out between both customer types. Denote by $\mu \in [0,1]$ the fraction of the rate dedicated to the opportunity customer. Then, with rate $0 \leq \mu\bar{\mu} \leq \bar{\mu}$ the opportunity customer is served, leaving rate $(1-\mu)\bar{\mu}$ for the regular customer. Here μ is a decision variable, where we assume $\mu = 0$ when there are no opportunity customers in the system. For generality, we charge costs $c(\mu)$ when rate μ is chosen, assuming $c(0) = \min_{\mu\in[0,1]} c(\mu)$. The service times of both opportunity and regular customers are exponentially distributed with mean 1. We assume all processes to be mutually independent. Furthermore, we require $\lambda_{reg} < \bar{\mu}$, hence a policy resulting in a stable system always exists. We derive the optimal policy structure that minimizes the expected long-run discounted cost, with continuous discount factor $\alpha > 0$.

8.2.2 Dynamic programming formulation

Denote by $x \in \{0,1\}$ the number of opportunity customers in the system, and by $y \in \mathbb{N}_0$ the number of regular customers in the system, hence

$$(x,y) \in \mathcal{S} = \{(x,y) \mid x \in \{0,1\},\ y \in \mathbb{N}_0\},$$

denoting by $\mathcal{S}$ the state space.

As the interarrival times of customer arrivals as well as service times are independent exponentially distributed random variables, we can apply uniformization (cf. [131]) to convert the semi-Markov decision problem into an equivalent Markov decision problem (MDP). The existence of a stationary optimal policy is guaranteed by [155, Theorem 11.5.3].

Let the value function $V_n : \mathcal{S} \mapsto \mathbb{R}^+$ be the minimum expected discounted costs when there are n events (customer arrivals or (potential) service completions) left, starting in state $(x, y) \in \mathcal{S}$. It is given by:

$$V_{n+1}(x,y) = T_{\text{costs}}\Big(T_{\text{unif}}\big(T_{CA(1)}V_n(x,y),\ T_{A(2)}V_n(x,y),\ \tilde{T}_{CTD(1)}V_n(x,y)\big)\Big), \tag{8.2.1}$$

starting with $V_0 \equiv 0$. Below, we define the operators the value function consists of. Let $\nu = \lambda_{opp} + \lambda_{reg} + \bar{\mu} + \alpha$ be the uniformization rate.

The costs operator T_{costs} is defined by

$$T_{\text{costs}} f(x,y) = h_{opp}(x) + h_{reg}(y) + f(x,y),$$

where $h_{opp}(x) = \tilde{h}_{opp}(x)/\nu$ and $h_{reg} = \tilde{h}_{reg}(x)/\nu$ are the holding costs per time unit $1/\nu$. These are non-negative, increasing, and convex as well.

The uniformization operator T_{unif} for this problem is defined by

$$T_{\text{unif}}(f_1, f_2, f_3)(x,y) = \frac{1}{\nu}\Big(\lambda_{opp} f_1(x,y) + \lambda_{reg} f_2(x,y) + \bar{\mu} f_3(x,y)\Big).$$

The operator $T_{CA(1)}$ models the (controlled) arrivals of opportunity customers, and is defined by

$$T_{CA(1)} f(x,y) = \begin{cases} \min\{V_n(x+1,y), V_n(x,y) + C_{opp}\} & \text{if } x = 0, \\ V_n(x,y) + C_{opp}\} & \text{otherwise.} \end{cases} \tag{8.2.2}$$

Upon arrival of an opportunity customer, one has to decide to either accept it (moving the process to state $(x+1, y)$) or reject it (at costs C_{opp}), with the restriction that there can maximally be one opportunity customer in the system at a time.

The operator $T_{A(2)}$ models the (uncontrolled) arrivals of regular customers, and is defined by

$$T_{A(2)} f(x,y) = V_n(x, y+1). \tag{8.2.3}$$

The operator $\tilde{T}_{CTD(1)}$ models the (potential) service completions, and is defined by

$$\tilde{T}_{CTD(1)} f(x,y) = \min_{\mu \in [0,1]} \Big\{ c(\mu) + \mu V_n((x-1)^+, y) + (1-\mu) V_n(x, (y-1)^+) \Big\}, \tag{8.2.4}$$

where $x^+ = \max\{x, 0\}$. Here, μ is a decision variable, deciding which part of the server speed is allocated to the opportunity customer. Hence, with rate μ, x decreases by one (when x is positive). A fraction $1-\mu$ is left to the regular customers, hence with this rate y decreases by one (when y is positive). Costs $c(\mu)$ are incurred when μ is chosen. Recall that we assume $\mu = 0$ when $x = 0$, where $c(0) = \min_{\mu \in [0,1]} c(\mu)$. If $y = 0$, fictitious transitions are made to a state itself, hence assuring that the rate at which $\tilde{T}_{CTD(1)}$ occurs is always equal to $\bar{\mu}$.

The operator $\tilde{T}_{CTD(1)}$ is almost equal to the operator that is used for modeling the service completions in a two-stage tandem queue, where the total service capacity is split out between the two queues, such that μ is a decision variable. This operator is defined as $T_{CTD(1)}$ in [116, Definition 5.4]. Remarkably, this operator coincides with our operator $\tilde{T}_{CTD(1)}$, when a coordinate transformation is y is made. Hence, known results for $T_{CTD(1)}$ can easily be adapted for $\tilde{T}_{CTD(1)}$. We use this in the proofs of the propagation results for $\tilde{T}_{CTD(1)}$.

8.3 Structural results

In this section we prove our main result: the structure of the optimal policy. For this, we prove that the value function V_n is increasing and multimodular by showing that each of the operators in V_n preserve these properties. From this we derive the structure of the optimal policy, which is a threshold policy for accepting an opportunity customer, and a monotone deceasing function for the optimal server speed dedicated to the opportunity customer. We illustrate the policy by examples. All proofs are given in Appendix 8.A.

8.3.1 Properties of operators and value function

Consider, as introduced in Section 2.3.2, the following properties of a function f, defined for all x such that the states appearing in the right-hand and left-hand side of the inequalities exist in $\mathcal{S}$:

$$\begin{aligned}
\text{Incr}(x): &\quad f(x+1,y) \geq f(x,y), && (8.3.1)\\
\text{Incr}(y): &\quad f(x,y+1) \geq f(x,y), && (8.3.2)\\
\text{Conv}(x): &\quad f(x,y)+f(x+2,y) \geq 2f(x+1,y), && (8.3.3)\\
\text{Conv}(y): &\quad f(x,y)+f(x,y+2) \geq 2f(x,y+1), && (8.3.4)\\
\text{Supermod}: &\quad f(x,y)+f(x+1,y+1) \geq f(x+1,y)+f(x,y+1), && (8.3.5)\\
\text{SuperC}(x,y): &\quad f(x+2,y)+f(x,y+1) \geq f(x+1,y)+f(x+1,y+1), && (8.3.6)\\
\text{SuperC}(y,x): &\quad f(x,y+2)+f(x+1,y) \geq f(x,y+1)+f(x+1,y+1), && (8.3.7)\\
\text{Incr}: &\quad \text{Incr}(x)\cap\text{Incr}(y), && (8.3.8)\\
\text{MM}: &\quad \text{Supermod}\cap\text{SuperC}(x,y)\cap\text{SuperC}(y,x). && (8.3.9)
\end{aligned}$$

LEMMA 8.3.1. *The operators $T_{CA(1)}$, $T_{A(2)}$, $\tilde{T}_{CTD(1)}$, T_{unif}, and T_{costs} preserve Incr and MM.*

That is, if some function f is Incr and MM, then Tf is Incr and MM as well, where T is one of the mentioned operators. By induction on n, and using that $V_0 \equiv 0$, the next result immediately follows.

THEOREM 8.3.2. *V_n is Incr and MM for all $n \geq 0$.*

We use this result to derive the structure of the optimal policy.

8.3.2 Structure of optimal policy

The next theorem states the optimal policy structure, which minimizes the expected long-run discounted costs.

THEOREM 8.3.3. *a) The optimal policy for admitting an opportunity customer is a threshold policy. That is, there exist a threshold, say $T \in \mathbb{N}_0$, such that the optimal decision is to accept the opportunity customer if $y \leq T$, and to reject it otherwise.*
b) The optimal server speed dedicated to the opportunity customer is a monotone deceasing function in y.

Here, the decreasingness in part b) is understood to be non-strict (i.e., the server speed is non-increasing). The optimal policy structure is in line with our intuition. When the workload of regular customers is low, one is more likely to accept an opportunity customer. Also, the more regular customers there are in the system, the larger the fraction of the server's attention is assigned to these customers. As a consequence, the server speed for the opportunity customer is decreasing in y.

REMARK 8.3.4. In the case that no holding costs are charged for an opportunity customer in service, i.e. when $h_{opp}(1) = 0$, or in case $h_{opp}(0) = h_{opp}(1)$, the opportunity customer is always accepted. That is, $T = \infty$. However, it might receive no service ($\mu = 0$) when y is large. It might even be the case that when taken into service, μ is positive, but if it happens that the number of regular customers increases, the service rate may decrease to zero.

When $c(\mu) \equiv 0$, the optimal μ is always either 0 or 1. This follows directly from the fact that in this case a linear function is minimized in (8.2.4). Hence, the optimal policy can be described by a single threshold.

COROLLARY 8.3.5. *If $c(\mu) \equiv 0$, then the optimal policy for the server speed dedicated to the opportunity customer is a threshold policy. That is, there exist a threshold, say $M \in \mathbb{N}_0$, such that when $x = 1$, the optimal fraction of the service rate dedicated to the opportunity customer is 1 if $y \le M$, and 0 otherwise.*

8.3.3 Examples

We consider two examples, one for which $c(\mu) \equiv 0$, and one for which it is positive for some values of μ.

Example 8.3.1. Consider an example with the following parameters: $\lambda_{reg} = 3$, $\lambda_{opp} = 1$, $\bar{\mu} = 10$, $\alpha \downarrow 0$, $C_{opp} = 8$, $c(\mu) = 0$, $h_{opp}(x) = x$ and

$$h_{reg}(y) = \begin{cases} 0.05\, y^2 & \text{if } y < 20; \\ 100\, y & \text{otherwise.} \end{cases}$$

Hence, the holding costs are more than linearly increasing. Moreover, for computational purposes, we can truncate the state space for y large, as the optimal policy avoids getting to states for which $y \ge 20$. The optimal policy for accepting opportunity customers is given by:

$x \backslash y$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Here a 1 indicates acceptance, and a 0 indicates that the customer is reject under the optimal policy. So, the optimal policy is indeed a threshold policy for accepting an opportunity customer. The threshold is $T = 6$, where the opportunity customer is accepted when $y \le T$, and rejecting otherwise. The optimal fraction $\mu \in [0, 1]$ of the server speed dedicated to the opportunity customer is given by:

$x \backslash y$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

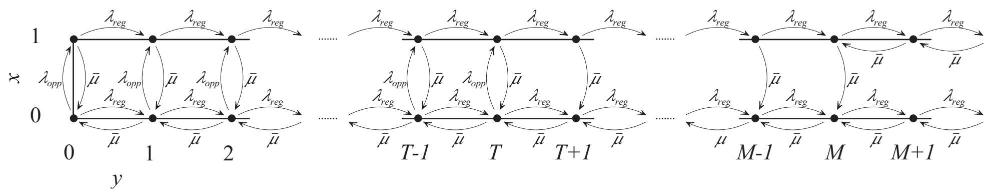


Figure 8.1: Transition rates when executing the optimal policy in case $c(\mu) \equiv 0$. *Figure for* $T < M$.

In line with Corollary 8.3.5, the opportunity customers either gets full attention of the server, or no attention at all. The threshold for this is $M = 10$.

Example 8.3.2. We use the same parameter values as in Example 8.3.1, however, for $c(\mu)$ we now take:

$$c(\mu) = \begin{cases} 0 & \text{if } 0 \le \mu < 0.25; \\ 0.5 & \text{if } 0.25 \le \mu < 0.50; \\ 1 & \text{if } 0.50 \le \mu < 0.75; \\ 1.5 & \text{if } 0.75 \le \mu \le 1, \end{cases}$$

which clearly satisfies the assumption $c(0) = \min_{\mu \in [0,1]} c(\mu)$. The threshold for accepting opportunity customers now is $T = 5$. The optimal fraction $\mu \in [0, 1]$ of the server speed dedicated to the opportunity customer, is given by:

$x \backslash y$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0.75	0.25	0.25	0.25	0.25	0	0	0

So, the server speed for the opportunity customer is indeed monotone decreasing.

8.4 Steady-state probability distribution

Given the optimal policy structure, we now derive a closed-form expression for the steady-state probability distribution (i.e., for the equilibrium probabilities), for the case that $c(\mu) \equiv 0$. Hence, μ is either 0 or 1. By deriving the steady-state probabilities, the average costs of a policy can easily and quickly be calculated. In this way, one can also easily and quickly calculate the optimal policy parameters. We state the results in this section, where the details of the derivation are given in Appendix 8.B.

The transition rates when executing the optimal policy are indicated in Figure 8.1. Denote by $p(x, y)$ the stationary probability of being in state (x, y). When $T < M$, these are given by:

$$p(x,y) = \begin{cases} c_1 v_1(x) + c_2 v_2(x) \alpha_2^y + c_3 v_3(x) \alpha_3^y & \text{for } 0 \le y < T; \\ d_1 w_1(x) + d_2 w_2(x) \left(\frac{\lambda_{reg}}{\mu}\right)^{y-T} + d_3 w_3(x) \left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{y-T} & \text{for } T \le y < M; \\ q(x) \left(\frac{\lambda_{reg}}{\mu}\right)^{y-M} & \text{for } y \ge M. \end{cases}$$

The constants α_2, α_3, $v_i(x)$, and $w_i(x)$, for $i = 1, 2, 3$, are given in (8.B.4), (8.B.5), and (8.B.12) respectively. The remaining 8 constants ($q(0)$, $q(1)$, d_1, d_2, d_3, c_1, c_2,

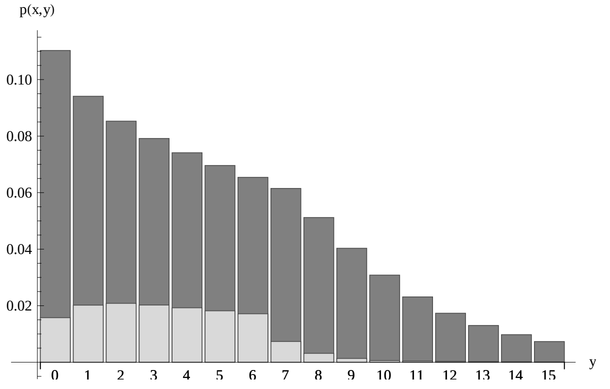


Figure 8.2: Example of the steady-state probabilities $p(x, y)$ (where $p(0, y)$ in dark gray and $p(1, y)$ in light gray) for $T = 6$, $M = 10$, when $\lambda_{opp} = 1$, $\lambda_{reg} = 3$, and $\bar{\mu} = 4$.

and c_3) follow from solving a system of 8 equations, see Appendix 8.B. An example of the steady-state probabilities is given in Figure 8.2. For the cases $T > M$ and $T = M$ similar expressions can be obtained.

Denote the average costs per time unit when executing the (not necessarily optimal) described threshold policy with thresholds T and M, for the case that $c(\mu) \equiv 0$, by $c(T, M)$. Using the steady-state probabilities $p(x, y)$, it can be expressed as:

$$\begin{aligned} c(T,M) = \sum_y \Big(\tilde{h}_{reg}(y) p(0,y) + \big(\tilde{h}_{opp}(1) + \tilde{h}_{reg}(y)\big) p(1,y) \Big) \\ + \lambda_{opp} C_{opp} \Bigg(\sum_{y>T} p(0,y) + \sum_y p(1,y) \Bigg). \end{aligned} \tag{8.4.1}$$

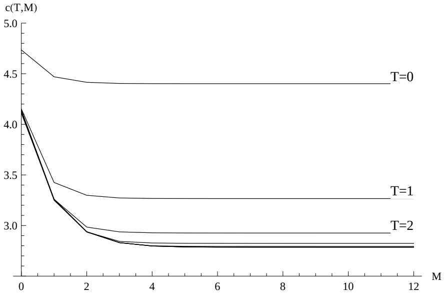
The optimal thresholds, say T^* and M^*, are now given by

$$(T^*, M^*) = \underset{(T,M)}{\operatorname{argmin}}\, c(T, M).$$

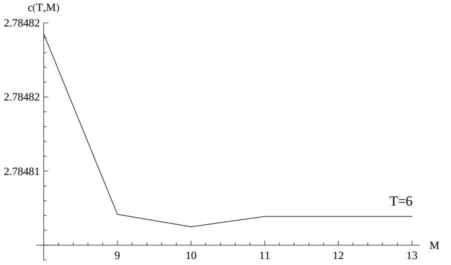
In Figure 8.3, $c(T, M)$ is given for multiple values of T and M for the parameter settings of Example 8.3.1. Indeed, the minimum costs are attained for $T = 6$ and $M = 10$, as we established using value iteration in Example 8.3.1. Note that by the expressions for $p(x, y)$, the resulting infinite geometric sums in (8.4.1) can be calculated exactly.

8.5 Generalized model

We consider two generalizations of the model of Section 8.2. Firstly, we allow opportunity customers to queue as well. Secondly, we allow regular customers to be rejected as well. We show that both generalizations fit in the same framework as the original model. Combining them leads to a general two-queue head-of-line processor sharing model.



(a) Graphs for $T \in \{0, 1, \dots, 7\}$.



(b) Graph for $T = 6$ (note: different scales than in Figure (a)).

Figure 8.3: Average costs per time unit $c(T,M)$ as in (8.4.1) *for Example 8.3.1, where $c(T,M)$ is plotted against M. The costs are minimized for $(T,M) = (6,10)$.*

8.5.1 Queueing of opportunity customers

Instead of having to take an opportunity customer directly into service, we now allow them to queue as well. Here, we assume that both types of customers (regular and opportunity) form separate queues. The server can only work on the first in line customers of both queues.

For this generalization, we have to adapt $T_{CA(1)}$ of (8.2.2) into say $\tilde{T}_{CA(1)}$:

$$\tilde{T}_{CA(1)} f(x,y) = \min\{V_n(x+1,y), V_n(x,y) + C_{opp}\}. \tag{8.5.1}$$

This removes the restriction that there can maximally be one opportunity customer in the system at a time. In V_n of (8.2.1), we replace $T_{CA(1)}$ by $\tilde{T}_{CA(1)}$. The resulting V_n is Incr and MM for all $n \geq 0$, on the state space $\{(x,y) \mid x \in \mathbb{N}_0,\ y \in \mathbb{N}_0\}$.

For this model, the following theorem describes the optimal policy. We encode acceptance of a customer by 1, and rejecting by 0.

THEOREM 8.5.1. *a) The optimal policy for admitting opportunity customers is a state-dependent threshold policy. That is, there exist a switching curve, say $T_{opp}(x)$, such that the optimal decision in state (x,y) is to accept the opportunity customer when $y \leq T_{opp}(x)$, and to reject it otherwise. Moreover, $T_{opp}(x)$ is strictly decreasing in x (until it reaches 0).*
b) The optimal server speed dedicated to an opportunity customer is a function, say $M(x,y)$, which is monotone increasing in x and monotone decreasing in y.

Again, the increasingness and decreasingness in part b) are understood to be non-strict.

8.5.2 Rejecting of regular customers

We now allow regular customers to be rejected as well, at costs $C_{reg} \geq 0$. To model this decision, instead of the operator $T_{A(2)}$ of (8.2.3), we now have controlled arrivals of regular customers:

$$T_{CA(2)} f(x,y) = \min\{V_n(x,y+1), V_n(x,y) + C_{reg}\}.$$

In V_n of (8.2.1), we replace $T_{A(2)}$ by $T_{CA(2)}$. The resulting V_n is Incr and MM for all $n \geq 0$.

Analogously to Theorem 8.5.1, part a), the optimal decision for accepting customers can again be characterized by a state-depended threshold:

THEOREM 8.5.2. *The optimal policy for admitting regular customers is a state-dependent threshold policy. That is, there exist a switching curve, say $T_{reg}(x)$, such that the optimal decision in state (x, y) is to accept the regular customer when $y \leq T_{reg}(x)$, and to reject it otherwise. Moreover, $T_{reg}(x)$ is strictly decreasing in x (until it reaches 0).*

The optimal server speed dedicated to the opportunity customer is still as described in Theorem 8.5.1, part b).

REMARK 8.5.3 (General model). When combining the generalizations of Sections 8.5.1 and 8.5.2, we have a general two queue head-of-line processor sharing model, controlling the allocation of the service rate as well as the acceptance of customers in both queues. For this model, the optimal control policy is as described in Theorems 8.5.1 and 8.5.2.

8.6 Conclusion

We presented a single server head-of-line processor sharing model. For this, we derived the structure of the optimal policy. The results are in line with one's intuition for the control of such a system. We derived the steady-state probability distribution of the number of customers in the system, when executing a threshold policy. This policy structure is optimal for $c(\mu) \equiv 0$. We also discussed a more general model, allowing opportunity customers to queue, and regular customers to be rejected.

Using the steady-state probabilities, the average costs follow. In further research, this can be used in a numerical study, to compare the performance of the optimal policy to simple policies, e.g. a policy that always accepts or always rejects the opportunity customers, or that always gives full attention to the opportunity customer in service. Another interesting questions for further research is whether the structural results will remain to hold when the total service rate is increasing (or decreasing) when the server divides its attention to two customers.

8.A Appendix: Proofs

8.A.1 Proof of Lemma 8.3.1

PROOF. For $T_{CA(1)}$, $T_{A(2)}$, T_{unif}, and T_{costs}, the statements follow from [116, Theorems 7.1 and 7.2] (for the specific case of a two dimensional state space). Note that in [116] $T_{CA(1)}$ is defined as $T_{CA(1)}f(x,y) = \min\{V_n(x+1,y), V_n(x,y) + C_{opp}\}$, however, one can easily incorporate the restriction $x \leq 1$ by replacing $\tilde{h}_{opp}(x)$ in (8.2.1) by

$$\begin{cases} \tilde{h}_{opp}(x) & \text{if } x \leq 1; \\ K\,x & \text{if } x > 1, \end{cases} \tag{8.A.1}$$

with K a large constant (cf. [116, p.57]). Hence, when $x = 1$, the minimum is always attained for $V_n(x,y) + C_{opp}$, preventing the system from moving to some state $(2, y)$.

$\tilde{T}_{CTD(1)}$ is a small variation of $T_{CTD(1)}$ as in [116, Definition 5.4]. It holds that

$$\tilde{T}_{CTD(1)} f(x,y) = T_{CTD(1)} f(x,y-1) \text{ for } y > 0.$$

Hence, for $y > 0$, the statements directly follows from [116, Theorem 7.4]. For $y = 0$ the properties can be proven along the same lines as in the original proves for $T_{CTD(1)}$, using the property Incr. □

8.A.2 Proof of Theorem 8.3.3

PROOF. a) Consider the arrival of an opportunity customer. For $(x,y) \in \mathcal{S}$, $u \in \{0,1\}$, and $n \geq 0$, define

$$w^{(n)}(u,x,y) := \begin{cases} V_n(x,y) + C_{opp} & \text{if } u = 0 \text{ (reject)}, \\ V_n(x+1,y) & \text{if } u = 1 \text{ (accept)}, \end{cases}$$

where $V_n(x,y) := \infty$ if $(x,y) \notin \mathcal{S}$. Hence $T_{CA(1)} V_n(x,y) = \min_{u\in\{0,1\}} w^{(n)}(u,x,y)$. Define, for $u \in \{0,1\}$ and $n \geq 0$:

$$\Delta w_y^{(n)}(u,x,y) := w^{(n)}(u,x,y+1) - w^{(n)}(u,x,y).$$

Then, for each $n \geq 0$:

$$\Delta w_y^{(n)}(1,x,y) - \Delta w_y^{(n)}(0,x,y) = V_n(x+1,y+1) - V_n(x+1,y) - V_n(x,y+1) + V_n(x,y) \geq 0,$$

as, by Theorem 8.3.2, V_n is MM and hence Supermod. So, $\Delta w_y^{(n)}(u,x,y)$ is increasing in u:

$$\Delta w_y^{(n)}(1,x,y) \geq \Delta w_y^{(n)}(0,x,y).$$

This implies that, for every $n \geq 0$, there exists a threshold for y, say $T^{(n)}$, from which on it is optimal to reject an opportunity customer. Hence, the described optimal policy structure holds for an arbitrary $n \geq 0$. As $V_n \to V$ as $n \to \infty$, it follows that V satisfies properties (7.3.1)–(7.3.4) (cf. Lemma 7.3.1), and hence satisfies the above results as well. Then, from (7.2.1) (for $n \to \infty$) it follows that this is also the optimal policy structure in the limit as $n \to \infty$.

b) Consider a (potential) service completion. For $(x,y) \in \mathcal{S}$, $u \in \{0,1\}$, and $n \geq 0$, define

$$m^{(n)}(\mu,x,y) := c(\mu) + \mu V_n((x-1)^+, y) + (1-\mu) V_n(x,(y-1)^+). \tag{8.A.2}$$

Hence $\tilde{T}_{CTD(1)} V_n(x,y) = \min_{\mu\in[0,1]} m^{(n)}(\mu,x,y)$.

Define, for all $\mu \in [0,1]$ and and $n \geq 0$:

$$\Delta m_y^{(n)}(\mu,x,y) := m^{(n)}(\mu,x,y+1) - m^{(n)}(\mu,x,y)$$

We only need to consider $x > 0$, as when $x = 0$, $\mu = 0$ by assumption. So, for each $n \geq 0$, $x > 0$, $y > 0$, and $0 \leq \mu_1 \leq \mu_2 \leq 1$:

$$\begin{aligned}
&\Delta m_y^{(n)}(\mu_1,x,y) - \Delta m_y^{(n)}(\mu_2,x,y) \\
&= \mu_1 V_n(x-1,y+1) + (1-\mu_1) V_n(x,y) - \mu_1 V_n(x-1,y) - (1-\mu_1) V_n(x,y-1) \\
&\qquad - \mu_2 V_n(x-1,y+1) - (1-\mu_2) V_n(x,y) + \mu_2 V_n(x-1,y) + (1-\mu_2) V_n(x,y-1) \\
&= (\mu_1 - \mu_2)\Big(V_n(x-1,y+1) - V_n(x,y) - V_n(x-1,y) + V_n(x,y-1)\Big) \leq 0,
\end{aligned}$$

as the first factor is non-positive, as $\mu_1 \leq \mu_2$, and the second factor is non-negative, as by Theorem 8.3.2, V_n is MM and hence SuperC(y,x). So, $\Delta m_y^{(n)}(\mu,x,y)$ is increasing in μ:

$$\Delta m_y^{(n)}(\mu_1,x,y) \leq \Delta m_y^{(n)}(\mu_2,x,y) \text{ for } \mu_1 \leq \mu_2.$$

This implies that, for every $n \geq 0$, the optimal fraction of the server speed dedicated to an opportunity customer, is decreasing in y.

When $y = 0$ (and $x > 0$), we find

$$\Delta m_y^{(n)}(\mu_1,x,0) - \Delta m_y^{(n)}(\mu_2,x,0) = \big(\mu_1 - \mu_2\big)\big(V_n(x-1,1) - f(x-1,0)\big) \leq 0,$$

as the second factor in non-negative as V_n is Incr(y). Hence, also for $y = 0$ the result holds. □

8.A.3 Proof of Theorem 8.5.1

PROOF. a) The proof is along the same lines as the proof of Theorem 8.5.2, but then for an opportunity customer instead of a regular customer.
b) The proof is along the same lines as the proof of Theorem 8.3.3, part b), generalized to allow for $x > 1$. □

8.A.4 Proof of Theorem 8.5.2

PROOF. Consider the arrival of a regular customer. For $(x,y) \in \mathcal{S}$, $u \in \{0,1\}$, and $n \geq 0$, define

$$\tilde{w}^{(n)}(u,x,y) := \begin{cases} V_n(x,y) + C_{reg} & \text{if } u = 0 \text{ (reject)}, \\ V_n(x,y+1) & \text{if } u = 1 \text{ (accept)}, \end{cases}$$

Hence $T_{CA(2)}V_n(x,y) = \min_{u\in\{0,1\}} \tilde{w}^{(n)}(u,x,y)$.

Define, for $u \in \{0,1\}$ and $n \geq 0$:

$$\Delta \tilde{w}_y^{(n)}(u,x,y) := \tilde{w}^{(n)}(u,x,y+1) - \tilde{w}^{(n)}(u,x,y).$$

Then, for each $n \geq 0$:

$$\Delta \tilde{w}_y^{(n)}(1,x,y) - \Delta \tilde{w}_y^{(n)}(0,x,y) = V_n(x,y+2) - V_n(x,y+1) - V_n(x,y+1) + V_n(x,y) \geq 0,$$

as, by Theorem 8.3.2, V_n is MM and hence Conv(y). So, $\Delta \tilde{w}_y^{(n)}(u,x,y)$ is increasing in u:

$$\Delta \tilde{w}_y^{(n)}(1,x,y) \geq \Delta \tilde{w}_y^{(n)}(0,x,y).$$

This implies that, for every $n \geq 0$, there exists a threshold for y, which can depend on x, say $T_{reg}^{(n)}(x)$, from which on it is optimal to reject a regular customer. By the same reasoning as in the proof of part a) of Theorem 8.3.3, this proves the optimality of a state-dependent threshold policy.

Define, for $u \in \{0,1\}$ and $n \geq 0$:

$$\Delta \tilde{w}_{x-y}^{(n)}(u,x,y) := \tilde{w}^{(n)}(u,x+1,y) - \tilde{w}^{(n)}(u,x,y+1).$$

Then, for each $n \geq 0$:

$$\Delta\tilde{w}^{(n)}_{x-y}(1,x,y) - \Delta\tilde{w}^{(n)}_{x-y}(0,x,y) \tag{8.A.3}$$
$$= V_n(x+1,y+1) - V_n(x,y+2) - V_n(x+1,y) + V_n(x,y+1) \leq 0 \tag{8.A.4}$$

as, by Theorem 8.3.2, V_n is MM and hence SuperC(y,x). So, $\Delta\tilde{w}^{(n)}_{x-y}(u,x,y)$ is decreasing in u:

$$\Delta\tilde{w}^{(n)}_{x}(1,x,y) \leq \Delta\tilde{w}^{(n)}_{x}(0,x,y).$$

Hence, $T_{reg}(x)$ is strictly decreasing in x (until it reaches 0). □

8.B Appendix: Steady-state probability distribution

We show the derivation of the stationary probability distribution under the assumption that $T < M$. The cases $M = T$ and $M \geq T$ proceed along the same lines.

For deriving the stationary probability distribution of (x,y), we partition the state space into three mutually disjoint subsets. The upper part consists of all (x,y) for which $y \geq M$, the middle part of all (x,y) for which $T \leq y < M$, and the lower part of all (x,y) for which $0 \leq y < T$. For each part, the transition rates differ, see Figure 8.1. Also, for each part, the expression for $p(x,y)$ contains three (lower and middle) or two (upper) constants, which only depend on the model parameters. These constants can then be found by solving a system of linear equations, which is found by linking the expressions for $p(x,T)$ and $p(x,M)$, $x = 0,1$. Together with the boundary conditions and the normalization equation, one can then solve these constants.

Lower part. For $y \leq T$, we have the system:

$$p(0,y)(\lambda_{opp} + \lambda_{reg} + \mu) = p(0,y-1)\lambda_{reg} + p(0,y+1)\mu + p(1,y)\mu, \tag{8.B.1}$$
$$p(1,y)(\lambda_{reg} + \mu) = p(1,y-1)\lambda_{reg} + p(0,y)\lambda_{opp}. \tag{8.B.2}$$

This holds for the part where $y < T$. Plugging in

$$p(0,y) = v(0)\alpha^y,$$
$$p(1,y) = v(1)\alpha^y.$$

yields, after division by α^{y-1}:

$$v(0)\alpha(\lambda_{opp} + \lambda_{reg} + \mu) = v(0)\lambda_{reg} + v(0)\alpha^2\mu + v(1)\alpha\mu,$$
$$v(1)\alpha(\lambda_{reg} + \mu) = v(1)\lambda_{reg} + v(0)\alpha\lambda_{opp}.$$

Rewriting gives

$$v(0)\Big(\alpha(\lambda_{opp} + \lambda_{reg} + \mu) - \lambda_{reg} - \alpha^2\mu\Big) - v(1)\alpha\mu = 0,$$
$$v(1)\Big(\alpha(\lambda_{reg} + \mu) - \lambda_{reg}\Big) - v(0)\alpha\lambda_{opp} = 0,$$

and hence

$$\begin{pmatrix} \alpha(\lambda_{opp}+\lambda_{reg}+\mu)-\lambda_{reg}-\alpha^2\mu & -\alpha\mu \\ -\alpha\lambda_{opp} & \alpha(\lambda_{reg}+\mu)-\lambda_{reg} \end{pmatrix} \begin{pmatrix} v(0) \\ v(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{8.B.3}$$

The determinant of the matrix is set to 0:

$$\Big(\alpha(\lambda_{opp}+\lambda_{reg}+\mu)-\lambda_{reg}-\alpha^2\mu\Big)\Big(\alpha(\lambda_{reg}+\mu)-\lambda_{reg}\Big)-\alpha^2\lambda_{opp}\mu=0,$$

which gives three solutions for α:

$$\begin{aligned} \alpha_1 &= 1, \\ \alpha_2 &= \frac{\lambda_{reg}}{\lambda_{reg}+\mu}\cdot\frac{\lambda_{opp}+\lambda_{reg}+2\mu-\sqrt{(\lambda_{opp}+\lambda_{reg})^2+4\lambda_{opp}\mu}}{2\mu}, \\ \alpha_3 &= \frac{\lambda_{reg}}{\lambda_{reg}+\mu}\cdot\frac{\lambda_{opp}+\lambda_{reg}+2\mu+\sqrt{(\lambda_{opp}+\lambda_{reg})^2+4\lambda_{opp}\mu}}{2\mu}. \end{aligned} \tag{8.B.4}$$

We now solve the system (8.B.3), for α_1, α_2, and α_3 to find a non-null solution for respectively v_1, v_2, and v_3. We find;

$$\begin{aligned} &v_1(0)=1,\ v_1(1)=\lambda_{opp}/\mu, \\ &v_2(0)=1,\ v_2(1)=\frac{\lambda_{reg}-\lambda_{opp}+\sqrt{(\lambda_{opp}+\lambda_{reg})^2+4\lambda_{opp}\mu}}{2(\lambda_{reg}+\mu)}, \\ &v_3(0)=1,\ v_3(1)=\frac{\lambda_{opp}-\lambda_{reg}+\sqrt{(\lambda_{opp}+\lambda_{reg})^2+4\lambda_{opp}\mu}}{2(\lambda_{reg}+\mu)}. \end{aligned} \tag{8.B.5}$$

We write the solution of this system of (8.B.1) and (8.B.2) as:

$$p(x,y)=c_1v_1(x)\alpha_1^y+c_2v_2(x)\alpha_2^y+c_3v_3(x)\alpha_3^y,$$

Plugging in α_1, the $v_i(0)$, $i=1,2,3$, and $v_1(0)$ as in (8.B.5), this becomes

$$p(0,y)=c_1+c_2\alpha_2^y+c_3\alpha_3^y, \tag{8.B.6}$$

$$p(1,y)=c_1\frac{\lambda_{opp}}{\mu}+c_2v_2(1)\alpha_2^y+c_3v_3(1)\alpha_3^y, \tag{8.B.7}$$

where α_2, α_3, $v_2(1)$, and $v_3(1)$ as given in (8.B.4) and (8.B.5) respectively (note: they only depend on λ_{opp}, λ_{reg}, and μ).

Middle part. For $T \le y < M$, we have the system:

$$p(0,y)(\lambda_{reg}+\mu)=p(0,y-1)\lambda_{reg}+p(0,y+1)\mu+p(1,y)\mu, \tag{8.B.8}$$

$$p(1,y)(\lambda_{reg}+\mu)=p(1,y-1)\lambda_{reg}. \tag{8.B.9}$$

Plugging in

$$\begin{aligned} p(0,y)&=w(0)\beta^y, \\ p(1,y)&=w(1)\beta^y. \end{aligned}$$

yields, after division by β^{y-1}:

$$\begin{aligned} w(0)\beta(\lambda_{reg}+\mu) &= w(0)\lambda_{reg} + w(0)\beta^2\mu + w(1)\beta\mu, \\ w(1)\beta(\lambda_{reg}+\mu) &= w(1)\lambda_{reg}. \end{aligned}$$

Rewriting gives

$$\begin{aligned} w(0)\Big(\beta(\lambda_{reg}+\mu) - \lambda_{reg} - \beta^2\mu\Big) - w(1)\beta\mu = 0, \\ w(1)\Big(\beta(\lambda_{reg}+\mu) - \lambda_{reg}\Big) = 0, \end{aligned}$$

and hence

$$\begin{pmatrix} \beta(\lambda_{reg}+\mu) - \lambda_{reg} - \beta^2\mu & -\beta\mu \\ 0 & \beta(\lambda_{reg}+\mu) - \lambda_{reg} \end{pmatrix} \begin{pmatrix} w(0) \\ w(1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{8.B.10}$$

The determinant of the matrix is set to 0:

$$\Big(\beta(\lambda_{reg}+\mu) - \lambda_{reg} - \beta^2\mu\Big)\Big(\beta(\lambda_{reg}+\mu) - \lambda_{reg}\Big) = 0,$$

which gives three solutions for β:

$$\begin{aligned} \beta_1 &= 1, \\ \beta_2 &= \lambda_{reg}/\mu, \\ \beta_3 &= \lambda_{reg}/(\lambda_{reg}+\mu). \end{aligned} \tag{8.B.11}$$

We now solve the system (8.B.10), for β_1, β_2, and β_3 to find a non-null solution for respectively w_1, w_2, and w_3. We find:

$$\begin{aligned} &w_1(0) = 1,\ w_1(1) = 0, \\ &w_2(0) = 1,\ w_2(1) = 0, \\ &w_3(0) = 1,\ w_3(1) = -\lambda_{reg}/(\lambda_{reg}+\mu). \end{aligned} \tag{8.B.12}$$

We write the solution of the system of (8.B.8) and (8.B.9) as:

$$p(x,y) = d_1 w_1(y)\beta_1^{x-T1} + d_2 w_2(y)\beta_2^{x-T1} + d_3 w_3(y)\beta_3^{x-T1}.$$

Plugging in the β_i's and $w_i(y)$'s as given in (8.B.11) and (8.B.12) respectively, this becomes

$$p(x,0) = d_1 + d_2\left(\frac{\lambda_{reg}}{\mu}\right)^{y-T} + d_3\left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{y-T}, \tag{8.B.13}$$

$$p(x,1) = -d_3\left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{y+1-T}. \tag{8.B.14}$$

Upper part. For $y \geq M$, we have

$$p(x,y) = q(x)\left(\frac{\lambda_{reg}}{\mu}\right)^{y-M}. \tag{8.B.15}$$

Boundary. The boundary conditions are given by

$$p(0,0)(\lambda_{opp}+\lambda_{reg}) = p(0,1)\mu + p(1,0)\mu, \tag{8.B.16}$$
$$p(1,0)(\lambda_{reg}+\mu) = p(0,0)\lambda_{opp}. \tag{8.B.17}$$

We can discard one equation, say (8.B.16), and replace it by the normalization equation:

$$\sum_{(x,y)} p(x,y) = 1. \tag{8.B.18}$$

Solving the system. Note that we have 8 unknown: $q(0)$, $q(1)$, d_1, d_2, d_3, c_1, c_2, and c_3. We solve them by deriving 8 linear equations. For this, we consider the expressions for the coordinates:

$$(0,0),\ (1,0),\ (0,T),\ (1,T),\ (0,M),\ (1,M),$$

where for $(0,T)$ and $(0,M)$ two linear equations are derived, hence the total number of equations equals 8.

For $p(0,0)$ and $p(1,0)$ we have the boundary equation as given in (8.B.16) and (8.B.17) respectively. We plug in (8.B.6) and (8.B.7) to find:

$$\begin{aligned}(\lambda_{opp}+\lambda_{reg})\Big(c_1+c_2+c_3\Big) &= \mu\Big(c_1+c_2\alpha_2+c_3\alpha_3\Big)\\ &\quad + \mu\Big(c_1\frac{\lambda_{opp}}{\mu}+c_2v_2(1)+c_3v_3(1)\Big),\\ (\lambda_{reg}+\mu)\Big(c_1\frac{\lambda_{opp}}{\mu}+c_2v_2(1)+c_3v_3(1)\Big) &= \lambda_{opp}\Big(c_1+c_2+c_3\Big).\end{aligned}$$

For $(1,T)$ we have from (8.B.14) and (8.B.7):

$$-d_3\frac{\lambda_{reg}}{\lambda_{reg}+\mu} = c_1\frac{\lambda_{opp}}{\mu}+c_2v_2(1)\alpha_2^T+c_3v_3(1)\alpha_3^T. \tag{8.B.19}$$

For $(0,T)$ we have from (8.B.13) and (8.B.6):

$$d_1+d_2+d_3 = c_1+c_2\alpha_2^T+c_3\alpha_3^T. \tag{8.B.20}$$

For $(0,T)$ we have as well, from (8.B.1), plugging in (8.B.6) for $p(0,T-1)$, plugging in (8.B.13) for $p(0,T)$ and $p(0,T+1)$, and plugging in (8.B.14) for $p(1,T)$, which gives, after simplification:

$$(\lambda_{opp}+\lambda_{reg}+\mu)(d_1+d_2+d_3) = \lambda_{reg}\Big(c_1+d_2+c_2\alpha_2^{T-1}+c_3\alpha_3^{T-1}\Big)+d_1\mu. \tag{8.B.21}$$

Note that we could also have used (8.B.6) and (8.B.7) for $p(0,T)$ and $p(1,T)$ respectively, as these are set equal to (8.B.13) and (8.B.14) in (8.B.20) and (8.B.19) respectively.

For $(1,M)$ we have from (8.B.15) and (8.B.14):

$$q(1) = -d_3\left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{M+1-T}. \tag{8.B.22}$$

For $(0,M)$ we have from (8.B.15) and (8.B.13):

$$q(0) = d_1 + d_2 \left(\frac{\lambda_{reg}}{\mu}\right)^{M-T} + d_3 \left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{M-T}. \tag{8.B.23}$$

For $(0,M)$ we have as well, from (8.B.8), plugging in (8.B.13) for $p(0,M-1)$ and plugging in (8.B.15) for $p(0,M)$, $p(0,M+1)$, and $p(1,M)$, which gives, after simplification:

$$\mu\big(q(0)-q(1)\big) = d_1\,\lambda_{reg} + d_2\,\mu \left(\frac{\lambda_{reg}}{\mu}\right)^{M-T} + d_3 \left(\frac{\lambda_{reg}}{\lambda_{reg}+\mu}\right)^{M-T} (\lambda_{reg}+\mu). \tag{8.B.24}$$

Note that we could also have used (8.B.13) and (8.B.14) for $p(0,M)$ and $p(1,M)$ respectively, as these are set equal to (8.B.15) for $y=0$ and $y=1$ in (8.B.23) and (8.B.22) respectively.

Finally, we use the boundary condition (8.B.17). We plug in (8.B.6) and (8.B.7) for $p(0,0)$ and $p(1,0)$ respectively. This gives

$$(\lambda_{reg}+\mu)\left(c_1\frac{\lambda_{opp}}{\mu} + c_2 v_2(1) + c_3 v_3(1)\right) = \lambda_{opp}\big(c_1+c_2+c_3\big). \tag{8.B.25}$$

We now have a linear system of 8 equations: (8.B.19), (8.B.20), (8.B.21), (8.B.22), (8.B.23), (8.B.24), (8.B.25), and the normalization equation (8.B.18). We can solve this system to find the 8 unknowns ($q(0)$, $q(1)$, d_1, d_2, d_3, c_1, c_2, and c_3), in terms of λ_{opp}, λ_{reg}, and μ. Hence, from (8.B.6), (8.B.7), (8.B.13), (8.B.14), and (8.B.15) we find the steady-state probabilities $p(x,y)$ in terms of λ_{opp}, λ_{reg}, and μ.

Part II

Polling

9

Introduction polling

In the second part of this thesis, we study polling models. A polling model is a queueing model consisting of multiple queues that are served cyclically by a single server. In the first two chapters, we derive general results for the mean performance analysis of polling models. Then, we present two chapters studying the operation of a polling model in such a way as to achieve both 'fairness' and 'efficiency'.

We provide a general introduction to polling models in this chapter. We introduce the model and the notation used throughout the chapters. Moreover, we present techniques known in the literature to derive the main performance characteristics of a polling model. Of most importance for our studies, are the mean steady-state waiting times at each of the queues. We discuss those techniques that are used in multiple chapters. Finally, we give an overview of the chapters, addressing the models studied and the contributions.

A polling model can be interpreted as a form of pooling of the server capacity. Namely, the single server pools its capacity between the queues. The main difference with the studies in the first part of this thesis, is that we assume a given policy, e.g. the cyclic routing and the service discipline at each of the queues. Given such a policy, we derive the performance characteristics, or we optimize the parameters within a class of policies. Also, we adapt existing policies to create new strategies.

9.1 Introduction

A *polling system* is a queueing system with multiple queues and one single server. Typically, the server visits the queues in a cyclic order, where at each queue it serves the customers. A so-called *switch-over time* is incurred when the server switches from one queue to another. Such a setting was first studied by Mack [134, 135] in 1957, for a patrolling machine repairman problem in the British cotton industry. The studies determine the efficiency of such a repairman, who walks round a set of machines, inspecting, servicing, and, when necessarily, fixing each of them upon a visit. Since these pioneering works, a huge body of literature has been developed in many application areas. The term 'polling' dates back to the application of a central computer which cyclically *polls* terminals to see whether these have data to transmit. This is called a 'polling data link control scheme'. Here, the central computer acts as the server, and the terminals are the queues.

Polling models are used in the modeling of many problems. Next to the patrolling repairman and the data transmission application, they are used in e.g. telecommunication, computer networks, maintenance problems, and production systems. The surveys of Takagi [177], Levy and Sidi [130], and Vishnevskii and Semenova [200] provide good overviews of applications of polling systems. A recent literature study can be found in Boon et al. [27], which categorizes the literature both on the models used and on the applications studied. Also Winands [208] provides an extensive literature review. Takagi [176] is a seminal work on polling models.

In a polling model, a key issue is the decision when the server should switch to the next queue. The strategy for this, applied at a queue, is called the *service discipline*, for which there are many possible choices. Those most often studied are the *exhaustive service discipline* (when the server serves all customers at a queue, until the queue has become empty) and the *gated service discipline* (when the server arrives at a queue, a gate closes and only the customers who are before the gate, i.e., who are already present, will be served in this server visit). The service discipline might vary at each of the queues, although often the same discipline is chosen for all queues. The service disciplines influence both the performance (waiting times and queue lengths) at each of the queues, as well as the overall performance of the system. We study both the classical gated and exhaustive discipline, but moreover we introduce variations on and new combinations of these disciplines.

Note that we assume a given policy, and either evaluate the performance characteristics (exact or approximately) or optimize the parameters the policy depends on. The complexity of polling models makes the study of optimal dynamic policies almost intractable. Hence, such policies have hardly been studied in the literature, and when studied, only for special cases such as a two-queue model, or heavy traffic limits [136, 137, 159].

9.2 Model and notation

We consider a polling system [176], with N queues, $Q_1, \ldots, Q_N$, where each queue has infinite capacity. The queues are served by a single server, in fixed cyclic order $Q_1, Q_2, \ldots, Q_N$, $Q_1, Q_2, \ldots$. Customers in each queue are served in order of arrival: first come, first served (in [33] variations on this assumption are studied). The arrival processes at the queues are independent Poisson processes with arrival rate λ_i at Q_i, $i = 1, \ldots, N$. The service times at Q_i are i.i.d. non-negative random variables, denoted by B_i, having finite first and second moment, and Laplace-Stieltjes transform (LST) $\beta_i(.)$. The switch of the server from Q_{i-1} to Q_i (from Q_N to Q_1 for $i = 1$) lasts for a switch-over time S_i, these being i.i.d. nonnegative random variables, with finite first two moments, and LST $\sigma_i(.)$. The sum of the switch-over times is denoted by $S = \sum_{i=1}^{N} S_i$, where we assume $\mathbb{E}[S] > 0$ (otherwise the mean cycle length in steady-state becomes zero and the analysis changes slightly; see [30] for the relation between polling systems with and without switch-over times). The (equilibrium) residual length of a random variable X is denoted by X^{res} with $\mathbb{E}[X^{res}] = \mathbb{E}[X^2]/(2\,\mathbb{E}[X])$. We assume that the arrival processes, the service times and the switch-over times are all mutually independent. Customers at Q_i are referred to as type i customers. Indices are understood to be modulo N: Q_{N+1} actually refers to Q_1.

The service discipline applied at a queue determines when the server switches to the next queue. We focus on the gated and exhaustive service disciplines. In case of gated service (*gat*), the server serves exactly the customers present upon its arrival at the queue. In case of exhaustive service (*exh*), the server serves customers until the queue where it is working on, is empty. Both disciplines are, from a practical point of view, relevant, and allow for exact analysis. When all queues are served according to the exhaustive (gated) service discipline, we refer to this as a system with purely exhaustive (gated) services. We restrict our attention to a single service discipline for all queues; in [24, 26] combinations of disciplines are studied.

The traffic offered per time unit at Q_i is denoted by ρ_i and is given by $\rho_i = \lambda_i \mathbb{E}[B_i]$. The total traffic offered to the system per time unit is $\rho = \sum_{i=1}^{N} \rho_i$. A necessary and sufficient condition for stability in case of gated and exhaustive services, is $\rho < 1$, see [82]. In the sequel we assume $\rho < 1$, and we concentrate on the steady-state behavior of the system. We are mainly interested in the waiting times of customers. By W_i we denote the steady-state waiting time of a type i customer, excluding its own service time. Also, by L_i we denote the steady-state queue length of Q_i, excluding the customer in service. Furthermore, BP_i denotes a busy period induced by a type i customer, having first moment $\mathbb{E}[BP_i] = \mathbb{E}[B_i]/(1-\rho_i)$.

By C_i we denote the cycle time starting from Q_i and consisting of the visit times to each of the queues and all switch-over times incurred. A well-known result [176] is that its first moment does not depend on i and is given by $\mathbb{E}[C] = \mathbb{E}[S]/(1-\rho)$. Moreover, $\mathbb{E}[C]$ does not depend on the service disciplines at the queues. A cycle time can be divided into the *visit time* to Q_i and the *intervisit time* of Q_i. The visit time is the duration that the server is serving the queue, and is denoted by V_i. Its first moment is given by $\mathbb{E}[V_i] = \rho_i \mathbb{E}[C]$, as a fraction ρ_i of the time the server is working at Q_i. The intervisit time is the duration between the moment the server leaves the queue until it starts working on it again. It is denoted by I_i, and using that $\mathbb{E}[C] = \mathbb{E}[V_i] + \mathbb{E}[I_i]$ for all i, it follows that its first moment is given by $\mathbb{E}[I_i] = (1-\rho_i)\mathbb{E}[C]$.

9.3 Techniques

Various analysis techniques have been presented in the literature in order to derive performance characteristics of polling models, such as waiting times and queue lengths. These include the Pseudo Conservation Law, Mean Visit Times, Mean Value Analysis for polling systems, and Multi-type Branching Processes. These concepts are discussed below, as they are used in the sequel of this thesis. We mention that other techniques have been derived in the polling literature for determining the performance characteristics of polling models, which include the Buffer Occupancy approach [57, 56, 68, 114], the Descendant Set approach [111, 112], and the Station Time approach [78] (see also [129, 130, 177] for a discussion on these methods). However, we do not use these in our studies.

9.3.1 Pseudo conservation law

As typically a switch-over time is incurred when the server switches from one queue to another, polling models are not work conserving, and hence ordinary work conservation laws do not hold in general. However, Boxma and Groenendijk [34] derive a so-called *Pseudo Conservation Law* (PCL) for the case of cyclic order polling systems. These pseudo conservation laws give an expression for the weighted sum of the mean waiting times at each of the queues: $\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i]$. Based on a workload decomposition result, the following expression is derived, cf. [34, (3.10)]:

$$\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] = \frac{\rho}{1-\rho} \sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}] + \rho \mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)} \left(\rho^2 - \sum_{i=1}^{N} \rho_i^2 \right) + \sum_{i=1}^{N} \mathbb{E}[M_i], \tag{9.3.1}$$

where $\mathbb{E}[M_i]$ is the mean amount of work in Q_i at a departure epoch of the server from Q_i. This is the only term that depends on the service discipline at the queues. For the exhaustive discipline, trivially $\mathbb{E}[M_i^E] = 0$ (cf. [34, (3.11)]), and for the gated discipline, $\mathbb{E}[M_i^G] = \rho_i \mathbb{E}[V_i] = \rho_i^2 \mathbb{E}[S]/(1-\rho)$ (cf. [34, (3.12)]).

The expression in (9.3.1) is derived by considering a workload decomposition. Denote by V_{with} the amount of work in the cyclic service system at an arbitrary epoch in time, by $V_{without}$ the amount of work in the same system but without switch-over times at an arbitrary epoch in time, and by Y the amount of work in the system at an arbitrary epoch in a switch-over interval. It has been proven in [34] that $V_{without}$ and Y are independent and that the following relation holds:

$$V_{with} \stackrel{d}{=} V_{without} + Y,$$

where $\stackrel{d}{=}$ denotes equality in distribution. This gives that

$$\mathbb{E}[V_{with}] = \mathbb{E}[V_{without}] + \mathbb{E}[Y].$$

The mean amount of work in the system without switch-over times is given by

$$\mathbb{E}[V_{without}] = \frac{\sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}]}{1-\rho}, \tag{9.3.2}$$

independent of the service strategies. Denoting by L_i the number of customers at Q_i, then we also have

$$\begin{aligned} \mathbb{E}[V_{with}] &= \sum_{i=1}^{N} \mathbb{E}[B_i]\mathbb{E}[L_i] + \sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}] \\ &= \sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] + \sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}]. \end{aligned} \tag{9.3.3}$$

Combining (9.3.2) and (9.3.3) gives

$$\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] = \rho \frac{\sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}]}{1-\rho} + \mathbb{E}[Y]. \tag{9.3.4}$$

By Y_i we denote the amount of work at an arbitrary epoch in a switch-over interval when switching to Q_i. Then $\mathbb{E}[Y]$ is the weighted sum of $\mathbb{E}[Y_i]$:

$$\mathbb{E}[Y] = \sum_{i=1}^{N} \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]} \mathbb{E}[Y_i]. \tag{9.3.5}$$

In order to find the PCL, it remains to determine $\mathbb{E}[Y_i]$ for $i = 1, \dots, N$. Note that these depend on the service disciplines at the queues. In [34] this is done for the cases of purely exhaustive and purely gated services, and also for mixtures of these, which result in (9.3.1).

As the PCL gives an expression for $\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i]$, it is in that way a measure for the *efficiency* of the service disciplines at the queues. Exhaustive is the most efficient service discipline, as the server never switches when there are still customers in the queue it is serving. So, it leaves no customers behind that have to wait for an entire cycle. The latter is the case for the gated discipline, which is less efficient. In Chapters 12 and 13 we use the PCL when investigating the efficiency of (variations on) the exhaustive and gated disciplines.

9.3.2 Mean value analysis

The first moments of the waiting times, $\mathbb{E}[W_i]$, can be obtained in an efficient way using *mean value analysis* (MVA) for polling systems, introduced by Winands, Adan and Van Houtum [210] (see also [208]). The main idea is the setting up of a system of linear equations, making use of PASTA and Little's Law. Each equation has a probabilistic and intuitive explanation. For a system with purely exhaustive or purely gated service, a system of N^2, respectively $N(N+1)$ linear equations is derived. The system can (numerically) be solved in order to find the unknowns, in particular, the $\mathbb{E}[W_i]$'s. Below, we show the MVA approach for a systems with the exhaustive service discipline at all queues; the cases of gated or mixed service disciplines require only minor changes (see [210] for details).

The mean waiting time $\mathbb{E}[W_i]$ of a type i customer can be expressed in the following way: upon arrival of a (tagged) type i customer, it has to wait for the (residual) time it takes to serve all type i customers already present in the system, plus possibly the time before the server arrives at Q_i. By PASTA, the arriving customer finds in expectation $\mathbb{E}[L_i]$ waiting type i customers in queue, each having an expected service time $\mathbb{E}[B_i]$, and with probability ρ_i, it finds a type i customer currently in service, hence having to wait for a mean residual service time $\mathbb{E}[B_i^{res}]$. Let T_i be the time it takes before the server starts working on Q_i again (which depends on the service discipline at the queues). This gives, for $i = 1, \dots, N$:

$$\mathbb{E}[W_i^{exh}] = \mathbb{E}[L_i]\,\mathbb{E}[B_i] + \rho_i\,\mathbb{E}[B_i^{res}] + (1-\rho_i)\mathbb{E}[T_i]. \tag{9.3.6}$$

Moreover, Little's Law gives, for $i = 1, \dots, N$,

$$\mathbb{E}[L_i] = \lambda_i\,\mathbb{E}[W_i]. \tag{9.3.7}$$

Hence, it remains to derive $\mathbb{E}[T_i]$.

For this purpose, a system of equations is composed for the *conditional* mean queue lengths: $\mathbb{E}[L_{ij}]$, which is the expected queue length at Q_i during a switch-over time to or visit time at Q_j. These can be expressed in mean residual durations of (sums of) visit and switch-over times. We define period i as the switch-over time to plus the visit time at Q_i. Clearly, Q_i is empty at the end of period i. Denote by q_i the fraction of the time the system is in period i: $q_i = (\mathbb{E}[S_i] + \mathbb{E}[V_i])/\mathbb{E}[C]$, where $\mathbb{E}[V_i] = \rho_i \mathbb{E}[C]$. Then the mean number of type i customers waiting in the queue, $\mathbb{E}[L_i]$, is a weighted average of the $\mathbb{E}[L_{ij}]$, so for $i = 1, \ldots, N$:

$$\mathbb{E}[L_i] = \sum_{j=1}^{N} q_j \, \mathbb{E}[L_{ij}].$$

Further, let the interval (i, j) consist of the periods $i, i+1, \ldots, i+j-1$ and denote by $\mathbb{E}[R_{ij}]$ the expected residual time of interval (i, j). One can derive a system of linear equations relating $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$ (see [210, Eq. (6)–(9)]). The final step is to use the solution of this system to determine the $\mathbb{E}[T_i]$, and hence the $\mathbb{E}[L_i]$ and $\mathbb{E}[W_i]$ follow.

For (purely) gated services, one defines period i as the visit time at Q_i plus the switch-over time to Q_{i+1}. In this case, there are no customers behind the gate of Q_i at the start of period i. For the mean waiting time, we have $\mathbb{E}[W_i^{gat}] = \mathbb{E}[L_i]\,\mathbb{E}[B_i] + \mathbb{E}[T_i]$, as under the gated discipline an arriving type i customer always has to wait until the server starts working (on the customers behind the gate) at Q_i. The equations relating $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$ (and hence $\mathbb{E}[T_i]$) differ from those for the exhaustive case, see [210, Eq. (12)–(16)].

We illustrate the MVA approach with a two-queue example, where both queues are served according to the exhaustive discipline. The mean waiting times are as given in (9.3.6), and we use Little's Law, cf. (9.3.7). For the mean number of type 1 customers in the system, we have

$$\mathbb{E}[L_1] = q_1 \mathbb{E}[L_{11}] + q_2 \mathbb{E}[L_{12}].$$

As Q_1 is empty at the end of a server visit, which is the start of period 2, $\mathbb{E}[L_{12}]$ is just the mean number of type 1 customers that have arrived during period 2. Since the mean duration of period 2 already past is equal to the mean residual period duration, which is $\mathbb{E}[R_{21}]$, we find

$$\mathbb{E}[L_{12}] = \lambda_1 \, \mathbb{E}[R_{21}].$$

Because of the exhaustive service discipline, we have

$$\mathbb{E}[R_{21}] = \mathbb{E}[L_{22}]\,\mathbb{E}[BP_2] + \frac{\mathbb{E}[S_2]}{\mathbb{E}[S_2] + \mathbb{E}[V_2]} \frac{\mathbb{E}[{S_2}^{res}]}{1-\rho_2} + \frac{\mathbb{E}[V_2]}{\mathbb{E}[S_2] + \mathbb{E}[V_2]} \frac{\mathbb{E}[{B_2}^{res}]}{1-\rho_2},$$

where the terms $1/(1-\rho_2)$ come from the fact that each arriving type 2 customer induces a busy period. Finally, for $\mathbb{E}[T_1]$ we have

$$(1-\rho_1)\mathbb{E}[T_1] = \frac{\mathbb{E}[S_1]}{\mathbb{E}[C]} \mathbb{E}[{S_1}^{res}] + q_2(\mathbb{E}[R_{21}] + \mathbb{E}[S_1]).$$

Starting from a type 2 customer, a similar set of equations can be derived. From these equations, we can solve $\mathbb{E}[R_{i1}]$, $\mathbb{E}[L_{ij}]$, and $\mathbb{E}[T_i]$, and hence the $\mathbb{E}[L_i]$ and $\mathbb{E}[W_i]$ follow.

9.3.3 Multi-type branching processes

By the use of multi-type branching processes, polling systems where the service disciplines satisfy the so-called *branching property* [160, Property 1], can generally be analyzed exactly. The branching property states (where pgf stands for probability generating function):

> If the server arrives at Q_i to find n_i customers there, then during the course of the server's visit, each of these n_i customers will effectively be replaced in an i.i.d. manner by a random population having pgf $h_i(z_1,\dots,z_N)$, which can be any N-dimensional pgf.

Both the gated service discipline and the exhaustive service discipline satisfy this property. We determine the LST of the waiting times W_i analogously to Resing [160]. Given that the service discipline in each queue satisfies the branching property, the queue length process at polling instants of a fixed queue (the moments the server starts working on this queue) is a multi-type branching process (MTBP) with immigration in each state. This leads to expressions for the generating function of the joint queue length process at polling instants. Conform e.g. [25] we then derive the LSTs of the steady-state waiting times.

Let the start of the visit to Q_1 be the start of the cycle. By the branching property, each customer present will during the cycle be replaced in an i.i.d. manner by customers of type $1,\dots,N$, according to the probability generating function $h_i(z)$, where $z=(z_1,\dots,z_N)$. For the gated service discipline, this h_i is given by:

$$h_i^{gat}(z_1,z_2,\dots,z_N)=\beta_i\left(\sum_{j=1}^{N}\lambda_j(1-z_j)\right). \tag{9.3.8}$$

Recall that $\beta_i(\cdot)$ is the LST of the service time distribution of a type i customer. For the exhaustive service discipline, h_i is given by

$$h_i^{exh}(z_1,z_2,\dots,z_N)=\theta_i\left(\sum_{j\neq i}\lambda_j(1-z_j)\right), \tag{9.3.9}$$

where $\theta_i(\cdot)$ is the LST of a busy period triggered by one type i customer in Q_i in isolation.

Now [160, Theorem 2.2] states that, for a cyclic polling model where the service disciplines at each queue Q_i satisfy the branching property with pgf $h_i(z)$, the numbers of customers in Q_1 at successive time points that the server reaches Q_1 constitute a MTBP with immigration in each state. The offspring pgfs $f^{(i)}(z)$ are given by

$$f^{(i)}(z)=h_i\big(z_1,\dots,z_i,f^{(i+1)}(z),\dots,f^{(N)}(z)\big). \tag{9.3.10}$$

The pgf of the n^{th} generation offspring can be recursively defined as:

$$\begin{aligned}
&f_n(z)=(f^{(1)}(f_{n-1}(z)),\dots,f^{(N)}(f_{n-1}(z))), \quad n\geq 1,\\
&f_0(z)=z.
\end{aligned}$$

The immigration pgf $g(z)$ is given by

$$g(z)=\prod_{i=1}^{N}\sigma_{i+1}\Big(\sum_{k=1}^{i}\lambda_k(1-z_k)+\sum_{k=i+1}^{N}\lambda_k(1-f^{(k)}(z))\Big). \tag{9.3.11}$$

Recall that $\sigma_i(\cdot)$ is the LST of the switch-over time distribution when switching to Q_i, and index $N+1$ should understood to be 1.

Then the pgf P of the stationary distribution $\pi(j_1, j_2, \ldots, j_N)$ of the number of customers present in the system at the moment that the server starts working on Q_1, is given by

$$P(z) = \prod_{n=0}^{\infty} g(f_n(z)), \tag{9.3.12}$$

The infinite product in (9.3.12) converges only if $\rho < 1$. By renumbering the queues, in the same way we can find expressions for the queue lengths at the moment that the server starts working on Q_i, $i = 2, \ldots, N$.

Alternatively, let $V_{b_i}(z)$ and $V_{c_i}(z)$ be the pgfs of the steady-state joint queue length distributions at the beginning and completion, respectively, of a visit to Q_i (analogously to [25], hence $P = V_{b_1}$). We can express $V_{b_1}(z)$ in itself, by repeated application of the following relation, cf. [25, (2.2)]:

$$\begin{aligned} V_{b_{i+1}}(z) &= V_{c_i}(z)\sigma_i\Big(\sum_{j=1}^{N}\lambda_j(1-z_j)\Big) \\ &= V_{b_i}(z_1, \ldots, z_{i-1}, h_i(z), z_{i+1}, \ldots, z_N)\sigma_i\Big(\sum_{j=1}^{N}\lambda_j(1-z_j)\Big),\ i = 1, 2, \ldots, N, \end{aligned} \tag{9.3.13}$$

where $N+1$ is understood to be 1. Hence, we have a recursive expression, from which the V_{b_i} can be solved.

The LST of the steady-state waiting time distribution of a type i customer is given by, cf. [25, (2.8)]:

$$\mathbb{E}[\mathrm{e}^{-\omega W_i}] = \frac{\tilde{V}_{c_i}(1-\omega/\lambda_i) - \tilde{V}_{b_i}(1-\omega/\lambda_i)}{(\omega - \lambda_i(1-\beta_i(\omega)))\mathbb{E}[C]}, \tag{9.3.14}$$

where $\tilde{V}_{b_i}(\cdot)$ is the pgf of the steady-state marginal queue length distribution at a visit beginning of Q_i, given by $\tilde{V}_{b_i}(z) = V_{b_i}(1, \ldots, 1, z, 1, \ldots, 1)$, with z as the ith argument, and $\tilde{V}_{c_i}(\cdot)$ is defined analogously. By differentiation, moments of the steady-state waiting time for an arbitrary type i customer can be derived.

Note that polling systems that do *not* satisfy the branching property, rarely can be analyzed in an exact way. See [83, 160] for more details on this branching property.

9.4 Overview

We now discuss the models, research questions and contributions in each of the chapters in this part of the thesis. In the first two chapters, we derive general results on the mean performance analysis for polling models. We study a model variation on the arrival processes in Chapter 10, and derive a closed-form approximation for the mean waiting times in Chapter 11. Then, we devote two chapters on the study of 'fairness and efficiency'. We try to achieve (almost) equal mean waiting times at each of the queues by introducing two new service disciplines: the κ-Gated discipline in Chapter 12, and the Gated/Exhaustive discipline in Chapter 13. We draw conclusions and discuss possibilities for further research at the end of each of the chapters separately.

9.4.1 Polling: Mean performance analysis

In a classical polling model, the customers arrive to each of the queues according to a Poisson process. In Chapter 10 we study a variation on this. Using the concept of so-called *smart customers*, the arrival rate in each queue varies depending on the position of the server. For example, the arrival rate might increase when the server is approaching the queue, whereas it is much lower when the server has just served the queue, as an arriving customer has to wait for almost an entire cycle in this case. For this model, we derive the mean waiting times using MVA, and show that in a few special cases it is possible to derive a PCL. Furthermore, we show the typical features of the model by an example.

Expressions for the mean waiting times in a polling system typically require cumbersome computations resulting in unmanageable expressions. The results from solving a system, like in the MVA approach, yield lengthy and complicated expressions, which are difficult to interpret. Also, numerical procedures to exactly or approximately derive the $\mathbb{E}[W_i]$'s are computationally complex and non-transparent. Consequently, there is a lack of insight into the mean waiting times in polling models, and the impact of parameters on these and the system performance in general. For this reason, we derive in Chapter 11 closed-form approximate solutions for the mean waiting times (and mean marginal queue lengths) in a polling model. The expressions can be computed by simple calculations. Also, it is very suitable for the design and optimization phase of (the application of) a polling model, as it provides insights in the system behavior when parameters are changed. In addition, in this chapter, we relax the assumption of Poisson arrivals to renewal arrival processes.

9.4.2 Polling: Fairness and efficiency

In certain applications it is important to maintain *fairness*, in the sense of the queues having (almost) equal mean waiting times. In [152, 185] this was motivated by a dynamic bandwidth allocation problem of Ethernet Passive Optical Networks (EPON). In achieving fairness, however, one usually has to sacrifice the efficiency of the system. To overcome this, we introduce two new service disciplines in Chapters 12 and 13, which achieve both fairness and efficiency. For this, we modify the ordinary gated and exhaustive strategies. We define fairness as the maximal difference in the mean waiting times at each of the queues (although other definitions exist), where we use the PCL (see Section 9.3.1) as a measure for the efficiency of the system. We then optimize a weighted sum of the fairness and the efficiency.

Fairness has frequently played a role in the choice of a service discipline in polling systems. In [152, 185], a *two-stage gated* service discipline is studied. This was seen to give rise to relatively small differences between mean waiting times at the various queues, but at the expense of longer delays, i.e., at the expense of the efficiency of the system. The strategy was later generalized to multi-phase gated (see [186]). Besides the two- and multi-stage gated disciplines, a number of other disciplines have been proposed in the literature in order to achieve fairness. Altman, Khamisy and Yechiali [3] (see also Shoham and Yechiali [167]) consider a so-called elevator strategy in a globally gated regime. In this setting the queues are visited in the order: $1, 2, \ldots N-1, N, N, N-1, \ldots 2, 1, 1, 2, \ldots$ etc. When the server turns around at queue 1 or queue N, a gate closes at all queues: only those before the gate are served. This strategy turns out to be *perfectly* fair. However, it is less efficient because of the globally gated regime. Moreover, our focus is on cyclic models. Other options for the control of a polling model proposed in the literature include the use of a polling table (introduced by [41], see also [17, 205]), which prescribes the order in which queues are visited. This is related to the ideas of efficient visit orders [39] and efficient visit frequencies [38]. These options, however, do not focus on fairness.

Both the exhaustive and gated discipline have advantages and disadvantages with respect to fairness and efficiency. The main advantage of the exhaustive strategy, is that it is optimally efficient. That is, it minimizes the PCL. However, the differences between mean waiting times at the queues might be large. Typically, the heaviest loaded queue has the smallest mean waiting time in this discipline. Conversely, the gated discipline leads in general to much smaller differences in mean waiting times. But this is at the expense of the efficiency, which is lower for this discipline. We aim to combine the best of both worlds into new service disciplines. We do so in Chapter 12 by introducing a hybrid version of exhaustive and gated: the *κ-gated service discipline*. In Chapter 13 we try to achieve this by introducing a combination of the gated and exhaustive discipline, by applying an alternating pattern of both strategies: the *Gated/Exhaustive discipline*. The κ-gated service discipline has a vector of parameters κ which can be used to optimize the strategy. In that way, it includes the purely exhaustive and the purely gated discipline as a special case. The main research question studied in both chapters, is whether the proposed strategies achieve (Chapter 13) or can be optimized to achieve (Chapter 12) a combination of both fairness and efficiency.

Polling: Mean performance analysis

10

Smart customers

In this chapter we study a polling model where the arrival rate at a queue depends on the position of the server, which we refer to as a polling model with *smart customers*. We derive the mean waiting times using MVA, and show that in a few special cases it is possible to derive a PCL. Furthermore, we show the typical features of the model in an example.

10.1 Introduction

In this chapter we consider the basic polling model as described in Chapter 9, with the distinguishing feature that the rates of the Poisson arrival processes at the various queues depend on the server location. This model was first considered by Boxma [32], who refers to this model as a polling model with *smart customers*, because one way to look at this system is to regard it as a queueing system where customers choose which queue to join, based on the current server position.

Allowing arrival rates to depend on the location of the server has practical relevance because in, e.g., certain production environments or traffic intersections, the arrival rates are influenced by the position of the server.

A relevant application can be found in [89], where a polling model is used to model a dynamic order picking system (DPS). In a DPS, a worker picks orders arriving in real time during the picking operations and the picking information can dynamically change in a picking cycle. One of the challenging questions that on-line retailers now face, is how to organize the logistic fulfillment processes during and after order receipt. In traditional stores, purchased products can be taken home immediately. However, in the case of on-line retailers, the customer must wait for the shipment to arrive. In order to reduce throughput times, an efficient enhancement to an ordinary DPS is to have products stored at multiple locations. The system can be modeled as a polling system with queues corresponding to each of the locations, and customers corresponding to orders. The location of the worker determines in which of the queues an order is being placed. In this system arrival rates of the orders depend on the location of the server (i.e. the worker), which makes it a typical smart customers example. A graphical illustration is given in Figure 10.1. We focus on one specific order type, which is placed in two locations, say Q_i and Q_j. While the picker is on its way to Q_i, say at location 1, all of

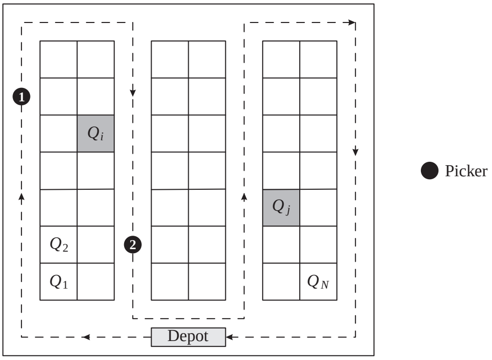


Figure 10.1: A dynamic order picking system. Orders are placed in queues $Q_1, \dots, Q_N$.

these orders are routed to Q_i and the arrival rate at Q_j is zero. If the picker is between Q_i and Q_j, say at location 2, the situation is reversed and Q_j receives all of these orders.

Besides practical relevance, the smart customers model also provides a powerful framework to analyze more complicated polling models. For example, a polling model where the service discipline switches each cycle between gated and exhaustive (the Gated/Exhaustive service discipline, see Chapter 13), can be analyzed constructing an alternative polling model with twice the number of queues and arrival rates being zero during specific visit periods. The same idea can be applied to analyze the κ-Gated service discipline, see Chapter 12. The idea of temporarily setting an arrival rate to zero is also used in [24] for the analysis of a polling model with multiple priority levels. Time varying arrival rates also play a role in the analysis of a polling model with reneging at polling instants [23].

Concerning state-dependent arrival rates, more literature is available for systems consisting of only one queue, often assuming phase-type distributions for vacations and/or service times. A system consisting of a single queue with server breakdowns and arrival rates depending on the server status is studied in [166]. A difference with the system studied in the present chapter, besides the number of queues, is that the machine can break down at arbitrary moments during the service of customers. Shanthikumar [164] discusses a stochastic decomposition for the queue length in an $M/G/1$ queue with server vacations under less restrictive assumptions than Fuhrmann and Cooper [84]. One of the relaxations is that the arrival rate of customers may be different during visit periods and vacations. Another system, with so-called working vacations and server breakdowns is studied in [107]. During these working vacations, both the service and arrival rates are different. Mean waiting times are found using a matrix analytical approach. For polling systems, a model with arrival rates that vary depending on the location of the server has not been studied in detail yet. Boxma [32] studies the joint queue length distribution at the beginning of a cycle, but no waiting times or marginal queue lengths are discussed. In a recent paper [36], a polling system with Lévy-driven, possibly correlated input is considered. Just as in the present chapter, the arrival process may depend on the location of the server. In [36] typical performance measures for Lévy processes are determined, such as the steady-state distribution of the joint amount of fluid at an arbitrary epoch, and at polling and switching instants. The present chapter studies a similar setting, but assumes Poisson arrivals of individual customers.

For polling models with smart customers, we provide the Mean Value Analysis for determining the mean queue lengths and mean waiting times. Also, we show that under certain conditions a Pseudo-Conservation Law for the total amount of work in the system holds. Finally, typical features of the model under consideration are demonstrated in a numerical example.

This chapter is structured as follows. In Section 10.2 we introduce the model and notation used. Then, in Section 10.3 we adapt the MVA framework for polling systems to smart customers. This results in a very efficient method to compute the mean waiting time of each customer type. In Section 10.4 we show that, under certain conditions, a PCL is satisfied by our model. We present a numerical example that illustrates some typical features and advantages of the model under consideration in Section 10.5. Finally, Appendix 10.A contains equations omitted from the main text. This chapter is based on [28], in which also distributions are derived for the joint queue lengths at polling instants, the marginal queue lengths, the cycle times, and the visit times.

10.2 Model and notation

We consider a polling model as described in Section 9.2, with one distinguishing feature, which is the arrival process. This arrival process is a standard Poisson process, but the rate depends on the location of the server. The arrival rate at Q_i is denoted by $\lambda_i^{(P)}$, where P denotes the position of the server, which is either serving a queue, or switching from one queue to the next: $P \in \{V_1, S_1, \ldots, V_N, S_N\}$. We refer to this as a polling model with smart customers. Throughout the sequel of this chapter, we focus on the exhaustive service discipline, although only minor changes are required for the gated discipline or for mixed service disciplines.

10.3 Mean Value Analysis

We extend the Mean Value Analysis (MVA) framework for polling models (see Section 9.3.2) in this section to suit the concept of smart customers. We first determine the mean visit times and the mean cycle time, and then present the MVA equations for a polling system with smart customers.

10.3.1 Mean visit times and mean cycle time

For the case of smart customers, the visit times to a queue depend on all arrival rates $\lambda_i^{(V_j)}$ and $\lambda_i^{(S_j)}$, $i, j = 1, \ldots, N$. In order to extend MVA to this case, we first derive the mean visit times to each of the queues, $\mathbb{E}[V_i]$, for $i = 1, \ldots, N$. We need these to determine the fraction of the time the server is switching to or visiting a certain queue. For this, we set up a system of N linear equations where the mean visit time of a queue is expressed in terms of the other mean visit times.

At the moment the server finishes serving Q_i, there are no type i customers present in the system any more. From this point on, the number of type i customers builds up at rates $\lambda_i^{(S_i)}, \lambda_i^{(V_{i+1})}, \ldots, \lambda_i^{(S_{i+N-1})}$ (depending on the position of the server), until the server starts working on Q_i again. Each of these customers initiates a busy period, with mean $\mathbb{E}[BP_i] = \mathbb{E}[B_i]/(1-\lambda_i^{(V_i)}\mathbb{E}[B_i])$. This gives:

$$\mathbb{E}[V_i] = \mathbb{E}[BP_i]\left(\lambda_i^{(S_i)}\mathbb{E}[S_i] + \sum_{j=i+1}^{i+N-1}\left(\lambda_i^{(V_j)}\mathbb{E}[V_j] + \lambda_i^{(S_j)}\mathbb{E}[S_j]\right)\right),$$

for $i = 1, \ldots, N$. The $\mathbb{E}[V_i]$ follow from solving this set of equations. Once these have been obtained, the mean cycle time follows from $\mathbb{E}[C] = \sum_{i=1}^{N}(\mathbb{E}[V_i] + \mathbb{E}[S_i])$.

10.3.2 MVA equations

We extend the MVA approach to polling systems with smart customers. First, we briefly introduce some extra notation, then we give expressions for the mean waiting times, and the mean conditional and unconditional queue lengths. After eliminating variables, we end up with a system of linear equations. The system can (numerically) be solved in order to find the unknowns, in particular, the mean unconditional queue lengths and the mean waiting times. Although all equations are discussed in the present section, for the sake of brevity of this section, some of them are presented in Appendix 10.A.

Denote by $\mathbb{E}[P]$ is the mean duration of a period $P \in \{V_1, S_1, \ldots, V_N, S_N\}$, and by $\mathbb{E}[P^2]$ its second moment. Then the mean residual duration of a period P, at an arbitrarily chosen point in this period, is $\mathbb{E}[P^{res}] = E[P^2]/(2\,\mathbb{E}[P])$. The fraction of time the system is in a given period P is denoted by $q^{(P)} := \mathbb{E}[P]/\mathbb{E}[C]$. The mean conditional number of type j customers in the queue during a random point in P is denoted by $\mathbb{E}[L_j^{(P)}]$, and the mean (unconditional) number of type j customers in queue is denoted by $\mathbb{E}[L_j]$ (both excluding a customer potentially in service).

We define an *interval* $(P_1 : P_2)$, $P_1, P_2 \in \{V_1, S_1, \ldots, V_N, S_N\}$, as the consecutive periods from period P_1 on, until and including period P_2. The mean residual duration of the interval is denoted by $\mathbb{E}[(P_1 : P_2)^{res}]$.

For the mean conditional durations of a period, we have the following: $\mathbb{E}[\overleftarrow{V_i}^{(V_j)}]$ denotes the mean duration of the *previous* period V_i, seen from an arbitrary point in V_j (i.e., V_i seen backward in time from the viewpoint of V_j), and $\mathbb{E}[\overrightarrow{V_i}^{(V_j)}]$ denotes the mean duration of the *next* period V_i (i.e., V_i seen forward in time from the viewpoint of V_j). For $i = j$ they both coincide, and represent the mean age, respectively the mean residual duration of V_i. Since the distribution of the age of a period is the same as the distribution of the residual period, we have $\mathbb{E}[\overleftarrow{V_i}^{(V_i)}] = \mathbb{E}[\overrightarrow{V_i}^{(V_i)}] = \mathbb{E}[V_i^{res}]$. Generally, however, $\mathbb{E}[\overleftarrow{V_i}^{(V_j)}] \neq \mathbb{E}[\overrightarrow{V_i}^{(V_j)}]$ for $i \neq j$, because of the dependencies between the durations of periods. Analogously, we define $\mathbb{E}[\overleftarrow{V_i}^{(S_j)}]$, $\mathbb{E}[\overrightarrow{V_i}^{(S_j)}]$, $\mathbb{E}[\overleftarrow{S_i}^{(V_j)}]$, and $\mathbb{E}[\overrightarrow{S_i}^{(V_j)}]$. Note that, e.g., $\mathbb{E}[\overrightarrow{S_i}^{(V_j)}] = \mathbb{E}[S_i]$, but $\mathbb{E}[\overleftarrow{S_i}^{(V_j)}] \neq \mathbb{E}[S_i]$. As switch-over times are independent, the following quantities directly simplify:

$$\mathbb{E}[\overleftarrow{S_i}^{(S_j)}] = \mathbb{E}[\overrightarrow{S_i}^{(S_j)}] = \begin{cases} \mathbb{E}[S_i] & \text{for } i \neq j, \\ \mathbb{E}[S_i^{res}] & \text{for } i = j. \end{cases}$$

Furthermore, we define

$$\overline{\lambda}_i = \frac{1}{\mathbb{E}[C]} \sum_{j=1}^{N} \left(\lambda_i^{(V_j)} \mathbb{E}[V_j] + \lambda_i^{(S_j)} \mathbb{E}[S_j] \right).$$

Then, the probability that the server is at position $P \in \{V_1, S_1, \ldots, V_N, S_N\}$ at the arrival of a type i customer, is

$$\frac{\lambda_i^{(P)} \mathbb{E}[P]}{\overline{\lambda}_i \mathbb{E}[C]} = \frac{q^{(P)} \lambda_i^{(P)}}{\overline{\lambda}_i}.$$

Having introduced the required notation, we now present the main theorem of this section, which gives a set of equations that can be solved to find the mean waiting times of customers in the system.

THEOREM 10.3.1. *The mean waiting times, $\mathbb{E}[W_i]$, for $i = 1, \ldots, N$, and the mean queue lengths, $\mathbb{E}[L_i]$, satisfy the following equations:*

$$\begin{aligned} \mathbb{E}[W_i] = {} & \frac{q^{(V_i)} \lambda_i^{(V_i)}}{\overline{\lambda}_i} \left(\mathbb{E}[L_i^{(V_i)}] \mathbb{E}[B_i] + \mathbb{E}[B_i^{res}] \right) \\ & + \sum_{j=i+1}^{i+N-1} \frac{q^{(V_j)} \lambda_i^{(V_j)}}{\overline{\lambda}_i} \left(\mathbb{E}[L_i^{(V_j)}] \mathbb{E}[B_i] + \sum_{k=j}^{i+N-1} \left(\mathbb{E}[S_k] + \mathbb{E}[\overrightarrow{V_k}^{(V_j)}] \right) \right) \\ & + \sum_{j=i}^{i+N-1} \frac{q^{(S_j)} \lambda_i^{(S_j)}}{\overline{\lambda}_i} \left(\mathbb{E}[L_i^{(S_j)}] \mathbb{E}[B_i] + \mathbb{E}[S_j^{res}] + \sum_{k=j+1}^{i+N-1} \left(\mathbb{E}[S_k] + \mathbb{E}[\overrightarrow{V_k}^{(S_j)}] \right) \right), \end{aligned} \tag{10.3.1}$$

$$\mathbb{E}[L_i] = \overline{\lambda}_i \mathbb{E}[W_i], \tag{10.3.2}$$

$$\mathbb{E}[L_i] = \sum_{j=i+1}^{i+N} \left(q^{(V_j)} \mathbb{E}[L_i^{(V_j)}] + q^{(S_j)} \mathbb{E}[L_i^{(S_j)}] \right), \tag{10.3.3}$$

where the conditional mean queue lengths $\mathbb{E}[L_i^{(V_j)}]$ and $\mathbb{E}[L_i^{(S_j)}]$, for $j = i+1, \ldots, i+N-1$, are given by

$$\mathbb{E}[L_i^{(V_j)}] = \sum_{k=i+1}^{j} \lambda_i^{(V_k)} \mathbb{E}[\overleftarrow{V_k}^{(V_j)}] + \sum_{k=i}^{j-1} \lambda_i^{(S_k)} \mathbb{E}[\overleftarrow{S_k}^{(V_j)}], \tag{10.3.4}$$

$$\mathbb{E}[L_i^{(S_j)}] = \sum_{k=i+1}^{j} \lambda_i^{(V_k)} \mathbb{E}[\overleftarrow{V_k}^{(S_j)}] + \sum_{k=i}^{j} \lambda_i^{(S_k)} \mathbb{E}[\overleftarrow{S_k}^{(S_j)}], \tag{10.3.5}$$

and where all $\mathbb{E}[\overleftarrow{P_1}^{(P_2)}]$ and $\mathbb{E}[\overrightarrow{P_1}^{(P_2)}]$, for $P_1, P_2 \in \{V_1, S_1, \ldots, V_N, S_N\}$, satisfy the set of equations (10.3.6) – (10.3.8) *below, and* (10.A.2)–(10.A.7) *in Appendix 10.A.*

PROOF. In order to derive the mean waiting time $\mathbb{E}[W_i]$, we condition on the period in which a type i customer arrives. A fraction $q^{(V_j)} \lambda_i^{(V_j)} / \overline{\lambda}_i$, and $q^{(S_j)} \lambda_i^{(S_j)} / \overline{\lambda}_i$ respectively, of the type i customers arrives during period V_j, and during period S_j respectively. If a tagged type i customer arrives during period V_i (i.e., while his queue is being served), he has to wait for a residual service time, plus the service times of all type i customers present in the system upon his arrival, which is (by conditional PASTA), $\mathbb{E}[L_i^{(V_i)}]$. As a fraction $q^{(V_i)}\lambda_i^{(V_i)}/\overline{\lambda}_i$ of the customers arrives during V_i, this explains the first line of (10.3.1). If the customer arrives in any other period, he has to wait until the server returns to Q_i again. For this, we condition on the period in which he arrives. If the arrival period is a visit to Q_j, say V_j for $j \neq i$, he has to wait for the residual duration of V_j and the interval $(S_j : S_{i-1})$, and for the service of the type i customers present in the system upon his arrival. This gives the second line of (10.3.1). The third line, the case where the customer arrives during the switch-over time from Q_j to Q_{j+1} (period S_j), can be interpreted along the same lines as the case V_j.

Equation (10.3.3) is obtained by unconditioning the conditional queue lengths $\mathbb{E}[L_i^{(P)}]$. The mean number of type i customers in the queue at an arbitrary point during V_j, given by (10.3.4), is the mean number of customers built up from the last visit to Q_i (when Q_i became empty) until and including a residual duration of V_j (as the mean residual duration of V_j is equal to the mean age of that period), taking into account the varying arrival rates. The mean number of type i customers queueing in the system during period S_j, given by (10.3.5), can be found similarly. Equations (10.3.4) and (10.3.5) show one of the difficulties in adapting the 'ordinary' MVA approach to that of smart customers. If the arrival rates remain constant during a cycle, these expressions would reduce to λ_i multiplied by the mean time passed since the server has left Q_i. However, for the smart customers case, we have to keep track of the duration of all the intermediate periods, from the viewpoint of period V_j respectively S_j.

As indicated in Theorem 10.3.1, at this point, the number of equations is insufficient to find all the unknowns, $\mathbb{E}[\overleftarrow{P_1}^{(P_2)}]$ and $\mathbb{E}[\overrightarrow{P_1}^{(P_2)}]$, for $P_1, P_2 \in \{V_1, S_1, \dots, V_N, S_N\}$. In the remainder of the proof, we develop additional relations for these quantities to complete the set of equations. We start by considering $\mathbb{E}[\overrightarrow{V_i}^{(V_j)}]$, which is the mean duration of the next period V_i, when observed from an arbitrary point in V_j. For $i = j$ this is just the residual duration of V_i, consisting of a busy period induced by a customer with a residual service time left, and the busy periods of all type i customers in the queue. The cases $i \neq j$ need some more attention. The duration of V_i now consists of the busy period induced by the type i customers in the queue, which are in expectation $\mathbb{E}[L_i^{(V_j)}]$ customers. During the periods $V_j, S_j, \dots, S_{i-1}$, however, new type i customers are arriving, each contributing a busy period to the duration of V_i. Hence, summing over these periods and taking into account the varying arrival rates, we get the mean total of newly arriving customers in this interval. This yields, for $i = 1, \dots, N$ and $j = i+1, \dots, i+N-1$:

$$\mathbb{E}[\overrightarrow{V_i}^{(V_i)}] = \mathbb{E}[BP_i]\,\mathbb{E}[L_i^{(V_i)}] + \mathbb{E}[B_i^{res}] / \left(1 - \lambda_i^{(V_i)}\mathbb{E}[B_i]\right), \tag{10.3.6}$$

$$\mathbb{E}[\overrightarrow{V_i}^{(V_j)}] = \mathbb{E}[BP_i]\left(\mathbb{E}[L_i^{(V_j)}] + \sum_{k=j}^{i+N-1}\left(\lambda_i^{(V_k)}\mathbb{E}[\overrightarrow{V_k}^{(V_j)}] + \lambda_i^{(S_k)}\mathbb{E}[S_k]\right)\right). \tag{10.3.7}$$

Analogously, $\mathbb{E}[\overrightarrow{V_i}^{(S_j)}]$ denotes the mean duration of the next period V_i, when observed from an arbitrary point in S_j. The explanation of its expression is along the same lines as that of $\mathbb{E}[\overrightarrow{V_i}^{(V_j)}]$, although it should be noted that $i = j$ is not a special case. See (10.A.1) in Appendix 10.A.

The last step in the proof of Theorem 10.3.1 needs the following lemma to find the final relations between $\mathbb{E}[\overleftarrow{P_1}^{(P_2)}]$ and $\mathbb{E}[\overrightarrow{P_1}^{(P_2)}]$:

LEMMA 10.3.2. *For $i = 1,\ldots,N$, and $j = i+1,\ldots,i+N$:*

$$\begin{aligned}\sum_{k=i}^{j-1} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(S_i:V_j)]} \Bigg(& \mathbb{E}[\overleftarrow{S_i}^{(S_k)}] + \sum_{l=i+1}^{k} \left(\mathbb{E}[\overleftarrow{S_l}^{(S_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(S_k)}] \right) \\ & - \mathbb{E}[S_k{}^{res}] - \mathbb{E}[\overrightarrow{V_j}^{(S_k)}] - \sum_{l=k+1}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(S_k)}] \right) \Bigg) \\ = \sum_{k=i+1}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(S_i:V_j)]} \Bigg(& \mathbb{E}[\overrightarrow{V_j}^{(V_k)}] + \sum_{l=k}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(V_k)}] \right) \\ & - \mathbb{E}[\overleftarrow{S_i}^{(V_k)}] - \mathbb{E}[\overleftarrow{V_k}^{(V_k)}] - \sum_{l=i+1}^{k-1} \left(\mathbb{E}[\overleftarrow{S_l}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(V_k)}] \right) \Bigg). \end{aligned} \tag{10.3.8}$$

PROOF. Equation (10.3.8) can be proven by studying all mean residual interval lengths $\mathbb{E}[(S_i:V_j)^{res}]$, $\mathbb{E}[(S_i:S_j)^{res}]$, $\mathbb{E}[(V_i:V_j)^{res}]$, and $\mathbb{E}[(V_i:S_j)^{res}]$. Consider $\mathbb{E}[(S_i:V_j)^{res}]$, the mean residual duration of the interval $S_i, V_{i+1}, \ldots, V_j$. We condition on the period in which the interval is observed. As the mean duration of the interval is given by $\mathbb{E}[(S_i:V_j)]$, it follows that $\mathbb{E}[S_k]/\mathbb{E}[(S_i:V_j)]$ is the probability that the interval is observed in period S_k. The remaining duration of the interval consists of the remaining duration of S_k plus the mean durations of the (coming) periods $V_{k+1}, S_{k+1}, \ldots, V_j$, when observed from period S_k. When observing $\mathbb{E}[(S_i:V_j)]$ from V_k, a similar way of reasoning is used. This gives, for $i = 1,\ldots,N$, and $j = i+1,\ldots,i+N$:

$$\begin{aligned}\mathbb{E}[(S_i:V_j)^{res}] = & \sum_{k=i}^{j-1} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(S_i:V_j)]} \left(\mathbb{E}[S_k{}^{res}] + \mathbb{E}[\overrightarrow{V_j}^{(S_k)}] + \sum_{l=k+1}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(S_k)}] \right) \right) \\ & + \sum_{k=i+1}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(S_i:V_j)]} \left(\mathbb{E}[\overrightarrow{V_j}^{(V_k)}] + \sum_{l=k}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(V_k)}] \right) \right). \end{aligned} \tag{10.3.9}$$

We now use that the distribution of the residual length of an interval is the same as the distribution of the age of this interval. Again, focus on $\mathbb{E}[(S_i:V_j)^{res}]$, conditioning on the period in which the interval is observed, but now looking forward in time. Consider all the periods in $(S_i:V_j)$ that have already passed when observing during S_k. The interval has lasted for the sum of these periods, plus the age of S_k. The same can be done for an arbitrary point in V_k. This gives, for $i = 1,\ldots,N$, $j = i+1,\ldots,i+N$:

$$\begin{aligned}\mathbb{E}[(S_i:V_j)^{res}] = & \sum_{k=i}^{j-1} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(S_i:V_j)]} \left(\mathbb{E}[\overleftarrow{S_i}^{(S_k)}] + \sum_{l=i+1}^{k} \left(\mathbb{E}[\overleftarrow{S_l}^{(S_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(S_k)}] \right) \right) \\ & + \sum_{k=i+1}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(S_i:V_j)]} \left(\mathbb{E}[\overleftarrow{S_i}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_k}^{(V_k)}] + \sum_{l=i+1}^{k-1} \left(\mathbb{E}[\overleftarrow{S_l}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(V_k)}] \right) \right). \end{aligned} \tag{10.3.10}$$

The proof of Lemma 10.3.2 is completed by equating (10.3.9) and (10.3.10) and rearranging the terms. □

Similar to the proof of Lemma 10.3.2, we can develop two different expressions for each of the terms $\mathbb{E}[(S_i:S_j)^{res}], \mathbb{E}[(V_i:V_j)^{res}]$, and $\mathbb{E}[(V_i:S_j)^{res}]$. For the sake of brevity of this section, they are presented in Appendix 10.A, Equations (10.A.2)–(10.A.7). Equating each pair of these expressions, completes the set of (linear) equations for the mean waiting times and mean queue lengths. This concludes the proof of Theorem 10.3.1. □

10.4 Pseudo-Conservation Law

In this section we derive a Pseudo-Conservation Law (PCL), which gives an expression for the weighted sum of the mean waiting times at each of the queues. For 'ordinary' cyclic polling systems, Boxma and Groenendijk [34] derive a PCL under various service disciplines (see Section 9.3.1):

$$\sum_{i=1}^{N}\rho_i\mathbb{E}[W_i] = \frac{\rho}{1-\rho}\sum_{i=1}^{N}\rho_i\mathbb{E}[B_i^{res}] + \rho\mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)}\left(\rho^2 - \sum_{i=1}^{N}\rho_i^2\right) + \sum_{i=1}^{N}\mathbb{E}[M_i]. \tag{10.4.1}$$

A required restriction for our approach in this section, is that the Poisson process according to which work arrives in the system, has a fixed arrival rate during all *visit periods*. We also require that the amounts of work brought by an individual arrival are identically distributed for all visit periods. We mention two typical cases where this requirement is satisfied. Firstly, the case when the arrival rate at a given queue stays constant during different *visit* times, and secondly when the *total* arrival rate remains constant during visit times *and* the service times are identically distributed:

$$\text{Case 1:} \qquad \lambda_i^{(V_1)} = \lambda_i^{(V_2)} = \ldots = \lambda_i^{(V_N)} =: \lambda_i^{(V)}, \qquad i = 1,\ldots,N, \tag{10.4.2}$$

$$\text{Case 2:} \qquad \sum_{i=1}^{N}\lambda_i^{(V_j)} =: \Lambda^{(V)}, \text{ and } B_1 \stackrel{d}{=} \ldots \stackrel{d}{=} B_N, \qquad j = 1,\ldots,N, \tag{10.4.3}$$

denoting by $\stackrel{d}{=}$ equality in distribution. During visit periods, let $\Lambda^{(V)}$ be the total arrival rate of all customer types, and let $B^{(V)}$ denote the generic service time of an arbitrary customer entering the system. In particular, this means for Case 1 that $\Lambda^{(V)} = \sum_{i=1}^{N}\lambda_i^{(V)}$ and $B^{(V)} \stackrel{d}{=} B_i$ with probability $\lambda_i^{(V)}/\Lambda^{(V)}$ for $i = 1,\ldots,N$. We introduce $\rho^{(V)}$ to denote the mean amount of work entering the system per time unit during a visit period, so $\rho^{(V)} = \Lambda^{(V)}\mathbb{E}[B^{(V)}]$.

For deriving the PCL, we focus on the mean amount of *work* in the system at an arbitrary point in time. Denote this by Y, and let $Y^{(V)}$ and $Y^{(S)}$ be the amount of work at an arbitrary point during respectively a visit period, and a switch-over period. Then

$$Y \stackrel{d}{=} \begin{cases} Y^{(V)} & \text{w.p. } \overline{\rho}, \\ Y^{(S)} & \text{w.p. } 1-\overline{\rho}, \end{cases} \tag{10.4.4}$$

where $\overline{\rho} := \sum_{i=1}^{N}\overline{\rho}_i = \sum_{i=1}^{N}\overline{\lambda}_i\mathbb{E}[B_i]$ is the mean amount of work offered per time unit. Hence,

$$\mathbb{E}[Y] = \overline{\rho}\,\mathbb{E}[Y^{(V)}] + (1-\overline{\rho})\mathbb{E}[Y^{(S)}]. \tag{10.4.5}$$

Another way to obtain the mean total amount of work in the system, is by taking the sum of the mean workloads. The mean workload in Q_i is the mean amount of work of all customers in the queue, plus, with probability $\overline{\rho}_i = \overline{\lambda}_i \mathbb{E}[B_i]$, the mean remaining amount of work of a customer in service at Q_i:

$$\mathbb{E}[Y] = \sum_{i=1}^{N} \left(\mathbb{E}[L_i]\mathbb{E}[B_i] + \overline{\rho}_i \mathbb{E}[B_i^{res}] \right). \tag{10.4.6}$$

In the next subsections we show that equating (10.4.5) and (10.4.6), and applying Little's Law, $\mathbb{E}[L_i] = \overline{\lambda}_i \mathbb{E}[W_i]$, gives a PCL for the mean waiting times in the system. However, we first derive $\mathbb{E}[Y^{(S)}]$ and $\mathbb{E}[Y^{(V)}]$.

10.4.1 Work during switch-over periods

The term $\mathbb{E}[Y^{(S)}]$ denotes the mean amount of work in the system when observed at a random point in a switch-over interval. Denoting by $\mathbb{E}[Y^{(S_i)}]$ the mean amount of work in the system at an arbitrary moment during S_i, we can condition on the switch-over interval in which the system is observed:

$$\mathbb{E}[Y^{(S)}] = \sum_{i=1}^{N} \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]} \mathbb{E}[Y^{(S_i)}]. \tag{10.4.7}$$

We can split $\mathbb{E}[Y^{(S_i)}]$ into two parts: the mean amount of work present at the start of S_i, plus the mean amount of work built up since the start of the switch-over time. In expectation, a duration $\mathbb{E}[S_i^{res}]$ has passed since the beginning of the switch-over time, in which work arrived at rate $\lambda_j^{(S_i)} \mathbb{E}[B_j]$ at Q_j. Hence, this gives a contribution to $\mathbb{E}[Y^{(S_i)}]$ of $\sum_{j=1}^{N} \lambda_j^{(S_i)} \mathbb{E}[B_j]\mathbb{E}[S_i^{res}]$. Denote by M_j the amount of work, present at Q_j at the end of the visit to this queue. For the work present at the start of the switch-over period, we start looking at the moment that the server left Q_j, leaving a mean amount of work $\mathbb{E}[M_j]$ behind in this queue. For exhaustive service, $\mathbb{E}[M_j] = 0$, for gated service $\mathbb{E}[M_j] = \lambda_j^{(V_j)} \mathbb{E}[B_j]\mathbb{E}[V_j]$. Since then, the interval $(S_j : V_{i+N})$ has passed, for $j = i+1, \ldots, i+N-1$. In this interval the amount of type j work increased at rates $\lambda_j^{(S_j)} \mathbb{E}[B_j], \lambda_j^{(V_{j+1})} \mathbb{E}[B_j], \ldots, \lambda_j^{(S_{i-1})} \mathbb{E}[B_j], \lambda_j^{(V_i)} \mathbb{E}[B_j]$ during the various periods. This leads to the following expression for $\mathbb{E}[Y^{(S_i)}]$:

$$\begin{aligned} \mathbb{E}[Y^{(S_i)}] = &\sum_{j=1}^{N} \left(\lambda_j^{(S_i)} \mathbb{E}[B_j]\mathbb{E}[S_i^{res}] + \mathbb{E}[M_j] \right) \\ &+ \sum_{j=i+1}^{i+N-1} \sum_{k=j}^{i+N-1} \left(\lambda_j^{(S_k)} \mathbb{E}[B_j]\mathbb{E}[S_k] + \lambda_j^{(V_{k+1})} \mathbb{E}[B_j]\mathbb{E}[V_{k+1}] \right). \end{aligned} \tag{10.4.8}$$

10.4.2 Work during visit periods

For obtaining $\mathbb{E}[Y^{(V)}]$, we follow the proof of the PCL as in [34]. The key observation in this proof is the *work decomposition* property in a polling system. This property states that the amount of work at an arbitrary epoch in a visit period is distributed as the sum of two independent random variables: the amount of work in the "corresponding" $M/G/1$ queue at an arbitrary epoch during a busy period, denoted by $Y^{(V)}_{M/G/1}$, and the amount of work in the polling system at an arbitrary epoch during a switch-over time, $Y^{(S)}$. In a polling model with smart customers, this decomposition does not typically hold, but a minor adaptation is required. We follow the proof in [34] as closely as possible, meaning that we use the concepts of ancestral line and offspring of a customer, as introduced in [84]. We also copy the idea of comparing the polling system to an $M/G/1$ queue with vacations and Last-Come-First-Served (LCFS) service. The traffic process offered to this $M/G/1$ queue is identical to the traffic process of the polling system. The server of the $M/G/1$ queue takes vacations exactly during the switching periods of the polling system. These vacations might interrupt the service of a customer in the $M/G/1$ queue. This service is not resumed until all customers that have arrived during the vacation and their offspring have been served (in LCFS order).

We now focus on the amount of work in this $M/G/1$ system at an arbitrary moment *during a visit (busy) period*. Let K be the customer being served at this observation moment, and let K_A be his ancestor. By definition, K_A has arrived during a vacation period (i.e. switch-over period in the corresponding polling system). Denote by Y_{K_A} the amount of work present in the system at the moment that K_A enters the system. An important difference with the situation studied in [34] is that we *cannot* use the PASTA property, so in general $Y_{K_A} \neq Y^{(S)}$. We now condition on the customer type of K_A. The mean duration of the service of a type i ancestor and his entire ancestral line is $\mathbb{E}[B_i]/(1-\rho^{(V)})$. This can be regarded as the mean busy period commencing with the service of an exceptional first customer (namely a type i customer). Each type i customer arriving during S_j, with arrival rate $\lambda_i^{(S_j)}$, $i, j = 1, \dots, N$, starts such a busy period, so the probability that K_A is a type i customer is:

$$p_i = \frac{\sum_{j=1}^N \lambda_i^{(S_j)} \mathbb{E}[S_j]\mathbb{E}[B_i]/(1-\rho^{(V)})}{\sum_{k=1}^N \sum_{j=1}^N \lambda_k^{(S_j)} \mathbb{E}[S_j]\mathbb{E}[B_k]/(1-\rho^{(V)})} = \frac{\sum_{j=1}^N \lambda_i^{(S_j)} \mathbb{E}[S_j]\mathbb{E}[B_i]}{\sum_{k=1}^N \sum_{j=1}^N \lambda_k^{(S_j)} \mathbb{E}[S_j]\mathbb{E}[B_k]}. \quad (10.4.9)$$

Given that K_A is a type i customer, we again pick up the proof of the work decomposition in [34]. Denote by B_{K_A} the service requirement of K_A. Then, because of the LCFS service discipline of the $M/G/1$ queue, the amount of work when K_A goes into service is exactly $Y_{K_A} + B_{K_A}$, and the amount of work when the last descendant of K_A has been served equals Y_{K_A} again (for the first time, since the arrival of K_A). Ignoring the amount of work present at K_A's arrival, the residual amount of work evolves just as during a busy period in an $M/G/1$ queue with an exceptional first customer (having generic service requirement B_i). The only exception is caused by the vacations (i.e. switch-over times in the polling model), during which the work remains constant or may increase because of new arrivals. However, just as in [34], if we ignore these vacations and the (LCFS) service of the ancestral lines of the customers that arrive during these vacations, what remains is the workload process during a busy period initiated by a type i customer. Denote by $Y^{(V)}_{M/G/1|i}$ the amount of work at an arbitrary moment during this busy period, and denote by $Y^{(S)}_{A_i}$ the amount of work present in the polling system at an arbitrary *arrival epoch* of a type i customer *during a switch-over time*. Note that Y_{K_A} is distributed like $Y^{(S)}_{A_i}$. Then we have the following decomposition:

$$Y^{(V)} \stackrel{d}{=} Y^{(V)}_{M/G/1|i} + Y^{(S)}_{A_i} \qquad \text{w.p. } p_i, \qquad i = 1, \dots, N, \tag{10.4.10}$$

with p_i as given in (10.4.9), and $Y^{(V)}_{M/G/1|i}$ and $Y^{(S)}_{A_i}$ being independent. This leads to

$$\mathbb{E}[Y^{(V)}] = \sum_{i=1}^{N} p_i \left(\mathbb{E}[Y^{(V)}_{M/G/1|i}] + \mathbb{E}[Y^{(S)}_{A_i}] \right), \tag{10.4.11}$$

with

$$\mathbb{E}[Y^{(V)}_{M/G/1|i}] = \mathbb{E}[B_i^{res}] + \frac{\rho^{(V)}}{1-\rho^{(V)}} \mathbb{E}\left[\left(B^{(V)}\right)^{res} \right], \tag{10.4.12}$$

$$\mathbb{E}[Y^{(S)}_{A_i}] = \sum_{j=1}^{N} \frac{\lambda_i^{(S_j)} \mathbb{E}[S_j]}{\sum_{k=1}^{N} \lambda_i^{(S_k)} \mathbb{E}[S_k]} \mathbb{E}[Y^{(S_j)}]. \tag{10.4.13}$$

For (10.4.12) we use standard theory on an $M/G/1$ queue with an exceptional first customer (cf. [211]), and (10.4.13) is established by conditioning on the switch-over period in which a type i customer arrives.

10.4.3 PCL for smart customers

We are now ready to state the PCL.

THEOREM 10.4.1. *Provided that* (10.4.2) *or* (10.4.3) *is valid, the following Pseudo-Conservation Law holds:*

$$\sum_{i=1}^{N} \overline{\rho}_i \, \mathbb{E}[W_i] = (1-\overline{\rho}) \sum_{i=1}^{N} \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]} \mathbb{E}[Y^{(S_i)}] - \sum_{i=1}^{N} \overline{\rho}_i \mathbb{E}[B_i^{res}]$$
$$+ \overline{\rho} \sum_{i=1}^{N} p_i \left(\sum_{j=1}^{N} \frac{\lambda_i^{(S_j)} \mathbb{E}[S_j]}{\sum_{k=1}^{N} \lambda_i^{(S_k)} \mathbb{E}[S_k]} \mathbb{E}[Y^{(S_j)}] + \mathbb{E}[B_i^{res}] + \frac{\rho^{(V)}}{1-\rho^{(V)}} \mathbb{E}\left[\left(B^{(V)}\right)^{res} \right] \right), \tag{10.4.14}$$

where $\mathbb{E}[Y^{(S_i)}]$ *are as in* (10.4.8)*, and the* p_i *as in* (10.4.9).

PROOF. We have two equations, (10.4.5) and (10.4.6), for the mean total amount of work in the system. Combining these two equations, and plugging in (10.4.7) and (10.4.11), we find

$$\sum_{i=1}^{N} \left(\mathbb{E}[L_i]\mathbb{E}[B_i] + \overline{\rho}_i \mathbb{E}[B_i^{res}] \right)$$
$$= (1-\overline{\rho}) \sum_{j=1}^{N} \frac{\mathbb{E}[S_j]}{\mathbb{E}[S]} \mathbb{E}[Y^{(S_j)}] \overline{\rho} \sum_{i=1}^{N} p_i \left(\mathbb{E}[Y^{(V)}_{M/G/1|i}] + \mathbb{E}[Y^{(S)}_{A_i}] \right).$$

By application of Little's Law, $\mathbb{E}[L_i] = \overline{\lambda}_i \mathbb{E}[W_i]$, using that $\overline{\rho}_i = \overline{\lambda}_i \mathbb{E}[B_i]$, plugging in (10.4.12) and (10.4.13), after some rewriting we obtain (10.4.14), which is a PCL for a polling model with smart customers. □

REMARK 10.4.2. When $\lambda_i^{(S_1)} = \lambda_i^{(S_2)} = \ldots = \lambda_i^{(S_N)} = \lambda_i^{(V_1)} = \cdots = \lambda_i^{(V_N)} = \lambda_i$, for all $i = 1, \ldots, N$, Equation (10.4.14) reduces to the standard PCL (10.4.1). E.g., because of PASTA, $\mathbb{E}[Y_{A_i}^{(S)}] = \mathbb{E}[Y^{(S)}]$, and $p_i = \lambda_i/\Lambda$ for all i.

Case 2, where the assumptions of (10.4.3) hold, has a nice practical interpretation if we add the additional requirement that $\sum_{i=1}^N \lambda_i^{(S_j)} = \sum_{i=1}^N \lambda_i^{(V_j)} =: \Lambda$ for all $j = 1, \ldots, N$. Now, the model can be interpreted as a polling system with customers arriving in one Poisson stream with constant arrival rate Λ, and generic service requirement B, but joining a certain queue with a fixed probability that may depend on the location of the server at the arrival epoch. In Section 10.5, we discuss an example on how these probabilities may be chosen to minimize the mean waiting time of an arbitrary customer. The PCL (10.4.14) can be simplified considerably in this situation.

COROLLARY 10.4.3. *If* (10.4.3) *is valid, the PCL* (10.4.14) *reduces to:*

$$\sum_{i=1}^N \overline{\rho}_i \, \mathbb{E}[W_i] = \sum_{i=1}^N \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]} \mathbb{E}[Y^{(S_i)}] + \frac{\rho^2}{1-\rho} \mathbb{E}[B^{res}]. \tag{10.4.15}$$

PROOF. This is a direct consequence of assumptions (10.4.3). E.g., in the computation of (10.4.12) there is no need to condition on a special first customer, and hence the term $\mathbb{E}[Y_{M/G/1|i}]$ does not depend on i anymore:

$$\mathbb{E}[Y_{M/G/1|i}] = \frac{\mathbb{E}[B^{res}]}{1-\rho},$$

where $\rho = \Lambda\mathbb{E}[B]$. Additionally, the term $\sum_{i=1}^N p_i \mathbb{E}[Y_{A_i}^{(S)}]$ also simplifies considerably:

$$\sum_{i=1}^N p_i \mathbb{E}[Y_{A_i}^{(S)}] = \sum_{i=1}^N \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]} \mathbb{E}[Y^{(S_i)}].$$

Combining this, multiple terms cancel out and (10.4.15) follows. It is easily seen that (10.4.15) is in line with the standard PCL (10.4.1), when the arrival rates do not change during various visit and switch-over times. $\square$

10.5 Numerical example

We consider a polling system where arriving customers choose which queue they join, based on the current position of the server. In [32, 37] a fully symmetric case is studied with gated service, and it is proven that the mean sojourn time of customers is minimized if customers join the queue that is being served directly after the queue that is currently being served. Although the exhaustive case is not studied, it is intuitively clear that in this situation smart customers join the queue that is currently being served. Or, in case an arrival takes place during a switch-over time, join the next queue that is visited. In this example, we study this situation in more detail by adding an extra parameter that can be varied. The polling model is fully symmetric, except for the service time of customers in Q_1, which is varied. The practical interpretation is the following: customers arrive with a fixed arrival intensity, say Λ, and choose which queue they join. This does not affect their service time, except when they choose Q_1. In this case, the service time has a different distribution. To illustrate the dynamics of this system, we choose the following setting. The system consists of three queues with exhaustive service. The switch-over times are all exponentially distributed with mean 1. The service times are also exponentially distributed with $\mathbb{E}[B_2] = \mathbb{E}[B_3] = 1$, and $\mathbb{E}[B_1]$ is varied between 0 and 2. Arriving customers choose one queue which they want to join. This queue is the same for all customers, so there is no randomness involved in the selection, which is only based on the location of the server at their arrival epochs. We intend to find the optimal queue for customers to join. In terms of the model parameters: we seek to find values for $\lambda_i^{(V_j)}$ and $\lambda_i^{(S_j)}$, $i, j = 1, 2, 3$, that minimize the mean sojourn time of an arbitrary customer, under the restriction that for each value of j, exactly one $\lambda_i^{(V_j)}$ and exactly one $\lambda_i^{(S_j)}$ is equal to Λ, and all the other values are 0. A valid combination of these arrival intensities is called a *strategy*, and we introduce the short notation for a strategy by the indices of the queues that are joined in respectively $(V_1, S_1, V_2, S_2, V_3, S_3)$. E.g., for the fully symmetric case, with $\mathbb{E}[B_1] = 1$, it is intuitively clear that the optimal strategy is to join Q_i, if the arrival takes place during V_i, and to join Q_{i+1} if the arrival takes place during S_i. This strategy is denoted by $(1, 2, 2, 3, 3, 1)$, and corresponds to $\lambda_1^{(V_1)} = \lambda_2^{(V_2)} = \lambda_3^{(V_3)} = \Lambda$, and $\lambda_2^{(S_1)} = \lambda_3^{(S_2)} = \lambda_1^{(S_3)} = \Lambda$. The other arrival intensities are 0. As stated before, we vary $\mathbb{E}[B_1]$ between 0 and 2, and focus on the overall mean sojourn time. It is clear that making $\mathbb{E}[B_1]$ smaller, makes it more attractive to join Q_1 (even if another queue is served), whereas making $\mathbb{E}[B_1]$ larger, makes it less attractive to join Q_1. In order to obtain numerical results, we choose the (arbitrary) value $\Lambda = \frac{3}{5}$. It turns out that as much as *seven* different strategies can be optimal, depending on the value of $\mathbb{E}[B_1]$. We refer to these strategies as I through VII, listed in Table 10.1, along with their region of optimality. For each of these strategies, the mean sojourn time of an arbitrary customer is plotted versus $\mathbb{E}[B_1]$ in Figure 10.2.

Strategy	Queue to join during						Region of optimality
	V_1	S_1	V_2	S_2	V_3	S_3	
I	1	1	X	1	X	1	$0.00 \leq \mathbb{E}[B_1] \leq 0.41$
II	1	2	1	1	X	1	$0.41 \leq \mathbb{E}[B_1] \leq 0.66$
III	1	2	2	1	X	1	$0.66 \leq \mathbb{E}[B_1] \leq 0.73$
IV	1	2	2	3	1	1	$0.73 \leq \mathbb{E}[B_1] \leq 0.84$
V	1	2	2	3	3	1	$0.84 \leq \mathbb{E}[B_1] \leq 1.10$
VI	2	2	2	3	3	1	$1.10 \leq \mathbb{E}[B_1] \leq 1.16$
VII	X	2	2	3	3	2	$1.16 \leq \mathbb{E}[B_1]$

Table 10.1: The seven smartest strategies that minimize the mean waiting time of an arbitrary customer who can choose the queue in which he wants to be served. An 'X' means that the length of the corresponding visit time equals 0 because customers never join this queue.

As expected, Q_1 is most popular if $\mathbb{E}[B_1]$ is very small. In particular, for very small values of $\mathbb{E}[B_1]$, customers *always* join this queue (Strategy I). As $\mathbb{E}[B_1]$ becomes larger, Q_2 and later also Q_3 are chosen in more and more situations (Strategies II–V). Strategy V, which is optimal if the system is (nearly) symmetric, is the one where customers join the queue that is being served, or is going to be served next if the arrival takes place during

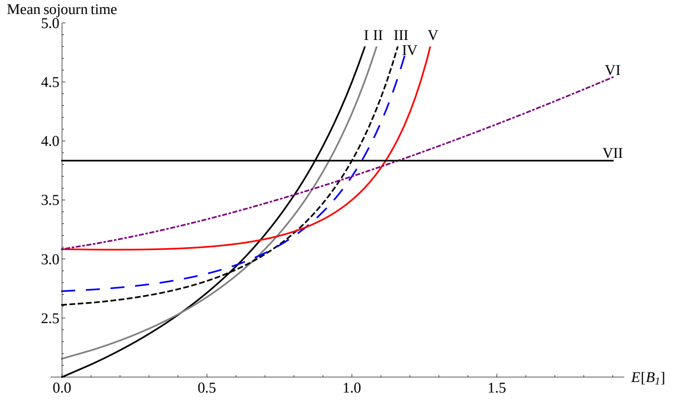


Figure 10.2: The mean sojourn time of an arbitrary customer for the seven smartest strategies, against the mean service time in Q_1.

a switch-over time. Strategy VI, which is optimal in only a very small range of values of $\mathbb{E}[B_1]$, states that customers only join Q_1 during the switch-over time S_3. Strategy VII, in which customers never join Q_1, is optimal for large values of $\mathbb{E}[B_1]$. The ergodicity constraint, considering all parameters are fixed except for $\mathbb{E}[B_1]$, for the different strategies is interesting to mention. For strategies I-V, the necessary and sufficient condition for stability is $\mathbb{E}[B_1] < \frac{5}{3}$. Strategies VI and VII always result in a stable system, regardless of $\mathbb{E}[B_1]$.

It is also interesting to discuss what *stupid customers* would do in this system. Stupid customers choose the worst possible strategy, in order to maximize the mean sojourn time of an arbitrary customer. We do not go into details and do not mention exactly which strategy is worst for each value of $\mathbb{E}[B_1]$, but we pick out some interesting cases. Obviously, when $\mathbb{E}[B_1] = 0$, the worst possible thing to do is never to join Q_1. The worst strategy in this case is $(X, 3, 3, 2, 2, 3)$, where X means that any queue can be chosen (because the length of the corresponding visit time equals 0, since customers never join this queue). This strategy leads to an overall mean sojourn time of 7.48. As $\mathbb{E}[B_1]$ grows larger, Q_1 gradually will be chosen more frequently. In the symmetric case, $\mathbb{E}[B_1] = 1$, customers arriving during V_i choose Q_{i-1}, and customers arriving during S_i choose Q_i, resulting in a mean sojourn time of 8.5. For large $\mathbb{E}[B_1]$, the worst possible strategy might be a bit surprising. It is *not* simply to always join Q_1, but it is $(1, 1, 1, 2, 1, 3)$. During visit periods, customers always join Q_1, but during S_i customers join Q_i. For $\mathbb{E}[B_1] \uparrow \frac{5}{3}$, this strategy results in the highest mean sojourn time of an arbitrary customer. For the situation $\mathbb{E}[B_1] \geq \frac{5}{3}$, there are many strategies for which the system becomes unstable and sojourn times become infinite. The worst possible strategy for $\mathbb{E}[B_1] \geq \frac{5}{3}$ that still results in a stable system, is $(3, 1, X, 1, 1, 1)$.

10.A Appendix: MVA equations

In this appendix we present all MVA equations that have been omitted in Section 10.3.

The mean duration of the next period V_i, when in S_j is denoted by $\mathbb{E}[\overrightarrow{V_i}^{(S_j)}]$. A difference with $\mathbb{E}[\overrightarrow{V_i}^{(V_j)}]$, is that $\mathbb{E}[\overrightarrow{V_i}^{(S_i)}]$ is not different from $\mathbb{E}[\overrightarrow{V_i}^{(S_j)}]$ for $j \neq i$. Similar to (10.3.7), we have for $i = 1, \dots, N$, $j = i, \dots, i+N-1$:

$$\mathbb{E}[\overrightarrow{V_i}^{(S_j)}] = \mathbb{E}[BP_i] \left(\mathbb{E}[L_i^{(S_j)}] + \lambda_i^{(S_j)} \mathbb{E}[S_j^{res}] + \sum_{k=j+1}^{i+N-1} \left(\lambda_i^{(V_k)} \mathbb{E}[\overrightarrow{V_k}^{(S_j)}] + \lambda_i^{(S_k)} \mathbb{E}[S_k] \right) \right). \tag{10.A.1}$$

Equation (10.3.9) for $\mathbb{E}[(S_i:V_j)^{res}]$, the mean residual duration of the interval $S_i, V_{i+1}, \dots, V_j$, is obtained by conditioning on the period in which the interval is observed, looking forward in time. Similarly, we find expressions for $\mathbb{E}[(S_i:S_j)^{res}]$, $\mathbb{E}[(V_i:V_j)^{res}]$, and $\mathbb{E}[(V_i:S_j)^{res}]$. For $i = 1, \dots, N$, $j = i+1, \dots, i+N-1$:

$$\begin{aligned}\mathbb{E}[(S_i:S_j)^{res}] = &\sum_{k=i}^{j} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(S_i:S_j)]} \left(\mathbb{E}[S_k{}^{res}] + \sum_{l=k+1}^{j} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(S_k)}] \right) \right) \\ &+ \sum_{k=i+1}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(S_i:S_j)]} \left(\sum_{l=k}^{j} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(V_k)}] \right) \right).\end{aligned} \tag{10.A.2}$$

For $i = 1, \dots, N$, $j = i+1, \dots, i+N-1$:

$$\begin{aligned}\mathbb{E}[(V_i:V_j)^{res}] = &\sum_{k=i}^{j-1} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(V_i:V_j)]} \left(\mathbb{E}[S_k{}^{res}] + \mathbb{E}[\overrightarrow{V_j}^{(S_k)}] + \sum_{l=k+1}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(S_k)}] \right) \right) \\ &+ \sum_{k=i}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(V_i:V_j)]} \left(\mathbb{E}[\overrightarrow{V_j}^{(V_k)}] + \sum_{l=k}^{j-1} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(V_k)}] \right) \right).\end{aligned} \tag{10.A.3}$$

For $i = 1, \dots, N$, $j = i+1, \dots, i+N-1$:

$$\begin{aligned}\mathbb{E}[(V_i:S_j)^{res}] = &\sum_{k=i}^{j} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(V_i:S_j)]} \left(\mathbb{E}[S_k{}^{res}] + \sum_{l=k+1}^{j} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(S_k)}] \right) \right) \\ &+ \sum_{k=i}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(V_i:S_j)]} \left(\sum_{l=k}^{j} \left(\mathbb{E}[S_l] + \mathbb{E}[\overrightarrow{V_l}^{(V_k)}] \right) \right).\end{aligned} \tag{10.A.4}$$

In Section 10.3, a second set of equations is discussed for $\mathbb{E}[(S_i:V_j)^{res}]$, $\mathbb{E}[(S_i:S_j)^{res}]$, $\mathbb{E}[(V_i:V_j)^{res}]$, and $\mathbb{E}[(V_i:S_j)^{res}]$. This set is obtained by conditioning on the period in which the interval is observed, but now looking backward in time. We use that the residual length of an interval has the same distribution as the elapsed time of this interval. The equation for $\mathbb{E}[(S_i:V_j)^{res}]$ is given by (10.3.10). The other equations are given below. For $i = 1, \dots, N$, $j = i+1, \dots, i+N-1$:

$$\begin{aligned}\mathbb{E}[(S_i:S_j)^{res}] = &\sum_{k=i}^{j} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(S_i:S_j)]} \left(\mathbb{E}[\overleftarrow{S_i}^{(S_k)}] + \sum_{l=i+1}^{k} \left(\mathbb{E}[\overleftarrow{S_l}^{(S_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(S_k)}] \right) \right) \\ &+ \sum_{k=i+1}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(S_i:S_j)]} \left(\mathbb{E}[\overleftarrow{S_i}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_k}^{(V_k)}] + \sum_{l=i+1}^{k-1} \left(\mathbb{E}[\overleftarrow{S_l}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(V_k)}] \right) \right),\end{aligned} \tag{10.A.5}$$

for $i = 1, \dots, N$, $j = i+1, \dots, i+N-1$:

$$\mathbb{E}[(V_i : V_j)^{res}] = \sum_{k=i}^{j-1} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(V_i : V_j)]} \left(\sum_{l=i}^{k} \left(\mathbb{E}[\overleftarrow{S_l}^{(S_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(S_k)}] \right) \right)$$
$$+ \sum_{k=i}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(V_i : V_j)]} \left(\mathbb{E}[\overleftarrow{V_k}^{(V_k)}] + \sum_{l=i}^{k-1} \left(\mathbb{E}[\overleftarrow{S_l}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(V_k)}] \right) \right), \tag{10.A.6}$$

and for $i = 1, \dots, N$, $j = i, \dots, i+N-1$:

$$\mathbb{E}[(V_i : S_j)^{res}] = \sum_{k=i}^{j} \frac{\mathbb{E}[S_k]}{\mathbb{E}[(V_i : S_j)]} \left(\sum_{l=i}^{k} \left(\mathbb{E}[\overleftarrow{S_l}^{(S_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(S_k)}] \right) \right)$$
$$+ \sum_{k=i}^{j} \frac{\mathbb{E}[V_k]}{\mathbb{E}[(V_i : S_j)]} \left(\mathbb{E}[\overleftarrow{V_k}^{(V_k)}] + \sum_{l=i}^{k-1} \left(\mathbb{E}[\overleftarrow{S_l}^{(V_k)}] + \mathbb{E}[\overleftarrow{V_l}^{(V_k)}] \right) \right). \tag{10.A.7}$$

11

Closed-form waiting time approximation

In this chapter, we derive closed-form approximations for the mean waiting times and mean marginal queue lengths of polling systems with renewal arrival processes. The approximations derived can be computed by simple calculations. The results may be useful and suitable for the design and optimization phase in many application areas, such as telecommunication, maintenance, manufacturing and transportation.

11.1 Introduction

When studying literature on polling systems (e.g. [130, 177, 200]), it rapidly becomes apparent that the computation of the distributions and moments of the waiting times and marginal queue lengths is very cumbersome. Closed-form expressions do not exist, and even when one specifies the number of queues and solves the set of equations that leads to the mean waiting times, the obtained expressions are still too lengthy and complicated to interpret directly. Numerical procedures, both approximate and exact, have been developed in the past to compute these performance measures. However, these methods have several drawbacks. Firstly, they are not transparent and act as a kind of black box. It is, for instance, rather difficult to study the impact of parameters like the occupation rate and the service level. Secondly, these procedures are computationally complex and hard, if not impossible, to implement in a standard spreadsheet program commonly used on the work floor. Finally, the vast majority of standard methods focuses on Poisson arrival processes, which may not be very realistic in many application areas. In the present chapter we study polling systems in which the arrival streams are not (necessarily) Poisson, i.e., the interarrival times follow a general distribution. The goal is to derive closed-form approximate solutions for the mean waiting times and mean marginal queue lengths, which can be computed by simple spreadsheet calculations.

Our approach in developing an approximation for the mean waiting times uses novel developments in polling literature. Recently, a heavy traffic (HT) limit has been developed for the mean waiting times as the system becomes saturated [187]. In the present chapter we derive an approximation for the light traffic (LT) limit, i.e. as the load goes down to zero, which is exact for Poisson arrivals. The main idea is to create an interpolation between the LT limit and the HT limit. This interpolation yields good results, and has several nice properties, like satisfying the Pseudo Conservation Law (PCL), and being exact for symmetric systems with Poisson arrivals and in many limiting cases. These properties are described in more detail in the present chapter. In polling literature, several alternative approximations have been developed before, most of which assume Poisson arrivals. For polling systems with Poisson arrivals and gated or exhaustive service, the best results, by far, are obtained by an approximation based on the PCL (see e.g. [31, 72, 92]). Fischer et al. [79] study an approximation for the mean waiting times in polling systems, which is also based on an interpolation between (approximate) LT and HT limits. Their approach, however, is applied to a system with Poisson arrivals and time-limited service. Hardly any closed-form approximations exist for non-Poisson arrivals. The few that exist, perform well in specific limiting cases, e.g., under HT conditions [149, 187], or if switch-over times become very large [207, 209], but performance deteriorates rapidly if these limiting conditions are abandoned. We show in an extensive numerical study that the quality of our approximation can be compared to the PCL approximation for systems with Poisson arrivals, but provides good results as well for systems with renewal arrivals.

Because of its simple form, the approximation function is very suitable for optimization purposes. Although only the mean waiting times of systems with exhaustive or gated service are studied, the results can be extended to higher moments and general branching-type service disciplines. Polling systems with polling tables and/or batch service can also be analyzed in a similar manner.

The structure of the present chapter is as follows: the next section introduces the model and the required notation, and states the main result. Section 11.3 illustrates how this main result is obtained, while Section 11.4 provides results on the accuracy of the approximation for a large set of combinations of input parameter values. Finally, Section 11.5 discusses further research topics and possible extensions of the model. This chapter is based on [29].

11.2 Model description and main result

We consider a polling model as described in Section 9.2, with a generalization to renewal arrivals. That is, we allow the interarrival times to have an arbitrary distribution. We focus on both the gated as well as the exhaustive discipline.

We regard several variables as a function of the load ρ in the system. Scaling is done by keeping the service time distributions fixed, and varying the interarrival times. For each variable x that is a function of the load in the system, ρ, its value evaluated at $\rho = 1$ is denoted by $\hat{x}$. For $\rho = 1$, the generic interarrival time of the stream in Q_i is denoted by $\hat{A}_i$. Reducing the load ρ is done by scaling the interarrival times, i.e., taking the random variable $A_i := \hat{A}_i/\rho$ as generic interarrival time at Q_i. After scaling, the load at Q_i becomes $\rho_i = \rho\, \mathbb{E}[B_i]/\mathbb{E}[\hat{A}_i]$. The (scaled) rate of the arrival stream at Q_i is defined as $\lambda_i = 1/\mathbb{E}[A_i]$. Similarly, we define arrival rates $\hat{\lambda}_i = 1/\mathbb{E}[\hat{A}_i]$, and the proportional load at Q_i, $\hat{\rho}_i = \rho_i/\rho$. Note that $\sum_{i=1}^{N} \hat{\rho}_i = 1$. We use B to denote the generic service requirement of an arbitrary customer entering the system, with

$$\mathbb{E}[B^k] = \frac{\sum_{i=1}^{N} \hat{\lambda}_i \mathbb{E}[B_i^k]}{\sum_{j=1}^{N} \hat{\lambda}_j}$$

for any integer $k > 0$. The system is assumed to be stable, so ρ is varied between 0 and 1.

We now present the main result of this chapter, which is a closed-form approximation formula for the mean waiting time $\mathbb{E}[W_i]$ of a type i customer as a function of ρ:

$$\mathbb{E}[W_{i,app}] = \frac{K_{0,i} + K_{1,i}\rho + K_{2,i}\rho^2}{1-\rho}, \qquad i = 1, \dots, N. \tag{11.2.1}$$

The constants $K_{0,i}$, $K_{1,i}$, and $K_{2,i}$ depend on the input parameters and the service discipline. If all queues receive *exhaustive* service, the constants become:

$$K_{0,i} = \mathbb{E}[S^{res}], \tag{11.2.2}$$

$$K_{1,i} = \hat{\rho}_i \big(\mathbb{E}[\hat{A}_i]\hat{g}_i(0) - 1\big)\mathbb{E}[B_i^{res}] + \mathbb{E}[B^{res}] + \hat{\rho}_i \big(\mathbb{E}[S^{res}] - \mathbb{E}[S]\big) - \frac{1}{\mathbb{E}[S]} \sum_{j=0}^{N-1} \sum_{k=0}^{j} \hat{\rho}_{i+k} \mathbb{V}ar[S_{i+j}], \tag{11.2.3}$$

$$K_{2,i} = \frac{1-\hat{\rho}_i}{2} \left(\frac{\sum_{j=1}^{N} \hat{\lambda}_j \left(\mathbb{V}ar[B_j] + \hat{\rho}_j^2 \mathbb{V}ar[\hat{A}_j]\right)}{\sum_{j=1}^{N} \hat{\rho}_j (1-\hat{\rho}_j)} + \mathbb{E}[S] \right) - K_{0,i} - K_{1,i}. \tag{11.2.4}$$

If all queues receive *gated* service, we get:

$$K_{0,i} = \mathbb{E}[S^{res}], \tag{11.2.5}$$

$$K_{1,i} = \hat{\rho}_i \big(\mathbb{E}[\hat{A}_i]\hat{g}_i(0) - 1\big)\mathbb{E}[B_i^{res}] + \mathbb{E}[B^{res}] + \hat{\rho}_i \mathbb{E}[S^{res}] - \frac{1}{\mathbb{E}[S]} \sum_{j=0}^{N-1} \sum_{k=0}^{j} \hat{\rho}_{i+k} \mathbb{V}ar[S_{i+j}], \tag{11.2.6}$$

$$K_{2,i} = \frac{1+\hat{\rho}_i}{2} \left(\frac{\sum_{j=1}^{N} \hat{\lambda}_j \left(\mathbb{V}ar[B_j] + \hat{\rho}_j^2 \mathbb{V}ar[\hat{A}_j]\right)}{\sum_{j=1}^{N} \hat{\rho}_j (1+\hat{\rho}_j)} + \mathbb{E}[S] \right) - K_{0,i} - K_{1,i}. \tag{11.2.7}$$

The term $\hat{g}_i(t)$ is the density of $\hat{A}_i$, the interarrival times at $\rho = 1$. This term is discussed in more detail in the next section, but for practical purposes it is useful to know that $\mathbb{E}[\hat{A}_i]\hat{g}_i(0)$ can be very well approximated by

$$\mathbb{E}[\hat{A}_i]\hat{g}_i(0) \approx \begin{cases} 2\,\frac{cv_{A_i}^2}{cv_{A_i}^2+1} & \text{if } cv_{A_i}^2 > 1, \\ \left(cv_{A_i}^2\right)^4 & \text{if } cv_{A_i}^2 \leq 1, \end{cases}$$

where $cv_{A_i}^2$ is the squared coefficient of variation (SCV) of A_i (and, hence, also of $\hat{A}_i$). Note that this simplification results in an approximation that requires only the first two moments of each input variable (i.e., service times, switch-over times, and interarrival times).

REMARK 11.2.1. In case of Poisson arrivals, the constants $K_{1,i}$ and $K_{2,i}$ simplify considerably. E.g., for exhaustive service they simplify to:

$$K_{1,i}^{Poisson} = \mathbb{E}[B^{res}] + \hat{\rho}_i \left(\mathbb{E}[S^{res}] - \mathbb{E}[S] \right) - \frac{1}{\mathbb{E}[S]} \sum_{j=0}^{N-1} \sum_{k=0}^{j} \hat{\rho}_{i+k} \mathbb{V}ar[S_{i+j}],$$

$$K_{2,i}^{Poisson} = (1 - \hat{\rho}_i) \left(\frac{\mathbb{E}[B^{res}]}{\sum_{j=1}^{N} \hat{\rho}_j (1 - \hat{\rho}_j)} + \frac{\mathbb{E}[S]}{2} \right) - K_{0,i} - K_{1,i}^{Poisson}.$$

The derivation of this approximative formula for the mean waiting time is the topic of the next section. An approximation for the mean *queue length* at Q_i, $\mathbb{E}[L_i]$ is obtained by application of Little's Law to the *sojourn time* of type i customers, i.e. the waiting time plus the service time. As a function of ρ, we have

$$\mathbb{E}[L_{i,app}] = \rho \frac{\mathbb{E}[W_{i,app}] + \mathbb{E}[B_i]}{\mathbb{E}[\hat{A}_i]}.$$

11.3 Idea behind the approximation

In the present section, we explain the idea behind approximation (11.2.1) for $\mathbb{E}[W_i]$. The restrictions that we impose on our approximation, are firstly that the formula should be closed-form, and easy to implement, since these are necessities for optimization purposes and implementation in a spreadsheet. Secondly, we want the approximation to capture the light traffic limit, i.e. $\rho \downarrow 0$, and high traffic limit, i.e. $\rho \uparrow 1$, behavior in an exact way. Based on these restrictions, we have chosen the form

$$\mathbb{E}[W_{i,app}] = \frac{K_{0,i} + K_{1,i}\rho + K_{2,i}\rho^2}{1-\rho}, \qquad i = 1, \ldots, N.$$

It is proved in [187] that capturing the HT behavior in an exact way, requires the $(1-\rho)$ term in the denominator. This term is not surprising at all, because the mean waiting times of practically all queueing systems show this behavior (the best known exception is an $M/G/1$ queue with shortest remaining processing time policy [163]). The motivation for taking a polynomial in the numerator of (11.2.1) can be found in several other approximations based on interpolation between LT and HT limits. E.g., Reiman and Simon [158] (see also [168]), and Whitt [206] use this approach to develop approximations for the mean waiting time in, respectively, queueing systems with Poisson input and $GI/G/1$ queues. A second-order polynomial fulfills the need for simplicity, *and* is sufficient to obtain an approximation which is exact for the two limiting situations (and, as is shown in Section 11.3.4, in many other limiting cases).

The remainder of this section is devoted to finding the constants $K_{0,i}$, $K_{1,i}$, and $K_{2,i}$. The requirement for the interpolation is an approximation for $\mathbb{E}[W_i]$ in light traffic. No such approximation exists in existing literature, so the next subsection is devoted to finding one. We can use this LT expression to find constants $K_{0,i}$ and $K_{1,i}$ in (11.2.1). The last unknown in the interpolation, $K_{2,i}$, is obtained using the HT limit of the mean waiting time, which has been found quite recently [187].

11.3.1 Light traffic

The mean waiting times in the polling model under consideration in light-traffic, have been studied in Blanc and Van der Mei [22], under the assumption of Poisson arrivals. They obtain expressions for the mean waiting times in light traffic that are exact up to (and including) first-order terms in ρ. These expressions have been found by carefully inspecting numerical results obtained with the Power-Series Algorithm, but no proof is provided. In the present section we shall not only prove the correctness of the light-traffic results in a system with Poisson arrivals, but also use them as base for an approximation for the mean waiting times in polling systems with renewal interarrival times. The key ingredient to the LT analysis of a polling system, is the well-known Fuhrmann-Cooper decomposition [84]. It states that in a vacation system with Poisson arrivals the queue length of a customer is the sum of two independent random variables: the number of customers in an isolated $M/G/1$ queue, and the number of customers during an arbitrary moment in the vacation period. The distributional form of Little's Law [109] can be used to translate this result to waiting times. Since no independence is required between the length of a vacation and the length of the preceding visit period, this decomposition also holds for polling systems with Poisson arrivals. Recall that V_i denotes the length of a visit period at Q_i, I_i denotes the length of the intervisit period, i.e. the time that the server is away between two successive visits to Q_i. Furthermore, C_i denotes the cycle time, starting at a visit beginning to Q_i. It holds that $\mathbb{E}[V_i] = \rho_i \mathbb{E}[C_i]$, $\mathbb{E}[I_i] = (1-\rho_i)\mathbb{E}[C_i]$, and $\mathbb{E}[C_i] = \mathbb{E}[C] = \mathbb{E}[S]/(1-\rho)$.

The Fuhrmann-Cooper decomposition, applied to the mean waiting time, results in:

$$\text{exhaustive:} \qquad \mathbb{E}[W_i] = \mathbb{E}[W_{i,M/G/1}] + \mathbb{E}[I_i^{res}], \tag{11.3.1}$$

$$\text{gated:} \qquad \mathbb{E}[W_i] = \mathbb{E}[W_{i,M/G/1}] + \mathbb{E}[I_i^{res}] + \frac{\mathbb{E}[V_i I_i]}{\mathbb{E}[I_i]}. \tag{11.3.2}$$

For our approximation, we assume that this decomposition also holds for renewal arrival processes in light traffic. Determining the LT limit of the mean waiting time, $\mathbb{E}[W_i^{LT}]$, in a polling system with exhaustive or gated service is based on the following two-step approach. The first step is to find the LT limit of $\mathbb{E}[W_{i,GI/G/1}]$, the mean waiting time of a $GI/G/1$ queue with only type i customers in isolation, $i = 1, \dots, N$. The second step is determining $\mathbb{E}[I_i^{res}]$, the mean residual intervisit time of Q_i, and $\mathbb{E}[V_i I_i]/\mathbb{E}[I_i]$, the mean visit time of Q_i given that it is being observed at a random epoch during the following intervisit time.

For the LT limit of the mean waiting time in a $GI/G/1$ queue, we use Whitt's result [206, Equation (16)], which gives:

$$\lim_{\rho_i \downarrow 0} \frac{\mathbb{E}[W_{i,GI/G/1}]}{\rho_i} = \frac{1+cv_{B_i}^2}{2} \mathbb{E}[\hat{A}_i]\hat{g}_i(0)\mathbb{E}[B_i], \tag{11.3.3}$$

where $cv_{B_i}^2$ is the SCV of the service times, and $\hat{g}_i(t)$ is the density of the interarrival times $\hat{A}_i$. For practical purposes, it may be more convenient to express $\hat{g}_i(0)$ in terms of the density of A_i, the generic interarrival time of Q_i in the scaled situation.The relation between the density of the scaled interarrival times $A_i\,(= \hat{A}_i/\rho)$, denoted by $g_i(t)$, and the density of $\hat{A}_i$, $\hat{g}_i(t)$, is simply: $g_i(t) = \rho\hat{g}_i(\rho t)$. This means that the term $\mathbb{E}[\hat{A}_i]\hat{g}_i(0)$ can be rewritten as

$$\mathbb{E}[\hat{A}_i]\hat{g}_i(0) = \mathbb{E}[A_i]g_i(0).$$

Because of this equality, in the remainder of the chapter we might use either notation. Since determining $\mathbb{E}[\hat{A}_i]\hat{g}_i(0)$ is a required step in the computation of our approximation for $\mathbb{E}[W_i]$, we give some practical examples.

Example 11.3.1. If the scaled interarrival times A_i are exponentially distributed with parameter $\lambda_i := 1/\mathbb{E}[A_i]$, we have $g_i(t) = \lambda_i \mathrm{e}^{-\lambda_i t}$. This implies that $\mathbb{E}[A_i]g_i(0) = 1$.

Example 11.3.2. Assume that A_i follows a H_2 distribution with balanced means. The SCV of A_i is denoted by $cv^2_{A_i}$. The density of this hyper-exponential distribution is (see e.g. [182])

$$g_i(t) = p\mu_1 \mathrm{e}^{-\mu_1 t} + (1-p)\mu_2 \mathrm{e}^{-\mu_2 t},$$

with

$$p = \frac{1}{2}\left(1 + \sqrt{\frac{cv^2_{A_i} - 1}{cv^2_{A_i} + 1}}\right),$$

$$\mu_1 = \frac{1}{\mathbb{E}\left[A_i\right]}\left(1 + \sqrt{\frac{cv^2_{A_i} - 1}{cv^2_{A_i} + 1}}\right),$$

$$\mu_2 = \frac{1}{\mathbb{E}\left[A_i\right]}\left(1 - \sqrt{\frac{cv^2_{A_i} - 1}{cv^2_{A_i} + 1}}\right).$$

This leads to $\mathbb{E}[A_i]g_i(0) = 1 + (cv^2_A - 1)/(cv^2_A + 1) = 2\,cv^2_A/(cv^2_A + 1)$.

Example 11.3.3. Assume that the interarrival times follow a mixed Erlang distribution. The density of the scaled interarrival times is:

$$g_i(t) = p\frac{\mu^{k-1}t^{k-2}}{(k-2)!}\mathrm{e}^{-\mu t} + (1-p)\frac{\mu^k t^{k-1}}{(k-1)!}\mathrm{e}^{-\mu t},$$

i.e., a mixture of an Erlang$(k-1)$ and an Erlang(k) distribution with

$$k = \left\lceil 1/cv^2_{A_i} \right\rceil,$$

$$p = \frac{k\,cv^2_{A_i} - \sqrt{k(1 + cv^2_{A_i}) - k^2\,cv^2_{A_i}}}{1 + cv^2_{A_i}},$$

$$\mu = \frac{k-p}{\mathbb{E}[A_i]}.$$

If $k > 2$, this leads to $\mathbb{E}[A_i]\,g_i(0) = 0$.

The distributions in Examples 11.3.1–11.3.3 are typical distributions to be used in a two-moment fit if the SCV of the interarrival times is respectively 1, greater than 1, and less than 1 (cf. [182]). The examples illustrate how $\mathbb{E}[A_i]g_i(0)$ can be computed if the density of the (scaled) interarrival times is known. If no information is available about the complete density, but the first two moments of A_i are known, Whitt [206] suggests to use the following approximation for $\mathbb{E}[A_i]g_i(0)$:

$$\mathbb{E}[A_i]g_i(0) = \begin{cases} 2\frac{cv^2_{A_i}}{cv^2_{A_i}+1} & \text{if } cv^2_{A_i} > 1, \\ \left(cv^2_{A_i}\right)^4 & \text{if } cv^2_{A_i} \leq 1, \end{cases}$$

where $cv^2_{A_i}$ is the squared coefficient of variation of the interarrival times of Q_i. This approximation is exact for $cv^2_{A_i} > 1$, if the interarrival time distribution is a hyper-exponential distribution as discussed in Example 11.3.2. For $cv^2_{A_i} \leq 1$, the approximation is rather arbitrary, but Example 11.3.3 shows that $\mathbb{E}[A_i]g_i(0)$ becomes small (or even zero) very rapidly as $cv^2_{A_i}$ gets smaller.

Summarizing, the LT limit of a $GI/G/1$ queue (ignoring $O(\rho_i^2)$ terms and higher) is:

$$\mathbb{E}[W^{LT}_{i,GI/G/1}] = \rho_i\, \mathbb{E}[A_i]g_i(0)\mathbb{E}[B_i^{res}]. \tag{11.3.4}$$

For Poisson arrivals ($\mathbb{E}[A_i]g_i(0) = 1$), it is known that

$$\mathbb{E}[W_{i,M/G/1}] = \frac{\rho_i}{1-\rho_i}\mathbb{E}[B_i^{res}] = \rho_i\mathbb{E}[B_i^{res}] + O(\rho_i^2),$$

which is consistent with our approximation.

The second step in determining the LT limit of the mean waiting time of a type i customer in a polling system, is finding the LT limits of $\mathbb{E}[I_i^{res}]$, the mean residual *intervisit time* of Q_i, and (for gated service only) $\mathbb{E}[V_i I_i]/\mathbb{E}[I_i]$, the mean visit time V_i given that it is observed from the following intervisit time I_i. In this LT analysis we need to focus on first order terms only. Noting the fact that $I_i = S_i + V_{i+1} + S_{i+1} + \ldots + V_{i+N-1} + S_{i+N-1}$, we condition on the moment at which I_i is observed. We distinguish between two cases. The moment of observation either takes place during a visit time, or during a switch-over time:

$$\begin{aligned}\mathbb{E}[I_i^{LT,res}] &= \sum_{j=1}^{N-1}\frac{\mathbb{E}[V_{i+j}]}{\mathbb{E}[I_i]}\mathbb{E}[I_i^{LT,res}|\text{observed during } V_{i+j}] \\ &+ \sum_{j=0}^{N-1}\frac{\mathbb{E}[S_{i+j}]}{\mathbb{E}[I_i]}\mathbb{E}[I_i^{LT,res}|\text{observed during } S_{i+j}]. \end{aligned}\tag{11.3.5}$$

Observation during visit time. The probability that a random observation epoch takes place during a visit time, say V_j, is $\mathbb{E}[V_j]/\mathbb{E}[I_i]$, for any $j \neq i$. However, we are only interested in order ρ terms, so this probability simplifies to

$$\frac{\mathbb{E}[V_j]}{\mathbb{E}[I_i]} = \frac{\rho_j\mathbb{E}[C]}{(1-\rho_i)\mathbb{E}[C]} = \rho_j + O(\rho^2).$$

The fact that this probability is $O(\rho)$, implies that all further $O(\rho)$ terms in $\mathbb{E}[I_i^{LT,res}|$ observed during $V_j]$ can be ignored, because in LT we focus on first order terms only.

The length of the residual intervisit time is the length of the residual visit period of type j customers, V_j^{res}, plus all switch-over times $S_j + \ldots + S_{i-1}$, plus all visit times

$V_{j+1} + \ldots + V_{i-1}$. The first term simplifies to $\mathbb{E}[V_j^{res}] = \mathbb{E}[B_j^{res}] + O(\rho)$. The terms $\mathbb{E}[V_k|\text{observed from } V_j], k = j+1, \ldots, i-1$, in light traffic, are all $O(\rho)$. Summarizing, the mean residual intervisit period when observed during V_j is simply a mean residual service time $\mathbb{E}[B_j^{res}]$, plus all mean switch-over times $\mathbb{E}[S_j + \ldots + S_{i-1}]$, plus $O(\rho)$ terms:

$$\mathbb{E}[I_i^{LT,res}|\text{observed during } V_j] = \mathbb{E}[B_j^{res}] + \sum_{k=j}^{i-1} \mathbb{E}[S_k] + O(\rho). \tag{11.3.6}$$

Observation during switch-over time. We continue by determining the mean residual intervisit period, conditioned on a random observation epoch during a switch-over time, say S_j, $j = 1, \ldots, N$. The probability that such an epoch takes place during S_j, is

$$\frac{\mathbb{E}[S_j]}{\mathbb{E}[I_i]} = \frac{\mathbb{E}[S_j]}{(1-\rho_i)\mathbb{E}[C]} = \frac{\mathbb{E}[S_j]}{\mathbb{E}[S]} \frac{1-\rho}{1-\rho_i} = \frac{\mathbb{E}[S_j]}{\mathbb{E}[S]}(1-\rho+\rho_i) + O(\rho^2).$$

It becomes apparent from this expression that things get slightly more complicated now, because order ρ terms in the conditional residual intervisit time may no longer be neglected. The residual intervisit time now consists of the residual switch-over time S_j^{res}, plus the switch-over times $S_j + \ldots + S_{i-1}$, plus all visit periods $V_{j+1} + \ldots + V_{i-1}$. The length of a visit period V_k, for $k > j$, is the sum of the busy periods of all type k customers that have arrived during $S_i, \ldots, S_{j-1}$, S_j^{past}, S_j^{res}, and $S_{j+1}, \ldots, S_{k-1}$. By S_j^{past} we denote the elapsed switch-over time during which the intervisit period is observed, which has the same distribution as the residual switch-over time S_j^{res}. Compared to an observation during a visit time, it is more difficult to determine the conditional mean length of a busy period $\mathbb{E}[V_k|\text{observed during } S_j]$ under LT. We use a heuristic approach, which is exact if the arrival process of type k customers is Poisson, and approximate it by:

$$\mathbb{E}[V_k|\text{observed during } S_j] \approx \rho_k \left(\sum_{l \neq j} \mathbb{E}[S_l] + \mathbb{E}[S_j^{past}] + \mathbb{E}[S_j^{res}] \right) + O(\rho^2),$$

for $k = j+1, \ldots, i-1$. If A_k is exponentially distributed, the above expression is exact. Nevertheless, numerical experiments have shown that this approximative assumption has no or at least negligible impact on the accuracy of the approximated mean waiting times. Summarizing:

$$\begin{aligned}
&\mathbb{E}[I_i^{LT,res}|\text{observed during } S_j] \\
&\approx \sum_{k=i}^{j-1} \mathbb{E}[S_k] \left(\sum_{l=j+1}^{i+N-1} \rho_l \right) + \mathbb{E}\left(S_j^{past}\right) \left(\sum_{k=j+1}^{i+N-1} \rho_k \right) + \mathbb{E}\left(S_j^{res}\right) \left(1 + \sum_{k=j+1}^{i+N-1} \rho_k \right) \\
&\quad + \sum_{k=j+1}^{i+N-1} \mathbb{E}[S_k] \left(1 + \sum_{l=j+1}^{i+N-1} \rho_l \right) + O(\rho^2).
\end{aligned} \tag{11.3.7}$$

The expression for I_i^{res} under light traffic conditions now follows from substituting

(11.3.6) and (11.3.7) in (11.3.5). The result can be rewritten to:

$$
\begin{aligned}
&\mathbb{E}[I_i^{LT,res}] \\
&\approx \sum_{j=i+1}^{i+N-1} \rho_j \mathbb{E}[B_j^{res}] + \sum_{j=i+1}^{i+N-1} \rho_j \sum_{k=j}^{i+N-1} \mathbb{E}[S_k] \\
&\quad + \sum_{j=i}^{i+N-1} \frac{1}{2\,\mathbb{E}[S]} \left[\mathbb{E}\left(S_j^2\right) \left(1-\rho+\rho_i+2\sum_{k=j+1}^{i+N-1}\rho_k\right)\right] \\
&\quad + \frac{1}{\mathbb{E}[S]} \left[\sum_{k=i}^{j-1} \mathbb{E}[S_j]\mathbb{E}[S_k] \left(\sum_{l=j+1}^{i+N-1}\rho_l\right)\right. \qquad (11.3.8)\\
&\qquad\qquad \left. + \sum_{k=j+1}^{i+N-1} \mathbb{E}[S_j]\mathbb{E}[S_k] \left(1-\rho+\rho_i+\sum_{l=j+1}^{i+N-1}\rho_l\right)\right] \\
&\quad + O(\rho^2) \\
&= \sum_{j=i+1}^{i+N-1} \rho_j \mathbb{E}[B_j^{res}] + \sum_{j=i+1}^{i+N-1} \rho_j \sum_{k=j}^{i+N-1} \mathbb{E}[S_k] \\
&\quad + (1-\rho+\rho_i)\mathbb{E}[S^{res}] + \frac{1}{\mathbb{E}[S]} \sum_{j=i}^{i+N-1}\sum_{k=i}^{i+N-1} \mathbb{E}[S_jS_k] \left(\sum_{l=j+1}^{i+N-1}\rho_l\right) + O(\rho^2) \\
&= \mathbb{E}[S^{res}] + \rho\mathbb{E}[B^{res}] - \rho_i\mathbb{E}[B_i^{res}] + \rho_i\left(\mathbb{E}[S^{res}] - \mathbb{E}[S]\right) \\
&\quad - \frac{1}{\mathbb{E}[S]} \sum_{j=0}^{N-1}\sum_{k=0}^{j} \rho_{i+k}\mathbb{V}ar[S_{i+j}] + O(\rho^2), \qquad (11.3.9)
\end{aligned}
$$

for $i = 1, \ldots, N$. The last step in (11.3.9) follows after some straightforward (but tedious) rewriting.

The Fuhrmann-Cooper decomposition of the mean waiting time for customers in a polling system with *gated* service (11.3.2), also requires computing $\mathbb{E}[V_iI_i]/\mathbb{E}[I_i]$ under LT conditions. Here, again, we have to resort to using a heuristic and use $\mathbb{E}[V_iI_i]/\mathbb{E}[I_i] = \rho_i\mathbb{E}[S] + O(\rho^2)$, because this value is *exact* in the case of Poisson arrivals. Intuitively this term can be explained by observing that the only thing that changes for gated service, compared to exhaustive service, is that type i customers arriving during V_i are not served until the next cycle. As we have seen before, the probability of a type i arrival taking place during V_i is $\rho_i + O(\rho^2)$. The mean residual cycle, observed from a random epoch in V_i, is $\mathbb{E}[C_i^{res}|\text{observed during } V_i] = \mathbb{E}[S]+O(\rho)$. Combined, this gives $\mathbb{E}[V_iI_i]/\mathbb{E}[I_i] = \rho_i\mathbb{E}[S] + O(\rho^2)$, in the case of Poisson arrivals. If the arrival process is not Poisson, this is not exact, but we use it as an approximation.

Having made all required preparations, we are ready to formulate the main result of the present subsection. Under light traffic, an approximation for the mean waiting time of a type i customer in a polling model with general arrivals and respectively exhaustive and gated service in Q_i, is:

$$\mathbb{E}[W_i^{LT,exh}] \approx \mathbb{E}[S^{res}] + \rho_i(\mathbb{E}[\hat{A}_i]\hat{g}_i(0) - 1)\mathbb{E}[B_i^{res}] + \rho\mathbb{E}[B^{res}] \\ + (\rho - \rho_i)(\mathbb{E}[S] - \mathbb{E}[S^{res}]) + \frac{1}{\mathbb{E}[S]}\sum_{k=i+1}^{i+N-1}\rho_k\sum_{j=i}^{k-1}\mathbb{V}ar[S_j] + O(\rho^2), \tag{11.3.10}$$

$$\mathbb{E}[W_i^{LT,gated}] \approx \mathbb{E}[W_i^{LT,exh}] + \rho_i\mathbb{E}[S], \tag{11.3.11}$$

for $i = 1, \dots, N$, where $\hat{g}_i(t)$ is the density of the interarrival times of type i customers at $\rho = 1$. Equation (11.3.10) follows from substitution of (11.3.4) and (11.3.9) in

$$\mathbb{E}[W_i] \approx \mathbb{E}[W_{i,GI/G/1}] + \mathbb{E}[I_i^{res}], \qquad i = 1, \dots, N. \tag{11.3.12}$$

For Poisson arrivals, (11.3.10) and (11.3.11) are exact. The LT limit for polling systems with Bernoulli service (and Poisson arrivals) has been experimentally found in [22] and, indeed, it can be shown that their result for exhaustive service, which is a special case of Bernoulli service, agrees with our result after substituting $\mathbb{E}[\hat{A}_i]\hat{g}_i(0) = 1$ in (11.3.10).

11.3.2 Heavy traffic

The mean delay in a polling system with renewal arrivals in HT, i.e. as ρ tends to 1, has been analyzed in [187], where the following result has been obtained:

$$\mathbb{E}[W_i^{HT}] = \frac{\omega_i}{1-\rho} + o((1-\rho)^{-1}), \qquad \rho \uparrow 1. \tag{11.3.13}$$

Obviously, in HT, all queues become unstable and, thus, $\mathbb{E}[W_i]$ tends to infinity for all i. The rate at which $\mathbb{E}[W_i]$ tends to infinity as $\rho \uparrow 1$ is indicated by ω_i, which is referred to as the *mean asymptotic scaled delay* at queue i, and depends on the service discipline. For exhaustive service,

$$\omega_i = \frac{1-\hat{\rho}_i}{2}\left(\frac{\sigma^2}{\sum_{j=1}^N \hat{\rho}_j(1-\hat{\rho}_j)} + \mathbb{E}[S]\right), \qquad i = 1, \dots, N,$$

with

$$\sigma^2 := \sum_{i=1}^N \hat{\lambda}_i\left(\mathbb{V}ar[B_i] + \hat{\rho}_i^2\mathbb{V}ar[\hat{A}_i]\right).$$

Here, the limits are taken such that the arrival rates are increased, while keeping the service-time distributions fixed, and keeping the distributions of the interarrival times A_i, $(i = 1, \dots, N)$ fixed up to a common scaling constant ρ. Notice that in the case of Poisson arrivals we have $\sigma^2 = \mathbb{E}[B^2]/\mathbb{E}[B]$.

For gated service, we have

$$\omega_i = \frac{1+\hat{\rho}_i}{2}\left(\frac{\sigma^2}{\sum_{j=1}^N \hat{\rho}_j(1+\hat{\rho}_j)} + \mathbb{E}[S]\right), \qquad i = 1, \dots, N.$$

11.3.3 Interpolation

Now that we have the expressions for the mean delay in both LT and HT, we can determine the constants $K_{0,i}$, $K_{1,i}$, and $K_{2,i}$ in approximation formula (11.2.1). We simply impose the requirements that approximation (11.2.1) results in the same mean waiting time for $\rho = 0$ as the LT limit, and for $\rho \uparrow 1$ as the HT limit. Since (11.3.10) (and (11.3.11) for gated service) has been determined up to the first order of ρ terms, we also add the requirement that the derivative with respect to ρ, taken at $\rho = 0$, of our approximation is equal to the derivative of the LT limit. A more formal definition of these requirements is presented below:

$$\begin{aligned}
\mathbb{E}[W_{i,app}]\big|_{\rho=0} &= \mathbb{E}[W_i]\big|_{\rho=0},\\
\frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[W_{i,app}]\big|_{\rho=0} &= \frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[W_i]\big|_{\rho=0},\\
(1-\rho)\mathbb{E}[W_{i,app}]\big|_{\rho=1} &= (1-\rho)\mathbb{E}[W_i]\big|_{\rho=1}.
\end{aligned}$$

This leads to (11.2.1) as approximation for $\mathbb{E}[W_i]$ in a polling system with general arrivals. Constants $K_{0,i}$, $K_{1,i}$, and $K_{2,i}$ are defined in (11.2.2)–(11.2.4) for systems with exhaustive service, or (11.2.5)–(11.2.7) for gated service.

11.3.4 Special cases

The approximation for the mean waiting time of a type i customer, $\mathbb{E}[W_{i,app}]$, has several nice properties discussed in the remainder of this subsection.

Pseudo-conservation law. A well-known result in polling literature, is the Pseudo-Conservation Law, derived by Boxma and Groenendijk [34], see Section 9.3.1:

$$\sum_{i=1}^{N}\rho_i\mathbb{E}[W_i] = \frac{\rho}{1-\rho}\sum_{i=1}^{N}\rho_i\mathbb{E}[B_i^{res}] + \rho\mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)}\left(\rho^2 - \sum_{i=1}^{N}\rho_i^2\right) + \sum_{i=1}^{N}\mathbb{E}[M_i], \tag{11.3.14}$$

where $\mathbb{E}[M_i]$ is the mean amount of work in Q_i at a departure epoch of the server from Q_i. Hence, $\mathbb{E}[M_i] = 0$ for the exhaustive discipline, and $\mathbb{E}[M_i] = \rho_i\mathbb{E}[V_i] = \rho_i^2\mathbb{E}[S]/(1-\rho)$ for the gated discipline.

It can be shown that our approximation satisfies the pseudo-conservation law in the case of Poisson arrivals: if $\mathbb{E}[\hat{A}_i]\hat{g}_i(0) = 1$ for $i = 1,\ldots,N$, then $\sum_{i=1}^{N}\rho_i\mathbb{E}[W_{i,app}]$ also equals the right-hand side of (11.3.14). The derivation consists of basic, but cumbersome, algebraic manipulations only, and is therefore omitted. We only mention a helpful intermediate result:

$$\sum_{i=1}^{N}\sum_{k=i+1}^{i+N}\sum_{j=i}^{k-1}\rho_i\rho_k = N\sum_{i=1}^{N}\sum_{k=i}^{N}\rho_i\rho_k,$$

so

$$\sum_{i=1}^{N}\sum_{k=i+1}^{i+N}\sum_{j=i}^{k-1}\rho_i\rho_k\mathbb{V}ar[S_j] = \frac{1}{2}\left(\rho^2 + \sum_{i=1}^{N}\rho_i^2\right)\mathbb{V}ar[S].$$

Using this result, it follows that $\sum_{i=1}^{N}\rho_i K_{2,i} = 0$.

Light and heavy traffic. The light traffic limit of $\mathbb{E}[W_i]$, given by (11.3.10) for exhaustive service and by (11.3.11) for gated service, is exact for Poisson arrivals. The heavy traffic limit (11.3.13) of $\mathbb{E}[W_i]$ is even exact for renewal arrivals. An appropriate choice of constants $K_{0,i}$, $K_{1,i}$, and $K_{2,i}$ can reduce (11.2.1) to either (11.3.10), (11.3.11), or (11.3.13). Since the LT and HT limits have been used in the set of equations that determine the coefficients of the approximation, it goes without saying that $\mathbb{E}[W_{i,app}]$ is equal to (11.3.10) (or (11.3.11) for gated service) and (11.3.13), for $\rho \downarrow 0$ and $\rho \uparrow 1$ respectively. This implies that the LT limit of our approximation is exact for Poisson arrivals, and the HT limit is exact for general arrivals.

Symmetric system. If $\hat{\rho}_i = 1/N$ for all $i = 1, \dots, N$, all B_i have the same distribution, and the variances $\mathbb{V}ar[S_i]$ of all switch-over times are equal, then our approximation is exact if all interarrival distributions are exponential. For exhaustive service, we obtain

$$\begin{aligned} K_{1,i} &= \mathbb{E}[B^{res}] + \frac{N-1}{N}\mathbb{E}[S] - \left(2 - \frac{1}{N}\right)\mathbb{E}[S^{res}] + \frac{1}{\mathbb{E}[S]}\sum_{k=i+1}^{i+N-1}\hat{\rho}_k \sum_{j=i}^{k-1}\mathbb{V}ar[S_j] \\ &= \mathbb{E}[B^{res}] + \frac{N-1}{N}\mathbb{E}[S] - \left(2 - \frac{1}{N}\right)\mathbb{E}[S^{res}] + \frac{N-1}{N}\frac{\mathbb{V}ar[S]}{2\,\mathbb{E}[S]} \\ &= \mathbb{E}[B^{res}] + \left(1 - \frac{1}{N}\right)\frac{\mathbb{E}[S]}{2} - \mathbb{E}[S^{res}], \end{aligned}$$

which means that $\mathbb{E}[W_{i,app}] = \mathbb{E}[W_{i,symm}]$ (because $K_{2,i} = 0$ in a symmetric system), with

$$\mathbb{E}[W_{i,symm}] = \frac{\rho}{1-\rho}\mathbb{E}[B^{res}] + \mathbb{E}[S^{res}] + \frac{\rho\left(1 - \frac{1}{N}\right)}{1-\rho}\frac{\mathbb{E}[S]}{2}.$$

Note that we do *not* require that the mean switch-over times $\mathbb{E}[S_i]$ are equal. One can verify that the same holds for gated service.

Single queue (vacation model). An immediate consequence of the fact that our approximation is exact in symmetric polling systems with Poisson arrivals, is that it also gives exact results for the mean waiting time of customers in a single-queue polling system with Poisson arrivals. A polling system consisting of only one queue, but with a switch-over time between successive visits to this queue, is generally referred to as a queueing system with multiple server vacations.

Large switch-over times. For S deterministic, $S \to \infty$, and, again, under the assumption of Poisson arrivals, it is proven in [208, 207] that $\frac{\mathbb{E}[W_i]}{S} \to \frac{1-\rho_i}{2(1-\rho)}$ for exhaustive service. It can easily be verified that our approximation has the same limiting behavior:

$$\lim_{S\to\infty}\frac{\mathbb{E}[W_{i,app}]}{S} = \frac{1-\rho_i}{2(1-\rho)}.$$

For gated service

$$\lim_{S\to\infty}\frac{\mathbb{E}[W_{i,app}]}{S} \to \frac{1+\rho_i}{2(1-\rho)},$$

which is also the exact limit (see e.g. [208]).

Miscellaneous other exact results. The approximation is also exact in several other cases, all with Poisson arrivals, when the parameter values are carefully chosen. The relations between the input parameters that yield exact approximation results become very complicated, especially in polling systems with more than two queues. We only mention one interesting example here: our approximation gives exact results for a two-queue polling system with exhaustive service and

$$\mathbb{E}[B_1] = \mathbb{E}[B_2],\ \mathbb{E}[S_1] = \mathbb{E}[S_2],\ cv^2_{A_1} = cv^2_{A_2},\ cv^2_{B_1} = cv^2_{B_2},\ cv^2_{S_1} = cv^2_{S_2}, \tag{11.3.15}$$

if the following constraint is satisfied:

$$\rho = \frac{1 + I^2_{A_i}}{2I_{A_i}} - \frac{cv^2_{S_i}}{1 + cv^2_{B_i}} \cdot \frac{\mathbb{E}[S_i]}{\mathbb{E}[B_i]}, \tag{11.3.16}$$

where $I_{A_i} = \hat{\rho}_1/\hat{\rho}_2$ is the ratio of the loads of the two queues. Obviously, if $I_{A_i} = 1$, the system is symmetric and our approximation gives exact results regardless of the other parameter settings.

11.4 Numerical study

11.4.1 Initial glance at the approximation

Before we study the accuracy of the approximation to a huge test bed of polling systems, we just pick a rather arbitrary, simple system to compare the approximation with exact results in order to get some initial insights. Consider a three-queue polling system with loads of Q_1, Q_2, and Q_3 divided as follows: $\hat{\rho}_1 = 0.1, \hat{\rho}_2 = 0.3$, and $\hat{\rho}_3 = 0.6$. All service times and switch-over times are exponentially distributed, with mean 1. The interarrival times have SCV $cv^2_{A_i} = 3$ for $i = 1, 2, 3$. In Figure 11.1 we plot the approximated mean waiting time of Q_2, $\mathbb{E}[W_{2,app}]$, versus the load of the system ρ. Since this system cannot be analyzed analytically, we compare the approximated values with simulated values. Both in the approximation and in the simulation we fit a H_2 distribution as described in Example 11.3.2.

The errors are largest for Q_2, which is the reason why we chose this queue in particular in Figure 11.1. The most important information that this figure reveals, is that even though the accuracy of the approximation is worst for this queue (a relative error of -4.47% for $\rho = 0.7$), the shape of the approximation function is very close to the shape of the exact function, which makes it very suitable for optimisation purposes. The maximum relative errors of Q_1 and Q_3 are 3.10% and 2.90% respectively.

In order to get more insight in the numerical accuracy of the approximation for a huge variety of different parameter settings, we create a large test bed in the next subsection and compare the approximation with exact or simulated results. It turns out that the maximum relative errors for most of the polling systems are smaller than the one selected in the above example.

11.4.2 Accuracy of the approximation

In the present section we study the accuracy of our approximation. We compare the approximated mean waiting times of customers in various polling systems to the exact

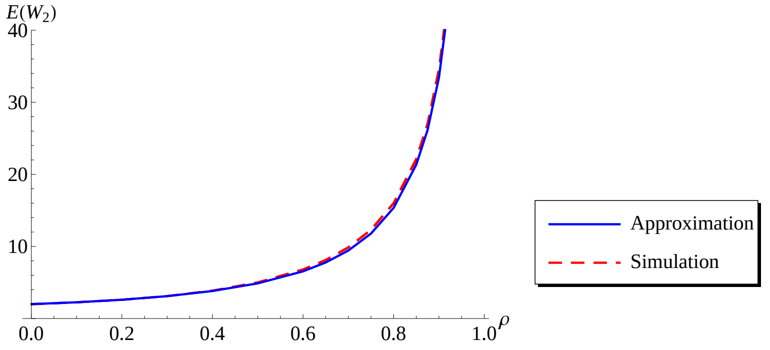


Figure 11.1: Approximated and simulated mean waiting time $\mathbb{E}[W_2]$ of Q_2 of the example in Section 11.4.1.

values. The complete test bed of polling systems that are analyzed, contains 2304 different combinations of parameter values, all listed in Table 11.1. We show detailed results for exhaustive service first, and discuss polling systems with gated service at the end of this section. We have varied the load between 0.1 and 0.9 with steps of 0.2, and included

Parameter	Notation	Values
Number of queues	N	$2, 3, 4, 5$
Load	ρ	$0.1, 0.3, 0.5, 0.7, 0.9, 0.99$
SCV interarrival times	$cv^2_{A_i}$	$0.25, 1, 2$
SCV service times	$cv^2_{B_i}$	$0.25, 1$
SCV switch-over times	$cv^2_{S_i}$	$0.25, 1$
Imbalance interarrival times	I_{A_i}	$1, 5$
Imbalance service times	I_{B_i}	$1, 5$
Ratio service and switch-over times	I_{S_i/B_i}	$1, 5$

Table 11.1: Test bed used to compare the approximation to exact results.

$\rho = 0.99$ to analyze the limiting behavior of our approximation when the load tends to 1. The SCV of the interarrival times, $cv^2_{A_i}$, is varied between 0.25 and 2. In case of non-Poisson arrivals, i.e. $cv^2_{A_i} \neq 1$, the exact values have been established through extensive simulation because they cannot be obtained in an analytic way. In these simulations we fit a phase-type distribution to the first two moments of the interarrival times, as described in Examples 11.3.2 and 11.3.3. For service times and switch-over times, only SCVs of 0.25 and 1 are considered. SCVs greater than 1 are less common in practice and are discussed separately from the test bed later in this section. The imbalance in interarrival times and service times, I_{A_i} and I_{B_i}, is the ratio between the largest and the smallest mean interarrival/service time. The interarrival times are determined in such a way, that the overall mean is always 1, λ_1 is the largest and λ_N the smallest, and the steps between the λ_i are linear. E.g., for $N = 5$ and $I_{A_i} = 5$ we get $\lambda_i = 2 - i/3, i = 1, \dots, 5$. The mean service times $\mathbb{E}[B_i]$ increase linearly in $i = 1, \ldots, N$, with $\mathbb{E}[B_N] = I_{B_i}\mathbb{E}[B_1]$ (so $\mathbb{E}[B_1]$ is the smallest mean service time). They follow from the relation $\sum_{i=1}^{N} \lambda_i \mathbb{E}[B_i] = \rho$. E.g., for $N = 5$, and $I_{A_i} = I_{B_i} = 5$ we get $\mathbb{E}[B_i]/\rho = 3i/35$. The last parameter that is varied in the test bed, is the ratio between the mean switch-over times and the mean service times, $I_{S_i/B_i} = \mathbb{E}[S_i]/\mathbb{E}[B_i]$. The total number of systems analyzed is $4 \times 6 \times 3 \times 2^5 = 2304$. A system consisting of N queues results N mean waiting times, $\mathbb{E}[W_1], \ldots, \mathbb{E}[W_N]$, so in total these 2304 systems yield 8064 mean waiting times. The absolute relative errors, defined as $|o - e|/e$, where o stands for observed (approximated) value, and e stands for expected (exact) value, are computed for all these 8064 queues. Table 11.2 shows these relative errors (times 100%) categorised in bins of 5%. In this table, and in all other tables, results for systems with a different number of queues are displayed in separate rows. The reader should keep in mind that the statistics in each row are based on $\frac{1}{4} \times 2304 \times N$ absolute relative errors, where N is the number of queues used in the specified row. Table 11.2 shows that, e.g., 98.84% of the approximated mean waiting times in polling systems consisting of 3 queues deviate less than 5% from their true values. From Table 11.2 it can be concluded that the approximation accuracy increases with the number of queues in a polling system. More specifically, for systems with more than 2 queues, no approximation errors are greater than 10%, and the vast majority is less than 5%. The mean relative errors for $N = 2, \ldots, 5$ are respectively 2.18%, 0.93%, 0.70%, and 0.57%. It is also noteworthy, that 193 out of the 2304 systems yield exact results. All of these 193 systems have Poisson input, and all of them – except for one – are symmetric. The only asymmetric case for which our approximation yields an exact result, happens to satisfy constraints (11.3.15) and (11.3.16).

In Table 11.3 the mean relative error percentages are shown for a combination of input parameter settings. The number of queues is always varied per row, while per column another input parameter is varied. This way we can find in more detail which (combinations of) parameter settings result in large approximation errors. In Table 11.3(a) the load ρ is varied, and it can be seen that for a load of $\rho = 0.7$ the approximation is least accurate. E.g., the mean relative error of all approximated waiting times in polling systems consisting of 3 queues with a load of $\rho = 0.7$ is 1.69%. Table 11.3(b) shows the impact of the SCV of the interarrival times on the accuracy. Especially for systems with more than 2 queues the accuracy is very satisfactory, in particular for the case $cv^2_{A_i} = 1$. In Table 11.3(c) the impact of imbalance in a polling system on the accuracy is depicted, and, as could be expected, it can be concluded that a high imbalance in either service or interarrival times has a considerable, negative, impact on the approximation accuracy. Polling systems with more than 2 queues are much less bothered by this imbalance than polling systems with only 2 queues.

N	$0-5\%$	$5-10\%$	$10-15\%$	$15-20\%$
2	86.46	10.24	2.78	0.52
3	98.84	1.16	0.00	0.00
4	99.78	0.22	0.00	0.00
5	99.93	0.07	0.00	0.00

Table 11.2: Errors of the approximation applied to the 2304 test cases with exhaustive service, as described in Section 11.4, categorized in bins of 5%.

N	Load (ρ)					
	0.10	0.30	0.50	0.70	0.90	0.99
2	0.31	1.81	3.41	4.17	2.70	0.67
3	0.16	0.84	1.44	1.69	1.07	0.39
4	0.13	0.68	1.14	1.28	0.73	0.25
5	0.11	0.57	0.94	1.03	0.57	0.22

(a)

N	SCV interarrival times ($cv^2_{A_i}$)		
	0.25	1	2
2	2.27	1.76	2.50
3	1.36	0.52	0.92
4	1.13	0.29	0.69
5	0.97	0.19	0.56

(b)

N	Imbalance interarrival and service times			
	$I_{A_i}=1$, $I_{B_i}=1$	$I_{A_i}=1$, $I_{B_i}=5$	$I_{A_i}=5$, $I_{B_i}=1$	$I_{A_i}=5$, $I_{B_i}=5$
2	0.69	2.92	2.80	2.30
3	0.65	1.27	0.75	1.06
4	0.56	0.89	0.62	0.73
5	0.49	0.69	0.53	0.59

(c)

Table 11.3: Mean relative approximation error, categorized by number of queues (N) and total load of the system (a), SCV interarrival times (b), and imbalance of the interarrival and service times (c).

11.4.3 Miscellaneous other cases

More queues. In this subsection we discuss several cases that are left out of the test bed because they might not give any new insights, or because the combination of parameter values might be rarely found in practice. Firstly, we discuss polling systems with more than 5 queues briefly. Without listing the actual results, we mention here that the approximations become more and more accurate when letting N grow larger, and still varying the other parameters in the same way as is described in Table 11.1. For $N = 10$ already, all relative errors are less than 5%, with an average of less than 0.5%, and it only gets smaller as N grows further.

More variation in service times and switch-over times. In the test bed we only use SCVs 0.25 and 1 for the service times and switch-over times, because these seem more relevant from a practical point of view. As the coefficient of variation grows larger, our approximation will become less accurate. E.g., for Poisson arrivals we took $cv^2_{B_i} \in \{2, 5\}$, $cv^2_{S_i} \in \{2, 5\}$, and varied the other parameters as in our test bed (see Table 11.1). This way we reproduced Table 11.2. The result is shown in Table 11.4 and indicates that the quality of our approximation deteriorates in these extreme cases. The mean relative errors for $N = 2, \dots, 5$ are respectively 3.58%, 1.78%, 1.07%, and 0.77%, which is still very good for systems with such high variation in service times and switch-over times. For non-Poisson input, no investigations were carried out because the results are expected to show the same kind of behaviour.

N	$0-5\%$	$5-10\%$	$10-15\%$	$15-20\%$
2	74.22	14.84	6.51	2.08
3	89.76	7.29	2.08	0.69
4	94.53	4.56	0.91	0.00
5	97.71	2.19	0.10	0.00

Table 11.4: Errors of the approximation applied to the 768 test cases with Poisson arrival processes and high SCVs of the service times and switch-over times, categorized in bins of 5%.

Small switch-over times. Systems with small switch-over times, in particular smaller than the mean service times, also show a deterioration of approximation accuracy, especially in systems with 2 queues. In Figure 11.2 we show an extreme case with $N = 2$, service times and switch-over times are exponentially distributed with $\mathbb{E}[B_i] = 9/40$ and $\mathbb{E}[S_i] = 9/200$ for $i = 1, 2$, which makes the mean switch-over times 5 times *smaller* than the mean service times. Furthermore, the interarrival times are exponentially distributed with $\lambda_1 = 5\,\lambda_2$. In Figure 11.2 the mean waiting times of customers in both queues are plotted versus the load of the system. Both the approximation and the exact values are plotted. For customers in Q_1 the mean waiting time approximations underestimate the true values, which leads to a maximum relative error of -11.2% for $\rho = 0.7$ ($\mathbb{E}[W_{1,app}] = 0.43$, whereas $\mathbb{E}[W_1] = 0.49$). The approximated mean waiting time for customers in Q_2 is systematically overestimating the true value. The maximum relative error is attained at $\rho = 0.5$ and is 28.8% ($\mathbb{E}[W_{1,app}] = 0.41$, whereas $\mathbb{E}[W_1] = 0.52$). Although the relative errors are high in this situation, the absolute errors are still rather small compared to the mean service time of an individual customer. This implies that the mean *sojourn time* is already much better approximated. Nevertheless, this example illustrates one of the situations where our approximation gives unsatisfactory results.

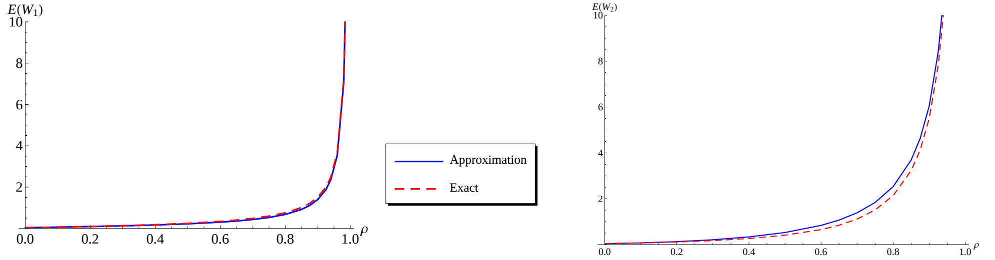


Figure 11.2: Approximated and exact mean waiting times for a two-queue polling system with small switch-over times.

11.4.4 Comparison with existing approximations

For non-exponential interarrival times hardly any good alternative approximations exist. In [149, 187] it is suggested to use the HT limit (11.3.13) as an approximation, but the accuracy is only found to be acceptable for $\rho > 0.8$. Another approximation for the mean waiting time in polling systems with non-exponential interarrival times uses the limit for $S \to \infty$, see [207, 209]. This approximation is usable if either the total setup time in the system is large and the setup times have low variance, or the total setup time in the system is large and the system is in heavy traffic. The approximation discussed in the present chapter is exact in all these limiting cases, but performs much better for systems under less extreme conditions. This makes our approximation the only one which can be applied under all circumstances.

For polling systems with Poisson arrivals, several alternative approximations have been developed in existing literature. The best one among them (see e.g. [31, 72, 92]) uses the relation $\mathbb{E}[W_i] = (1 \pm \rho_i)\mathbb{E}[C_i^{res}]$, where C_i is the cycle time, starting at a visit *completion* to Q_i when service is exhaustive, and starting at a visit *beginning* for gated service. By $\pm$ we mean $-$ for exhaustive service, and $+$ for gated service. The mean residual cycle time, $\mathbb{E}[C_i^{res}]$, is assumed to be equal for all queues, i.e. $\mathbb{E}[C_i^{res}] \approx \mathbb{E}[C^{res}]$, and can be found by substituting $\mathbb{E}[W_i] \approx (1 \pm \rho_i)\mathbb{E}[C^{res}]$ in the pseudo-conservation law (11.3.14). We have used this PCL-based approximation to estimate the mean waiting times of all queues in the test bed described in Table 11.1, but taking only the 768 cases where $cv^2_{A_i} = 1$. Table 11.5 shows the mean relative errors for our approximation (a) and the PCL approximation (b), categorized in bins of 5% as was done before in Table 11.2. From these tables (and from other performed experiments that are not mentioned for the sake of brevity) it can be concluded that for $N > 2$ both approximation have almost the same accuracy, our approximation being slightly better for small values of ρ, and the PCL approximation being slightly better for high values of ρ (both methods are asymptotically exact as $\rho \uparrow 1$). However, for $N = 2$ our method suffers greatly from imbalance in the system, whereas the PCL approximation proves to be more robust.

N	$0-5\%$	$5-10\%$	$10-15\%$
2	89.32	9.11	1.56
3	100.00	0.00	0.00
4	100.00	0.00	0.00
5	100.00	0.00	0.00

(a)

N	$0-5\%$	$5-10\%$	$10-15\%$
2	96.09	2.86	1.04
3	99.31	0.69	0.00
4	100.00	0.00	0.00
5	100.00	0.00	0.00

(b)

Table 11.5: Errors of the approximation applied to the 768 test cases with Poisson input, categorized in bins of 5%. In (a) the percentages of mean relative errors in each bin are shown for our approximation, in (b) results are shown for the PCL approximation.

11.4.5 Gated service

Until now we have only shown and discussed approximation results for polling systems with exhaustive service. The complete test bed described in Table 11.1 has also been analyzed for polling systems where each queue receives gated service. As can be seen in Table 11.6, the overall quality of the approximation is good, but worse than for polling systems with exhaustive service. More details on the reason for these inaccuracies can be found in Table 11.7, which is the equivalent of Table 11.3 for gated service. Table 11.7(b) illustrates that there is now a huge difference between systems with Poisson arrivals, and systems with non-Poisson arrivals. For the cases with $cv^2_{A_i} = 1$, the approximation is extremely accurate, even for two-queue polling systems. The accuracy in cases with $cv^2_{A_i} \neq 1$ is worse, which is caused by the assumptions that are made to approximate the LT limit (11.3.11). Firstly, the decomposition (11.3.2) does not hold for non-Poisson arrivals, and secondly, the terms $\mathbb{E}[I_i^{res}]$ and $\mathbb{E}[V_i I_i]/\mathbb{E}[I_i]$ in this decomposition have only been approximated. For exhaustive service, these assumptions do not have much negative impact on the accuracy, but apparently, for gated service, they do. The mean relative errors for $N = 2, \ldots, 5$ queues are respectively 2.70%, 2.25%, 1.90%, and 1.63%. The imbalance of the mean interarrival and service times hardly influences the accuracy of the approximation, as can be concluded from Table 11.7(c).

If we consider the 768 cases with Poisson arrivals only, the mean relative errors of our approximation for $N = 2, \ldots, 5$ are respectively 0.34%, 0.17%, 0.10%, and 0.08%. This accuracy is even better than the one achieved by the PCL approximation.

N	$0-5\%$	$5-10\%$	$10-15\%$	$15-20\%$
2	82.55	12.33	2.95	1.56
3	85.42	10.53	3.13	0.81
4	88.85	8.46	2.43	0.26
5	92.22	6.60	1.15	0.03

Table 11.6: Errors of the approximation applied to the 2304 test cases with gated service, as described in Section 11.4.5, categorized in bins of 5%.

11.5 Further research topics

The research that is done in the present chapter can be extended in many different directions.

Other service disciplines. In the present chapter, only exhaustive and gated service are discussed. In order to obtain results for polling systems with some queues receiving exhaustive service, and others receiving gated service, only minor modifications should be made. It would be more challenging to generalize the approximation to a wider variety of service disciplines. In particular, it would be nice to have one expression for the mean waiting time of customers in a queue with an arbitrary branching-type service discipline (cf. [160]). The *exhaustiveness* of a branching-type service discipline (cf. [208]) might

N	Load (ρ)					
	0.10	0.30	0.50	0.70	0.90	0.99
2	2.64	4.55	4.31	3.10	1.25	0.37
3	2.03	3.78	3.68	2.68	1.04	0.30
4	1.62	3.14	3.13	2.32	0.92	0.28
5	1.35	2.67	2.71	2.03	0.81	0.21

(a)

N	SCV interarrival times ($cv^2_{A_i}$)		
	0.25	1	2
2	4.72	0.34	3.05
3	4.06	0.17	2.53
4	3.45	0.10	2.16
5	2.98	0.08	1.84

(b)

N	Imbalance interarrival and service times			
	$I_{A_i} = 1$, $I_{B_i} = 1$	$I_{A_i} = 1$, $I_{B_i} = 5$	$I_{A_i} = 5$, $I_{B_i} = 1$	$I_{A_i} = 5$, $I_{B_i} = 5$
2	2.76	2.64	2.81	2.59
3	2.28	2.25	2.27	2.21
4	1.93	1.91	1.90	1.87
5	1.64	1.66	1.64	1.58

(c)

Table 11.7: For gated service: mean relative approximation error, categorized by number of queues (N) and total load of the system (a), SCV interarrival times (b), and imbalance of the interarrival and service times (c).

appear in this expression. Gated and exhaustive are both branching type service disciplines, but are discussed separately in the present chapter. The HT limit can most likely be established for arbitrary branching type service disciplines (see conjectures in [149]), so the question that remains is whether the LT limit can be found in a similar way.

Optimization. One of the main reasons to choose (11.2.1) as form of the interpolation, besides its asymptotic correctness, is its simplicity. Having this exact and simple expression for the approximate mean waiting times, makes it very useful for optimization purposes. In production environments, one can, for example, determine what the optimal strategy is to combine orders of different types (i.e., determine what queue customers should join). Because general arrivals are supported, one can determine optimal sizes of batches in which items are grouped and sent to a specific machine. The simplicity of (11.2.1) makes it possible for a manager to create a handy Excel sheet that can be used by operators to compute all kind of optimal parameter settings. No difficult computations are required at all, so a large variety of users can use the approximation.

In the present chapter the accuracy of the approximation has been investigated and has been found to be very good in most situations. Another advantage of our approximation regarding optimization purposes, is that the general shape of the approximated curve follows the exact curve very closely. Even in cases where the relative errors are rather large, like in Figure 11.1, the shape of the actual curves is still very well approximated. This means that plugging our approximation, instead of an exact expression if it had been available, in an optimization function yields an optimum that should be close to the true optimum.

Polling Table. The interpolation based approximation can also be extended to polling systems where the visiting order of the queues is not cyclic. Waiting times in polling systems with so-called polling tables can be obtained in the same way as shown in the present chapter. Both the LT and HT limits are not difficult to determine in this situation, and the interpolation follows directly from these limits.

Model. The form of the interpolation might be changed to improve the accuracy of approximations for cases that give less satisfactory results in the present form. E.g., one could try other functions than a second-order polynomial as numerator of (11.2.1). Alternatively, one could try to find a correction term which could be added to (11.2.1) to obtain better results for, e.g., two-queue polling systems. But most of all, if an *exact* LT limit of the mean waiting time in a polling system with non-Poisson arrivals could be found, the accuracy of the approximation in the case of gated service might be improved.

Distributions. In Dorsman et al. [64], an approximation is derived for the waiting-time distributions. Their approach uses the results derived in this chapter. They show their approximation to be highly accurate over a wide range of parameter settings.

Batch services. In Dorsman et al. [65] polling systems with batch services are studied. In optimizing the batch sizes, a cost structure that is a function of the mean waiting times is used. They use a similar approach for approximating the mean waiting times as in this chapter, however, they judge the form of the approximation in (11.2.1) to be too complex for their purposes. Instead, they use a first order polynomial in the numerator: $\mathbb{E}[W_{i,app}] = (a + b_i\,\rho)/(1-\rho)$.

Polling: Fairness and efficiency

12

κ-Gated

We study a polling model where we want to achieve a balance between the fairness of the waiting times and the efficiency of the system. For this purpose, we introduce a novel service discipline: the *κ-gated service discipline*. It is a hybrid of the classical gated and exhausted disciplines, and consists of using κ_i consecutive gated service phases at Q_i before the server switches to the next queue. The advantage of this discipline is that the parameters κ_i can be used to balance fairness and efficiency. We derive the distributions and means of the waiting times, a pseudo conservation law for the weighted sum of the mean waiting times, and the fluid limits of the waiting times. Our goal is to optimize the κ_i's so as to minimize the differences in the mean waiting times, i.e. to achieve maximal fairness, without giving up too much on the efficiency of the system. From the fluid limits we derive a heuristic rule for setting the κ_i's. In a numerical study the heuristic is shown to perform well in most cases.

12.1 Introduction

Polling models are used in the modeling of many problems, for example computer systems, maintenance systems and telecommunication. In these models, multiple queues are served by a single server, which cyclically visits the queues. A typical performance measure in such systems is the mean waiting time at each of the queues. In certain applications (see e.g. [152, 185]) it is important to maintain *fairness*, in the sense of the queues having (almost) equal mean waiting times. In achieving this, one usually has to sacrifice the efficiency of the system. In this chapter, however, we introduce a strategy which on the one hand achieves fairness, while on the other hand is still efficient. Here, the efficiency is given by the sum of the mean waiting times, weighted by the utilization rates, and fairness is understood as the maximal difference in the mean waiting times at each of the queues. In the literature, multiple meanings have been associated to fairness, e.g. serving customers in order of arrival (see [9, 40], where [9] is a survey on the matter of fairness). These interpretations, however, are different from the fairness considered here.

In a polling model, when the server switches to the next queue, a switch-over time is incurred. There are many possible choices for deciding when the server should switch to the next queue. The rules studied most often are the exhaustive service discipline (when the server arrives at a queue, it serves its customers until the queue has become empty) and the gated service discipline (when the server arrives at a queue, a gate closes and only the customers who are before the gate, i.e., who are already present, will be served in this server visit).

The main advantage of the exhaustive strategy, is that it is optimally efficient. That is, it minimizes the sum of the mean waiting times at the queues weighted by their utilization rates. However, the differences between mean waiting times at the queues might be large. Typically, the heaviest loaded one has the smallest mean waiting time in this discipline. Conversely, the gated discipline leads in general to much smaller differences. But this is at the expense of the efficiency, which is much lower for this discipline. We aim to combine the best of both worlds into a new service discipline, by introducing a hybrid version of exhaustive and gated: the *κ-gated service discipline*.

The κ-gated discipline consists of using κ_i consecutive gated service phases at queue i before the server switches to the next queue. That is, upon arrival of the server, it serves the queue consecutively (at most) κ_i times, according to the gated discipline. So upon arrival of the server, a first gate closes and only the customers before this gate are served. After this, a second gate closes, and again only the customers before this gate are served, etcetera. This is done κ_i times, or until the queue becomes empty. The parameters κ_i are specified in the vector $\kappa = (\kappa_1, \ldots, \kappa_N)$, where N is the number of queues. Note that when $\kappa_i = 1$, queue i is served according to the gated discipline; when $\kappa_i \to \infty$, queue i is served according to the exhaustive discipline (as it is served until it becomes empty). One of the main questions studied in the current chapter is whether the κ_i's can be optimized as to achieve both fairness and efficiency.

Fairness has frequently played a role in the choice of a service discipline in polling systems. For example, motivated by a dynamic bandwidth allocation problem of Ethernet Passive Optical Networks (EPON), in [152, 185] a *two-stage gated* service discipline is studied. In that case, a gate closes behind the customers in a stage-1 buffer at the moment the server arrives, the customers in the stage-2 buffer are being served, and then those present in stage-1 move to the stage-2 buffer. This was seen to give rise to relatively small differences between mean waiting times at the various queues, but at the expense of longer delays, i.e., at the expense of the efficiency of the system. The strategy was later generalized to multi-phase gated (see [186]). The κ-gated discipline can be seen as a variant of this discipline, where we have removed the extra cycles all customers have to wait for, in between moving to the next stage buffer. Hence, we expect it to lead to small differences between mean waiting times as well, but with significantly smaller total mean delays than for two- or multi-stage gated.

Besides the two- and multi-stage gated disciplines, a number of other disciplines have been proposed in the literature in order to achieve fairness (in the sense considered here). We mention a few in the following. Altman, Khamisy and Yechiali [3] (see also Shoham and Yechiali [167]) consider a so-called elevator strategy in a globally gated regime. In this setting the queues are visited in the order: $1, 2, \ldots N-1, N, N, N-1, \ldots 2, 1, 1, 2, \ldots$ etc. When the server turns around at queue 1 or queue N, a gate closes at all queues: only those before the gate are served. This strategy turns out to be perfectly fair. However, it is far less efficient because of the globally gated regime. Our focus here is on cyclic models. Boxma, Van Wijk and Adan [41] introduce the Gated/Exhaustive discipline (see Chapter 13): the queues are visited cyclically, where in one cycle alternately all queues are served according to the gated discipline or all queues are served exhaustively. The incentive for this mixed strategy arose from the well-known expressions for the mean waiting time of queue i for gated respectively exhaustive systems: $\mathbb{E}[W_i^{gat}] = (1+\rho_i)\mathbb{E}[C_i^{res}]$ respectively $\mathbb{E}[W_i^{exh}] = (1-\rho_i)\mathbb{E}[C_i^{*res}]$, where ρ_i is the workload. Furthermore $\mathbb{E}[C_i^{res}]$ and $\mathbb{E}[C_i^{*res}]$ denote the mean residual cycle duration in case the cycle is assumed to start at the visit completion, respectively visit beginning of Q_i. These can be approximated by $\mathbb{E}[C^{res}] \approx \mathbb{E}[C_i^{res}] \approx \mathbb{E}[C_i^{*res}]$. From the resulting approximations for $\mathbb{E}[W_i^{gat}]$ and $\mathbb{E}[W_i^{exh}]$, one might expect the mean waiting time in the Gated/Exhaustive discipline to become $\mathbb{E}[W_i^{G/E}] \approx \mathbb{E}[C^{res}]$, which does not depend on i. However, it turns out that this guess is incorrect, as the exhaustive cycle dominates in the mean waiting times. The difference in mean waiting times only marginally decreases compared to exhaustive. To overcome this, [41] proposes the use of a polling table (see also [17, 205]), which prescribes the order in which queues are visited. This is related to [39], in which efficient visit orders are studied. Another option are efficient visit frequencies, see [38]. These options, however, do not focus on fairness.

Our contribution in this chapter is as follows. We introduce the κ-gated discipline. Our motivation for this novel discipline is the search for a policy that achieves almost equal mean waiting times at the queues (fairness), without giving up too much of the efficiency. In earlier work in the literature, the focus has been solely on fairness, leading to inefficient disciplines [3, 185], whereas the advantage of the κ-gated discipline is that its parameter κ can be used to balance fairness and efficiency. For the κ-gated discipline we derive the distributions and means of the waiting times, a pseudo conservation law for the weighted sum of the mean waiting times, and the fluid limits of the waiting times. We want to set the κ_i's so as to achieve maximal fairness without giving up too much on the efficiency of the system. To accomplish this, we use the fluid limits to derive a heuristic for setting κ. Finally, in a numerical study we extensively test the performance of the heuristic. It turns out to perform well in most cases.

The structure of this chapter is as follows. In Section 12.2 we introduce the model in more detail and give the notation that is being used. In Section 12.3 we derive the mean visit times at the queues, a *Pseudo Conservation Law* for the weighted sum of the mean waiting times, the waiting time distributions at all queues using *Multi-type Branching Processes*, the mean waiting times using the *Mean Value Analysis* technique exploiting the concept of *Smart Customers*, and the *Fluid Limits* of the waiting times. In Section 12.4 we derive a heuristic rule for the setting of κ based on the fluid limits. Section 12.5 contains examples and a numerical study into the performance of the heuristic. We end with a conclusion and discussion of possible further work in Section 12.6. This chapter is based on [192].

12.2 Model and notation

We consider a polling model as described in Section 9.2, using a new service discipline at each of the queues: the *κ-gated service discipline*. This discipline works as follows. Upon arrival at Q_i, the server serves exactly those customers present on arrival (phase 1); when this is done, it serves exactly those customers present in Q_i at that moment (phase 2); and so on, until (at most) κ_i phases are completed, and then the server switches to the next queue. If the queue is empty at the start of a phase, the server also switches. This discipline consists of the prescription of $\kappa = (\kappa_1, \ldots, \kappa_N)$, with $\kappa_i \in \{1, 2, \ldots\} \cup \{\infty\}$ for all

$i = 1,\ldots,N$. For $\kappa_i = 1$ the discipline at Q_i is equivalent to the gated service discipline, and for $\kappa_i = \infty$ it is equivalent to the exhaustive service discipline, as the queue is served until it becomes empty. It is readily verified that the condition $\rho < 1$ is also necessary and sufficient for the stability in case of the κ-gated service discipline.

We want to achieve fairness in the waiting times, that is, we want the $\mathbb{E}[W_i]$ for $i = 1,\ldots,N$ to be (almost) equal. Hence, we want to minimize

$$\max_{i,j}\big(\mathbb{E}[W_i] - \mathbb{E}[W_j]\big).$$

On the other hand, we do not want to give up too much of the efficiency of the system. For the efficiency, we use the weighted sum over all mean waiting times:

$$\sum_{i=1}^{N}\rho_i\mathbb{E}[W_i].$$

This is a measure for the total workload in the system: it is the expected value of the waiting work in the system at an arbitrary moment. Hence, we focus on the following performance characteristic of the system:

$$\tilde{\gamma}(\alpha) := (1-\alpha)\max_{i,j}\big(\mathbb{E}[W_i] - \mathbb{E}[W_j]\big) + \alpha\sum_{i=1}^{N}\rho_i\mathbb{E}[W_i], \tag{12.2.1}$$

for some $\alpha \in [0,1]$. The expression in (12.2.1) represents the trade-off between fairness and efficiency, by assigning $100(1-\alpha)\%$ of the importance to fairness and the remaining $100\alpha\%$ to efficiency. Note that (12.2.1) depends on the service discipline at each of the queues. Under the κ-gated discipline, for a given α, the κ can be optimized to minimize $\tilde{\gamma}$. This optimization is a trade-off between the fairness (maximal difference in mean waiting times) and the efficiency (weighted sum of mean waiting times). One can distinguish two extreme cases. For $\alpha = 0$, only the fairness of the discipline counts. In that case, the elevator strategy in a globally gated regime (cf. [3, 167]) is the best choice, as it leads to equal mean waiting times. For $\alpha = 1$, only the efficiency of the system is important. The exhaustive discipline is optimal in that case. We remark that for the term measuring the efficiency, a so-called pseudo conservation law holds, and it is easily determined without having to calculate all individual mean waiting times (see Section 12.3.2).

12.3 Analysis of the κ-Gated Discipline

In this section we present the analysis of the κ-gated discipline. First, we derive the mean visit times at each of the queues. Then, we give a pseudo conservation law for the weighted sum of the mean waiting times. Next, we present the derivation of the waiting time distributions, using multi–type branching processes. Following that, we briefly indicate a simpler way to compute the mean waiting times. For this, we show that the discipline fits into the framework of smart customers, and then we apply mean value analysis for polling models. We end this section by presenting the fluid limits of the waiting times. These fluid limits are used in the next section to derive a heuristic for the optimal setting of κ.

12.3.1 Mean Visit Times

For the κ-gated discipline, we derive the expected duration of each of the visits and visit phases to a queue. The expected cycle duration is $\mathbb{E}[C] = \mathbb{E}[S]/(1-\rho)$. A fraction ρ_i of the cycle the server is working on Q_i, hence the expected duration of a visit to Q_i, denoted by $\mathbb{E}[V_i]$, is given by $\mathbb{E}[V_i] = \rho_i\mathbb{E}[C]$. This gives that the mean intervisit time, denoted by $\mathbb{E}[I_i]$, is given by $\mathbb{E}[I_i] = (1-\rho_i)\mathbb{E}[C]$. To further specify the visit times, let $\mathbb{E}[V_i^k]$ be the mean visit time of phase k at Q_i, for $k = 1,\dots,\kappa_i$. Then $\mathbb{E}[V_i] = \sum_{k=1}^{\kappa_i}\mathbb{E}[V_i^k]$. In the first phase, all work that arrived during the last phase of the previous cycle and the intervisit time has to be served. This gives for its mean duration:

$$\mathbb{E}[V_i^1] = \rho_i(\mathbb{E}[V_i^{\kappa_i}] + \mathbb{E}[I_i]). \tag{12.3.1}$$

In the second phase, the work that arrived during the first phase is served; in the third phase that of the second, and so on. This leads to:

$$\begin{aligned}\mathbb{E}[V_i^k] &= \rho_i\,\mathbb{E}[V_i^{k-1}]\\ &= \rho_i^{k-1}\mathbb{E}[V_i^1], \quad \text{for } k = 2,\dots,\kappa_i.\end{aligned}$$

Substituting this expression for $k = \kappa_i$ into (12.3.1) gives $\mathbb{E}[V_i^1] = \rho_i\big(\rho_i^{\kappa_i-1}\mathbb{E}[V_i^1] + (1-\rho_i)\mathbb{E}[C]\big)$. Solving this leads to

$$\mathbb{E}[V_i^1] = \rho_i\frac{1-\rho_i}{1-\rho_i^{\kappa_i}}\mathbb{E}[C],$$

and hence

$$\mathbb{E}[V_i^k] = \rho_i^k\frac{1-\rho_i}{1-\rho_i^{\kappa_i}}\mathbb{E}[C], \quad k = 1,\dots,\kappa_i. \tag{12.3.2}$$

Note that the mean duration of subsequent phases decreases, as is to be expected. It is readily verified that with (12.3.2), it indeed holds that: $\displaystyle\sum_{k=1}^{\kappa_i}\mathbb{E}[V_i^k] + \mathbb{E}[I_i] = \mathbb{E}[C]$, for $i = 1,\dots,N$.

12.3.2 Pseudo Conservation Law

Boxma and Groenendijk [34] derive a so-called Pseudo Conservation Law (PCL) for the case of cyclic order polling systems (see Section 9.3.1):

$$\sum_{i=1}^N \rho_i\mathbb{E}[W_i] = \frac{\rho}{1-\rho}\sum_{i=1}^N \rho_i\mathbb{E}[B_i^{res}] + \rho\mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)}\left(\rho^2 - \sum_{i=1}^N \rho_i^2\right) + \sum_{i=1}^N \mathbb{E}[M_i], \tag{12.3.3}$$

where $\mathbb{E}[M_i]$ is the mean amount of work in Q_i at a departure epoch of the server from Q_i. This is the only term that depends on the service discipline at the queues. For the exhaustive discipline $\mathbb{E}[M_i^{exh}]$ trivially equals zero (cf. [34, (3.11)]), and for gated it holds that $\mathbb{E}[M_i^{gat}] = \rho_i\mathbb{E}[V_i] = \rho_i^2\mathbb{E}[S]/(1-\rho)$ (cf. [34, (3.12)]). The workload decomposition result in [34] is also valid for the κ-gated discipline, and we find, using (12.3.2):

$$\mathbb{E}[M_i^{\kappa\text{-}gat}] = \rho_i\mathbb{E}[V_i^{\kappa_i}] = \rho_i^{\kappa_i+1}\frac{1-\rho_i}{1-\rho_i^{\kappa_i}}\frac{\mathbb{E}[S]}{1-\rho}.$$

Remark that for the two extreme cases $\kappa_i = 1$ and $\kappa_i = \infty$ this expression simplifies to that of the gated respectively exhaustive discipline.

Comparing the term $\mathbb{E}[M_i]$ for (κ-) gated and exhaustive, we find the following:

$$0 = \mathbb{E}[M_i^{exh}] \leq \mathbb{E}[M_i^{\kappa\text{-}gat}] \leq \mathbb{E}[M_i^{gat}],$$

with equality for the first '$\leq$' if and only if $\kappa_i = \infty$, and equality for the second '$\leq$' if and only if $\kappa_i = 1$. Exhaustive is the most efficient service discipline, as the server never switches when there are still customers in the queue it is serving. So, it leaves no customers behind that have to wait for an entire cycle. Under the (κ-) gated discipline, however, customers may be left behind. Gated (i.e. $\kappa_i = 1$) is less efficient than κ-gated for $\kappa_i \geq 2$, since more customers will be left behind when the server switches to the next queue. It follows that the efficiency of the κ-gated discipline is always between that of exhaustive and gated.

By substituting the expression for $\mathbb{E}[M_i^{\kappa\text{-}gat}]$ into (12.3.3) we find the pseudo conservation law for the κ-gated discipline:

$$\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] = \rho \frac{\sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}]}{1-\rho} + \rho \mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)} \left(\rho^2 - \sum_{i=1}^{N} \rho_i^2 \right)$$
$$+ \sum_{i=1}^{N} \rho_i^{\kappa_i+1} \frac{1-\rho_i}{1-\rho_i^{\kappa_i}} \frac{\mathbb{E}[S]}{1-\rho}.$$

As only the terms $\mathbb{E}[M_i]$ depend on the service discipline (and hence on κ for the κ-gated discipline), we can restrict our attention to $\sum_{i=1}^{N} \mathbb{E}[M_i]$ instead of $\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i]$. So in the sequel, instead of (12.2.1), we concentrate on optimizing:

$$\gamma(\alpha) := (1-\alpha) \max_{i,j} \big(\mathbb{E}[W_i] - \mathbb{E}[W_j] \big) + \alpha \sum_{i=1}^{N} \mathbb{E}[M_i], \qquad (12.3.4)$$

for some $\alpha \in [0,1]$.

12.3.3 Waiting time distributions

We determine the Laplace–Stieltjes transform (LST) of the waiting times W_i analogously to Resing [160] (as described in Section 9.3.3). In [160] it is shown, that if the service discipline in each queue satisfies the *branching property* ([160, Property 1]), then the queue length process at polling instants of a fixed queue is a multi-type branching process (MTBP) with immigration in each state. The κ-gated service discipline does satisfy the branching property. Let the start of the visit to Q_1 be the start of the cycle, then by the branching property, each customer present will during the cycle be replaced in an i.i.d. manner by customers of type $1, \ldots, N$, according to the probability generating function (pgf) $h_i(z)$, where $z = (z_1, \ldots, z_N)$. For the gated service discipline, this h_i is given by:

$$h_i^{(gated)}(z) = \beta_i \left(\sum_{j=1}^{N} \lambda_j (1 - z_j) \right).$$

For κ-gated we can recursively express h_i as follows:

$$h_i^{\textit{(1-gated)}}(z) = h_i^{\textit{(gated)}}(z),$$
$$h_i^{\textit{(m-gated)}}(z) = \beta_i \left(\sum_{j=1, j \neq i}^{N} \lambda_j (1 - z_j) + \lambda_i \left(1 - h_i^{\textit{((m-1)-gated)}}(z) \right) \right), \text{ for } m = 2, 3, \ldots.$$

For $\kappa_i = \infty$, the pgf h_i coincides with that of the exhaustive service discipline, which is given by:

$$h_i^{\textit{(∞-gated)}}(z) = h_i^{\textit{(exhaustive)}}(z) = \theta_i \left(\sum_{j=1, j \neq i}^{N} \lambda_j (1 - z_j) \right),$$

where $\theta_i(\cdot)$ is the LST of a busy period triggered by one type i customer in Q_i in isolation. Now, along the lines outlined in Section 9.3.3, the LSTs of the steady-state waiting time distributions can be derived. Then, by differentiation, moments of the steady-state waiting time for an arbitrary type i customer follow. These calculations are straightforward, but cumbersome. The next section explains an intuitive approach to calculate the first moments of the waiting times.

12.3.4 Mean waiting times

We briefly discuss how the first moments of the waiting times, $\mathbb{E}[W_i]$, can easily be obtained in a more efficient way. For this, we show that the κ-gated discipline fits into the framework of a polling model with smart customers (see Chapter 10). Hence, we can use mean value analysis (MVA) for polling systems, adapted for smart customers (cf. Section 10.3). Recall that in a model with smart customers, the arrival rate at a queue depends on the position of the server.

We use the concept of smart customers to route arriving customers to a specific queue, depending on the position of the server. Let the arrival rate be $\lambda_{i,j}$ at Q_i when the server is serving (or switching to) Q_j. The routing proceeds in the following way. We introduce a polling model with the gated discipline that is related to the one served according to the κ-gated discipline. In that model, we create multiple copies of the same queue. We refer to this as the corresponding model, in which customers are routed as follows. A customer arriving at Q_i in the original model is routed in the corresponding model to the copy of Q_i that will be served first. The underlying idea of this is the following. In the κ-gated model, arriving customers queue behind a gate, which only opens when the server starts one of the κ_i serving phases. In the corresponding model, each of these phases now becomes a separate queue. Hence, we create a polling model with κ_i copies of queue Q_i, denoted by $Q_i^{(1)}, \ldots, Q_i^{(\kappa_i)}$. No switch-over times are incurred between these copies. Denoting phase k of a visit to Q_i by $V_i^{(k)}$, then the cycle, including the switch-over times S_i (between $Q_i^{(\kappa_i)}$ and $Q_{i+1}^{(1)}$) becomes:

$$V_1^{(1)} - V_1^{(2)} - \ldots - V_1^{(\kappa_1)} - S_1 - V_2^{(1)} - V_2^{(2)} - \ldots - V_2^{(\kappa_2)} - S_2 - \ldots - S_{N-1} - V_N^{(1)} - \ldots - V_N^{(\kappa_N)} - S_N.$$

We now have an 'ordinary' cyclic polling model with $\sum_{i=1}^{N} \kappa_i$ queues, each of which is served according to the gated discipline. We want this system to have the same arrival process as the original one. For that, we have to route the arriving customers, depending

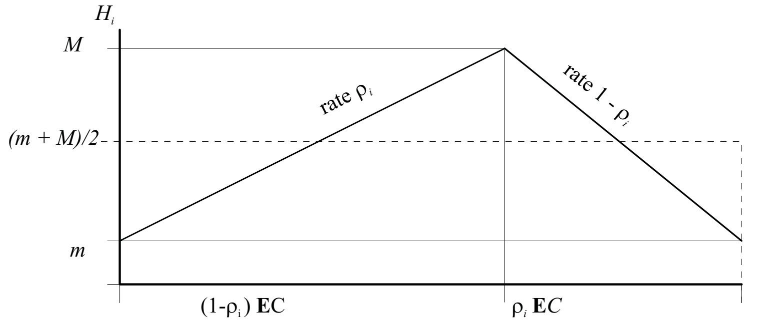


Figure 12.1: The fluid limit of the workload H_i at Q_i during one cycle.

on the position of the server. A customer arriving at Q_i in the original model is now routed to $Q_i^{(j)}$ during $V_i^{(j-1)}$, for $j = 2, \ldots, \kappa_i$, and to $Q_i^{(1)}$ otherwise.

The corresponding model is a polling model with smart customers, in which arriving type i customers are routed to $Q_i^{(j+1)}$ while the server is at $Q_i^{(j)}$, for $1 \leq j < \kappa_i$, and to $Q_i^{(1)}$ otherwise. In Section 10.3, we showed how a system of $\mathcal{O}\left(N^2\right)$ linear equations can be derived for an N queue polling model with the exhaustive service discipline, from which the $\mathbb{E}[W_i]$ can immediately be solved. Analogously, we can write down a system of $\mathcal{O}\left((\sum_{i=1}^{N} \kappa_i)^2\right)$ linear equations for the corresponding model with gated service, from which the $\mathbb{E}[W_i]$ directly follow.

REMARK 12.3.1. Boon et al. [28] also present the MTBP approach for polling models with smart customers. In the case that some of the arrival rates equal zero, they have to introduce extra queues requiring zero service times. However, by the structure of the κ-gated discipline, the MTBP analysis can be reduced to that presented in Section 12.3.3.

12.3.5 Fluid limits

The exact expressions for the mean waiting times, following from Sections 12.3.3 and 12.3.4, do not provide an easy way to determine the κ_i's minimizing $\gamma(\alpha)$. Therefore, we derive the fluid limit approximations of the mean waiting times. These approximations yield closed-form expressions, and can hence easily be used to (approximately) optimize the κ_i's.

By taking the fluid limits, we scale the interarrival and service times. For this, we let $\lambda_i \to \infty$ and $\mathbb{E}[B_i] \to 0$ while keeping the workload $\lambda_i\,\mathbb{E}[B_i] = \rho_i$ fixed. We concentrate on the amount of work present at a queue, denoted by H_i at queue Q_i. By the use of this scaling, we smoothen the discrete process H_i into a continuous one. In this way, work arrives at a constant rate ρ_i, and during the visit time work is removed at rate 1. So, during the intervisit time of mean length $\mathbb{E}[I_i] = (1-\rho_i)\mathbb{E}[C]$, the amount of work increases at rate ρ_i, and during the visit time, with mean length $\mathbb{E}[V_i] = \rho_i\,\mathbb{E}[C]$, the amount of work decreases at rate $1-\rho_i$. This cyclic pattern repeats itself in every cycle. Hence, the workload H_i during a cycle in the κ-gated discipline becomes as depicted in Figure 12.1.

At the *end* of the visit to Q_i, the amount of work present is equal to that built up during the last visit phase $V_i^{\kappa_i}$. So, it is $\rho_i\,\mathbb{E}[V_i^{\kappa_i}] =: m$. At the *start* of the visit time it is

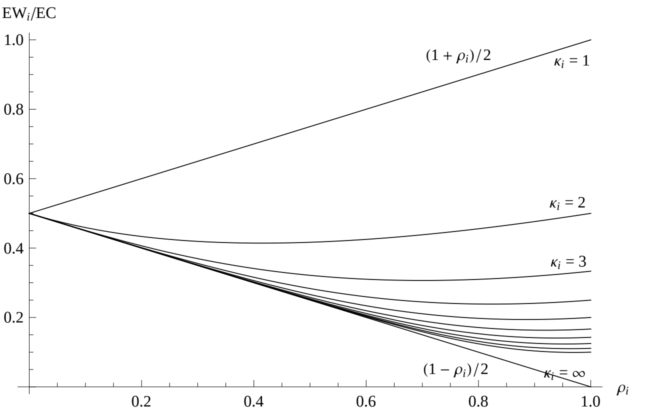


Figure 12.2: Fluid limits of the waiting times: $\mathbb{E}[W_i^{fluid}]/\mathbb{E}[C]$ *plotted versus* ρ_i *for* $\kappa_i = 1, 2, 3, \ldots, 10$ *and for* $\kappa_i = \infty$.

equal to the work already present at the beginning of I_i, which is m, plus the work built up during the intervisit time. Hence, it is $m + \rho_i\,\mathbb{E}[I_i] =: M$. Consequently, the average fluid level during a cycle, i.e. the mean workload $\mathbb{E}[H_i]$, is given by:

$$\begin{aligned}\mathbb{E}[H_i] = \frac{m+M}{2} &= m + \frac{\rho_i\,\mathbb{E}[I_i]}{2}\\ &= (1+\rho_i^{\kappa_i})\frac{\rho_i(1-\rho_i)}{2(1-\rho_i^{\kappa_i})}\mathbb{E}[C].\end{aligned}$$

Using Little's Law formulated for the workload, $\mathbb{E}[H_i] = \rho_i\,\mathbb{E}[W_i]$, the fluid limit of the mean waiting time of a type i customer directly follows:

$$\mathbb{E}[W_i^{fluid}] = \frac{m+M}{2\rho_i} = (1+\rho_i^{\kappa_i})\frac{1-\rho_i}{2(1-\rho_i^{\kappa_i})}\mathbb{E}[C]. \tag{12.3.5}$$

Figure 12.2 shows these fluid limits for different κ_i.

It is easily checked that for $\kappa_i = 1$, (12.3.5) reduces to $\mathbb{E}[W_i^{fluid}] = \frac{1+\rho_i}{2}\mathbb{E}[C]$, which is indeed the fluid limit for the gated discipline. For $\kappa_i = \infty$, (12.3.5) reduces to $\mathbb{E}[W_i^{fluid}] = \frac{1-\rho_i}{2}\mathbb{E}[C]$, which is indeed the fluid limit for the exhaustive discipline.

12.4 Balancing fairness and efficiency

We now want to choose κ such that on one hand we achieve fairness, while on the other hand the system is still efficient. For that, we want to determine the κ that minimizes $\gamma(\alpha)$ as given in (12.3.4). As we do not have closed-form expressions for the mean waiting times, optimization could be done by an exhaustive search over all κ_i. However, we use the fluid limits (12.3.5) as approximation for the mean waiting times in the optimization:

$$\min_{\kappa} \gamma^{fluid}(\kappa, \alpha) \tag{12.4.1}$$

where

$$\gamma^{fluid}(\kappa,\alpha) = (1-\alpha)\max_{i,j}\Big(\mathbb{E}[W_i^{fluid}] - \mathbb{E}[W_j^{fluid}]\Big) + \alpha\sum_{i=1}^{N}\mathbb{E}[M_i^{\kappa\text{-}gat}].$$

For deriving a heuristic rule for the optimal setting of κ, we take the following approach. First we determine the κ_i's such that all mean waiting times are equal (optimal fairness), then, using these κ_i's, we minimize the term $\sum_i \mathbb{E}[M_i]$ (maximal efficiency given optimal fairness). That is, we consider the following optimization problem:

$$\begin{aligned} &\min_{\kappa}\ \sum_{i=1}^{N}\mathbb{E}[M_i^{\kappa\text{-}gat}], \\ &\text{such that } \ \mathbb{E}[W_1^{fluid}] = \ldots = \mathbb{E}[W_N^{fluid}]. \end{aligned} \tag{12.4.2}$$

For the moment we allow the κ_i's to be fractional, later we round them to integers. Note that the problem in (12.4.2) does not depend on α. In an extensive numerical study in the next section we compare the performance of this heuristic setting to that of the optimal setting solving (12.4.1). We now solve (12.4.2), first for 2 queues, and then for N queues.

12.4.1 2 queues

For simplicity we start with the case of 2 queues. In this case we can explicitly solve $\mathbb{E}[W_1^{fluid}] = \mathbb{E}[W_2^{fluid}]$ for κ_2 in terms of κ_1, ρ_1 and ρ_2:

$$(1+\rho_1^{\kappa_1})\frac{1-\rho_1}{2(1-\rho_1^{\kappa_1})} = (1+\rho_2^{\kappa_2})\frac{1-\rho_2}{2(1-\rho_2^{\kappa_2})},$$

where we have divided by $\mathbb{E}[C] \neq 0$. Solving for κ_2, denoted by κ_2^{opt}, gives:

$$\rho_2^{\kappa_2^{opt}} = \frac{(1-\rho_1)\left(1+\rho_1^{\kappa_1}\right) - (1-\rho_2)\left(1-\rho_1^{\kappa_1}\right)}{(1-\rho_1)\left(1+\rho_1^{\kappa_1}\right) + (1-\rho_2)\left(1-\rho_1^{\kappa_1}\right)}. \tag{12.4.3}$$

So, this κ_2 achieves optimal fairness (recall that we allowed κ_2 to be fractional). Using this κ_2, we now optimize the efficiency, i.e. we minimize:

$$\begin{aligned} \sum_{i=1}^{2}\mathbb{E}[M_i] &= \rho_1^{\kappa_1+1}\frac{1-\rho_1}{1-\rho_1^{\kappa_1}} + \rho_2^{\kappa_2^{opt}+1}\frac{1-\rho_2}{1-\rho_2^{\kappa_2^{opt}}} \\ &= \frac{(\rho_1-\rho_2)\rho_2 + \rho_1^{\kappa_1}\left(2(1-\rho_1)\rho_1 + (2-\rho_1)\rho_2 - \rho_2^2\right)}{2(1-\rho_1^{\kappa_1})}. \end{aligned} \tag{12.4.4}$$

In (12.4.4) we have substituted (12.4.3) and simplified the expression.

The minimum of (12.4.4) (where $\kappa_1 > 0$, for $\rho_1 \neq \rho_2$) is found for $\kappa_1 \to \infty$. In this way, from (12.4.3), κ_2^{opt} becomes:

$$\kappa_2^{opt} = \log_{\rho_2}\ \frac{\rho_2-\rho_1}{2-\rho_1-\rho_2}. \tag{12.4.5}$$

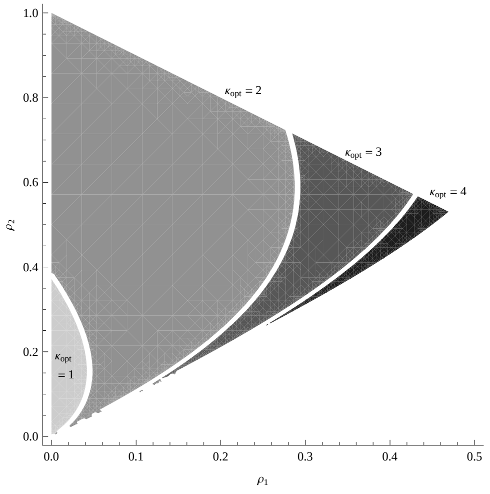


Figure 12.3: Optimal value of κ_2 (rounded to the nearest integer), given by $\kappa_2^{opt} = \left[\log_{\rho_2} \dfrac{\rho_2 - \rho_1}{2 - \rho_1 - \rho_2}\right]$, for $\rho_1 < \rho_2$ and $\rho_1 + \rho_2 < 1$.

This only makes sense for $\rho_1 < \rho_2$; if $\rho_1 > \rho_2$, we interchange the indices. In case $\rho_1 = \rho_2$ all $\kappa_1 = \kappa_2$ give equal mean waiting times. However, $\kappa_1 = \kappa_2 = \infty$ optimizes the efficiency. So, we come up with the following heuristic for the choice of κ_1 and κ_2:

$$\begin{cases} \text{if } \rho_1 < \rho_2\text{:} & \kappa_1 = \infty,\ \kappa_2 = \log_{\rho_2} \frac{\rho_2 - \rho_1}{2 - \rho_1 - \rho_2}, \\ \text{if } \rho_1 = \rho_2\text{:} & \kappa_1 = \kappa_2 = \infty, \\ \text{if } \rho_1 > \rho_2\text{:} & \kappa_1 = \log_{\rho_1} \frac{\rho_1 - \rho_2}{2 - \rho_1 - \rho_2},\ \kappa_2 = \infty. \end{cases}$$

In order to get integer κ_i's, we have three possibilities: rounding to the nearest integer, denoted by $[x]$; using the integer floor function, $\lfloor x \rfloor$; or using the integer ceiling function, $\lceil x \rceil$. We study all three options in the numerical study in Section 12.5. We denote a κ set according to the heuristic by $[\kappa]$, $\lfloor \kappa \rfloor$, respectively $\lceil \kappa \rceil$.

We plot (12.4.5) in Figure 12.3, for $\rho_1 < \rho_2$ (and $\rho_1 + \rho_2 < 1$ for stability) where we round κ_2. From the figure it becomes clear that $\kappa_2 = 2$ almost always is a proper choice.

12.4.2 *N* queues

For N queues we first determine which κ_i's give equal mean waiting times. We solve $\mathbb{E}[W_1^{fluid}] = \mathbb{E}[W_j^{fluid}]$ for $j = 2, \ldots, N$, which leads to an expression analogous to (12.4.3), with 2 replaced by j everywhere. We plug these into $\sum_{i=1}^{N} \mathbb{E}[M_i]$. The resulting expression depends only on κ_1, and on all ρ_i's. It only makes sense if ρ_1 is the smallest of all ρ_i, and it is minimized for $\kappa_1 \to \infty$. From this, the expressions for the optimal $\kappa_2, \ldots, \kappa_N$ directly follow: $\kappa_j^{opt} = \log_{\rho_j} \frac{\rho_j - \rho_1}{2 - \rho_1 - \ldots - \rho_N}$, for $j = 2, \ldots, N$.

Hence, we come up with the following heuristic for the choice of the κ_i's, $i = 1, \ldots, N$:

$$\begin{cases} \text{For all } i \text{ such that } i = \arg\min \rho_i, \text{ let } \kappa_i = \infty; \\ \text{For all } j = 1, 2, \ldots, N \text{ where } j \neq i, \text{ let } \kappa_j = \log_{\rho_j} \frac{\rho_j - \rho_i}{2-\rho}. \end{cases} \tag{12.4.6}$$

It is interesting to note that for any number of queues N, the queue(s) with the smallest traffic load ρ_i will always be served exhaustively (corresponding to $\kappa_i = \infty$).

Recall that we have three options to get integer κ_i's (round, floor, ceiling). An important notion here is that, by construction, this heuristic does *not* depend on α. The numerical results in the next section, however, show that it performs well for a wide range of α's. So, this heuristic is robust against the value of α.

12.5 Numerical analysis

In this section we first consider two examples, followed by an extensive numerical study into the performance of the heuristic setting of κ. We determine $\gamma(\alpha)$ as defined in (12.3.4), where, for brevity of notation, we define:

$$\Delta = \max_{i,j}\big(\mathbb{E}[W_i] - \mathbb{E}[W_j]\big),$$

$$\Sigma = \sum_{i=1}^{N} \mathbb{E}[M_i].$$

We compare the results of the κ-gated discipline with the elevator strategy in a globally gated regime, cf. [3, 167]. For this strategy all mean waiting times are equal: $\mathbb{E}[W_1^{elev.GG}] = \mathbb{E}[W_2^{elev.GG}] = \ldots = \mathbb{E}[W_N^{elev.GG}]$, and given by, cf. [3, (6), (10)]:

$$\mathbb{E}[W_1^{elev.GG}] = \frac{1}{1-\rho}\sum_{i=1}^{N}\rho_i \mathbb{E}[B_i^{res}] + \mathbb{E}[S^{res}] + \frac{1+\rho}{2(1-\rho)}\mathbb{E}[S].$$

The PCL easily follows: $\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] = \rho\, \mathbb{E}[W_1]$. Using (12.3.3), it follows that:

$$\sum_{i=1}^{N} \mathbb{E}[M_i^{elev.GG}] = \frac{\rho + \sum_{i=1}^{N} \rho_i^2}{2(1-\rho)}\, \mathbb{E}[S].$$

12.5.1 Examples

Example 12.5.1. Consider a polling model with $N = 2$ queues, S_i, $B_i \sim \exp(1)$, $i = 1, 2$, and $\lambda_1 = 0.6$, $\lambda_2 = 0.2$. Hence $\rho_1 = 0.6$ and $\rho_2 = 0.2$. We have $\rho_1 > \rho_2$ and $\log_{\rho_1} \frac{\rho_1 - \rho_2}{2-\rho_1-\rho_2} \approx 2.15$. Hence the heuristic settings are $[\kappa] = \lfloor \kappa \rfloor = (2, \infty)$ and $\lceil \kappa \rceil = (3, \infty)$.

For the κ-gated discipline, taking $\kappa_1, \kappa_2 \in \{1, 2, 3, \infty\}$, the performance $\gamma(\alpha)$ defined by (12.3.4) is listed in Table 12.1 for $\alpha = 0, \frac{1}{2}, \frac{2}{3}$, and $\frac{5}{6}$. It turns out that the heuristic settings for κ perform well in comparison to most other settings listed in Table 12.1. Although suboptimal for small α, their performance seems to be rather robust with respect to α. Despite $\kappa = (2, 2)$ performs better in this example for the four values of α chosen,

	κ_1	κ_2	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
	1	1	12.77	9.69	3.08	4.00	3.1	3.5	3.7	3.8
	2	1	8.42	11.49	3.07	1.75	3.1	2.4	2.2	1.0
	3	1	6.90	12.60	5.70	1.06	5.7	3.4	2.6	1.8
	∞	1	**5.04**	14.88	9.84	0.40	9.8	5.1	3.5	1.0
	1	2	13.00	7.21	5.83	3.67	5.8	4.8	4.4	4.0
	2	2	8.77	8.77	0.0038	1.42	0.0	**0.7**	**0.9**	1.2
	3	2	7.27	9.82	2.55	0.73	2.6	1.6	1.3	1.0
	∞	2	5.38	12.20	6.82	0.07	6.8	5.0	4.5	4.1
	1	3	13.10	6.76	6.34	3.61	6.3	5.0	4.5	4.1
	2	3	8.85	8.25	0.60	1.36	0.6	1.0	1.1	1.2
	3	3	7.36	9.29	1.92	0.67	1.9	1.3	1.1	0.9
	∞	3	5.47	11.66	6.19	0.01	6.2	3.1	2.1	1.0
	1	∞	13.12	**6.64**	6.48	3.60	6.5	5.0	4.6	4.1
$[\kappa], \lfloor\kappa\rfloor$:	$\underline{2}$	$\underline{\infty}$	$\underline{8.88}$	$\underline{8.11}$	$\underline{0.77}$	$\underline{1.35}$	$\underline{0.8}$	$\underline{1.1}$	$\underline{1.2}$	$\underline{1.3}$
$\lceil\kappa\rceil$:	$\underline{3}$	$\underline{\infty}$	$\underline{7.39}$	$\underline{9.13}$	$\underline{1.74}$	$\underline{0.66}$	$\underline{1.7}$	$\underline{1.2}$	$\underline{1.0}$	$\underline{\mathbf{0.8}}$
	∞	∞	5.50	11.50	6.00	**0.00**	6.0	3.0	2.0	1.0
	Elev.GG		15.00	15.00	**0.00**	6.00	**0.0**	3.0	4.0	5.0

Table 12.1: Results for Example 12.5.1 for the κ-gated strategy, where $\kappa_1, \kappa_2 \in \{1, 2, 3, \infty\}$. Smallest values per column are given in bold; the optimal settings from the heuristic are underlined. Recall that $\kappa_i = \infty$ is equivalent to the exhaustive service discipline; $\kappa_i = 1$ to the gated service discipline. Elevator strategy in a globally gated regime is added for comparison.

each of the heuristic settings will dominate in performance for α close to 1, since they are more efficient. In general, the heuristic settings outperform the $(2,2)$ setting (unless α is small), as the numerical study in Section 12.5.2 shows (see Table 12.5).

The difference in mean waiting times, Δ, is, in this example, minimal for $\kappa = (2,2)$. This is not surprising as for $N = 2$ and $\kappa = (2,2)$ the κ-gated discipline closely resembles the elevator strategy in a globally gated regime (cf. [3, 167]). In this discipline the visit order is $1, 2, \ldots, N-1, N, N, N-1, \ldots, 2, 1, 1, 2, \ldots$, and *all* gates are closed when turning around at 1 and at N. Hence, for $N = 2$ the queues are served as:

$$\ldots - Q_1 \overset{(*)}{-} Q_1 - S_1 - Q_2 \overset{(*)}{-} Q_2 - S_2 - Q_1 \overset{(*)}{-} Q_1 - S_1 - \ldots - \ldots$$

where $(*)$ denotes that the gates are closed at *both* queues. In the κ-gated strategy where $\kappa = (2,2)$, the queues are served as:

$$\ldots - Q_1^{(1)} - Q_1^{(2)} - S_1 - Q_2^{(1)} - Q_2^{(2)} - S_2 - Q_1^{(1)} - Q_1^{(2)} - S_1 - \ldots - \ldots$$

where the gate is closed when each service phase starts. As the elevator strategy in a globally gated regime leads to $\mathbb{E}[W_1] = \mathbb{E}[W_2]$, it should not be surprising that the $(2,2)$-gated strategy leads to *almost* equal mean waiting times.

Example 12.5.2. Now consider the following setting, again for $N = 2$ queues. Let $S_i \sim \exp(2)$, $B_i \sim \exp(1)$, $i = 1,2$ and $\lambda_1 = 0.35$, $\lambda_2 = 0.25$. Hence $\rho_1 > \rho_2$ and $\log_{\rho_1} \frac{\rho_1 - \rho_2}{2 - \rho_1 - \rho_2} \approx 2.51$, and thus the heuristic settings are $[\kappa] = \lceil\kappa\rceil = (3, \infty)$ and $\lfloor\kappa\rfloor = (2, \infty)$. The results are given in Table 12.2. The heuristic setting $(3, \infty)$ performs well compared to the other settings of κ in Table 12.2, and is even optimal for $\alpha = \frac{1}{2}, \frac{2}{3}$, and $\frac{5}{6}$. Note that Δ is again small for $\kappa = (2,2)$.

	κ_1	κ_2	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
	1	1	9.30	8.68	0.63	1.85	0.6	1.2	1.4	1.6
	2	1	6.36	9.16	2.80	0.94	2.8	1.9	1.6	1.3
	3	1	5.60	9.37	3.77	0.73	3.8	2.3	1.7	1.2
	∞	1	**5.19**	9.53	4.33	0.63	4.3	2.5	1.9	1.2
	1	2	9.57	6.30	3.28	1.35	3.3	2.3	2.0	1.7
	2	2	6.65	6.77	0.12	0.44	0.1	0.3	0.3	0.4
	3	2	5.87	6.98	1.11	0.23	1.1	0.7	0.5	0.4
	∞	2	5.46	7.16	1.71	0.13	1.7	0.9	0.7	0.4
	1	3	9.66	5.80	3.86	1.25	3.9	2.6	2.1	1.7
	2	3	6.74	6.25	0.49	0.35	0.5	0.4	0.4	0.4
	3	3	5.97	6.47	0.50	0.13	0.5	0.3	0.3	0.2
	∞	3	5.55	6.65	1.10	0.03	1.1	0.6	0.4	0.2
	1	∞	9.70	**5.63**	4.07	1.23	4.1	2.7	2.2	1.7
$\lfloor\kappa\rfloor$:	$\underline{2}$	$\underline{\infty}$	$\underline{6.79}$	$\underline{6.07}$	$\underline{0.72}$	$\underline{0.32}$	$\underline{0.7}$	$\underline{0.5}$	$\underline{0.5}$	$\underline{0.4}$
$[\kappa],\lceil\kappa\rceil$:	$\underline{3}$	$\underline{\infty}$	$\underline{6.02}$	$\underline{6.28}$	$\underline{0.26}$	$\underline{0.10}$	$\underline{0.3}$	$\underline{\mathbf{0.2}}$	$\underline{\mathbf{0.2}}$	$\underline{\mathbf{0.1}}$
	∞	∞	5.60	6.46	0.87	**0.00**	0.9	0.4	0.3	0.1
	Elev.GG		11.50	11.50	**0.00**	3.93	**0.0**	2.0	2.6	3.3

Table 12.2: Results for Example 12.5.2 for the κ-gated strategy. Smallest values per column are given in bold; the optimal settings from the heuristic are underlined.

12.5.2 Performance of fluid based heuristic

In a numerical experiment we study the performance of the heuristic settings for the κ_i's compared to the exhaustive, gated, and globally gated disciplines. For systems with only a few queues, we also compare their performance to that of the optimal κ-gated discipline. We use a test bed with 4,614 instances (see Table 12.3) with $N = 2, 3, 4$, and 5 queues. We calculate the mean waiting times and derive $\gamma(\alpha)$ for $\alpha = 0, \frac{1}{2}, \frac{2}{3}$, and $\frac{5}{6}$ for the following strategies:

- exhaustive;
- gated;
- κ-gated with κ as in the heuristic (cf. (12.4.6));
- κ-gated with optimal κ, found by enumeration of all possibilities (only for $N = 2, 3$);
- elevator strategy in a globally gated regime (cf [3, 167]).

The elevator strategy in a globally gated regime is added for comparison, as it is known to give identical mean waiting times. However, it is in general far less efficient. On the contrary, the exhaustive discipline is optimally efficient, however, it might be less fair. For the κ-gated discipline, we consider κ set according to the heuristic setting, as well as the optimal κ. The latter is found by enumerating over all combinations of $\kappa_i \in \{1, 2, 3, 4, 5, 6, \infty\}$ (for $N = 2$) or $\kappa_i \in \{1, 2, 3, \infty\}$ (for $N = 3$), for $i = 1, \dots, N$. For $N = 4$ and 5 we omit this, as enumeration over all combinations of κ_i is too time-consuming for these N.

The results over all cases of the test bed are given in Table 12.4, and are split out for $N = 2, 3, 4$, and 5 in Tables 12.5, 12.6, 12.7, and 12.8 respectively. The columns list

TEST BED	
$N = 2$ (2,925 settings)	
λ_1, λ_2	$\in \{0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95\}$,
B_1	$\sim \exp(1)$,
B_2, S_1, S_2	$\sim \exp(.)$ with mean $\in \{0.2, 0.5, 1., 2., 5.\}$,
$N = 3$ (243 settings)	
$\lambda_1, \lambda_2, \lambda_3$	$\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$,
B_1	$\sim \exp(1)$,
B_2, B_3	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,
S_1, S_2, S_3	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,
$N = 4$ (552 settings)	
$\lambda_1, \dots, \lambda_4$	$\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$,
B_1	$\sim \exp(1)$,
$B_2, \dots, B_4$	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,
$S_1, \dots, S_4$	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,
$N = 5$ (894 settings)	
$\lambda_1, \dots, \lambda_5$	$\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$,
B_1	$\sim \exp(1)$,
$B_2, \dots, B_5$	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,
$S_1, \dots, S_5$	$\sim \exp(.)$ with same mean $\in \{0.5, 1., 2.\}$,

Table 12.3: Test bed for numerical study: full factorial design of the given possibilities which are stable: $\sum_{i=1}^{N} \rho_i < 1$. In total 4,614 settings.

discipline \ averages	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
exhaustive	19.1	**0.0**	19.1	9.6	6.4	3.2
gated	6.4	9.3	6.4	7.9	8.3	8.8
elevator gg	**0.0**	14.7	**0.0**	7.4	9.8	12.3
κ-gat heur (round)	1.2	3.2	1.2	**2.2**	2.5	2.9
κ-gat heur (floor)	5.9	6.7	5.9	6.3	6.4	6.6
κ-gat heur (ceiling)	1.9	2.6	1.9	2.3	**2.4**	**2.5**

Table 12.4: Results over all 4,614 instances.

the average values over all cases considered. For example, the column $\mathbb{E}[W_1]$ shows the average of $\mathbb{E}[W_1]$ over all cases, and the column Δ the average of $\Delta = \max_{i,j}\big(\mathbb{E}[W_i] - \mathbb{E}[W_j]\big)$ over all cases. Note that the average of $\max_{i,j}\big(\mathbb{E}[W_i] - \mathbb{E}[W_j]\big)$ is not the same as the maximum of the differences between the average values of $\mathbb{E}[W_i]$ and $\mathbb{E}[W_j]$.

From the tables we can make the following observations. The elevator strategy in a globally gated regime, having equal mean waiting times (maximal fairness), is always optimal for $\alpha = 0$. This will be the case for small values of α near zero as well. The exhaustive strategy (in all queues), leading to $\Sigma = 0$ (maximal efficiency), will be optimal for values of α close to one. The κ-gated discipline, using the heuristic settings for κ, seems to perform well in the range of α's in between. For a specific α (and specific setting of the parameters), one can typically find a better performing κ, but this optimization by exhaustive search is very time-consuming. For $N = 2$ it outperforms $(2, 2)$ for all α except for α close to zero. When using the floor function in the heuristic, the results seem to be not that good. It is both less fair and less efficient on average than the rounding and ceiling. The performance of those two does not differ that much.

One might expect the performance of the different settings to depend heavily on the switch-over times incurred during a cycle, as during those intervals all work in the system

discipline \ averages	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
Q_1 exh - Q_2 exh	9.8	30.7	25.6	**0.0**	25.6	12.8	8.5	4.3
Q_1 exh - Q_2 gat	**7.5**	35.8	28.2	1.7	28.2	15.0	10.5	6.1
Q_1 gat - Q_2 exh	21.7	**10.5**	11.3	7.9	11.3	9.6	9.0	8.5
Q_1 gat - Q_2 gat	19.5	15.2	6.2	9.6	6.2	7.9	8.5	9.0
elevator gg	22.7	22.7	**0.0**	11.9	**0.0**	6.0	7.9	9.9
κ-gat heur (round)	13.2	12.9	0.7	4.4	0.7	2.6	3.2	3.8
κ-gat heur (floor)	15.3	12.9	3.9	5.8	3.9	4.9	5.2	5.5
κ-gat heur (ceiling)	12.3	13.5	1.9	2.9	1.9	2.4	2.6	2.7
$\kappa = (2,2)$	13.5	13.6	0.4	4.1	0.4	2.3	2.9	3.5
κ-gat opt $\alpha = 0$	13.5	13.5	0.3	4.1	0.3	2.2	2.8	3.5
κ-gat opt $\alpha = \frac{1}{2}$	13.1	13.2	0.5	3.7	0.5	**2.1**	2.6	3.2
κ-gat opt $\alpha = \frac{2}{3}$	12.0	13.5	2.0	2.7	2.0	2.4	**2.5**	2.6
κ-gat opt $\alpha = \frac{5}{6}$	10.4	15.3	6.7	1.1	6.7	3.9	3.0	**2.0**

Table 12.5: Results of numerical study for $N = 2$: average values over 2,925 cases (as described in Table 12.3). Minimum value per column in bold. Optimization of κ by exhaustive search over $\kappa_i \in \{1,2,3,4,5,6,\infty\}$, $i = 1,2$.

discipline \ averages	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	$\mathbb{E}[W_3]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
exhaustive	11.2	12.2	12.3	6.2	**0.0**	6.2	3.1	2.1	**1.0**
gated	16.0	15.3	15.4	4.2	5.6	4.2	4.9	5.1	5.4
elevator gg	21.4	21.4	21.4	**0.0**	10.2	**0.0**	5.1	6.8	8.5
κ-gat heur (round)	12.2	12.1	**12.2**	0.7	1.6	0.7	1.2	1.3	1.5
κ-gat heur (floor)	14.7	14.0	14.0	6.2	4.9	6.2	5.6	5.3	5.1
κ-gat heur (ceiling)	12.1	12.2	12.2	0.8	1.6	0.8	1.2	1.3	1.5
κ-gat opt $\alpha = 0$	12.2	12.2	12.3	0.7	1.7	0.7	1.2	1.4	1.5
κ-gat opt $\alpha = 1$	12.1	12.2	12.2	0.7	1.6	0.7	**1.2**	1.3	1.5
κ-gat opt $\alpha = 2$	11.9	12.0	12.3	1.1	1.4	1.1	1.3	**1.3**	1.4
κ-gat opt $\alpha = 5$	**10.7**	**11.7**	13.5	4.6	0.5	4.6	2.6	1.9	1.2

Table 12.6: Results of numerical study for $N = 3$: average values over 243 cases (as described in Table 12.3). Optimization of κ by exhaustive search over $\kappa_i \in \{1,2,3,\infty\}$, $i = 1,2,3$.

discipline \ averages	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	$\mathbb{E}[W_3]$	$\mathbb{E}[W_4]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
exhaustive	**20.3**	**22.4**	**22.4**	**22.6**	9.7	**0.0**	9.7	4.9	3.2	**1.6**
gated	30.6	29.0	29.0	29.1	8.0	11.1	8.0	9.6	10.1	10.6
elevator gg	43.5	43.5	43.5	43.5	**0.0**	23.0	**0.0**	11.5	15.3	19.2
κ-gat heur (round)	22.9	23.1	23.2	23.2	2.3	2.9	2.3	2.6	2.7	2.8
κ-gat heur (floor)	29.5	27.6	27.6	27.6	11.6	10.5	11.6	11.1	10.9	10.7
κ-gat heur (ceiling)	22.9	23.1	23.2	23.2	2.3	2.9	2.3	**2.6**	**2.7**	2.8

Table 12.7: Results of numerical study for $N = 4$: average values over 552 cases (as described in Table 12.3). Note: Ceiling differs in only 3 instances from Round.

is waiting. For that reason, we separate the results according to the value of $\mathbb{E}[S]$ (the mean total switch-over time during a cycle), see Table 12.9. We focused on $N = 2$, as this most clearly illustrates the results. From the table, we see that the performance of e.g. the elevator strategy in a globally gated regime is best for small values of $\mathbb{E}[S]$, as is to be expected, but it is outperformed by the κ-gated discipline already for small α, by all indicated choices for the setting of the κ (except the heuristic setting using the floor function). Note that it is also outperformed by $\kappa = (2, 2)$, although these settings closely resemble each other.

Summarizing, the κ-gated discipline with κ set according to the heuristic, either rounding or ceiling, is robust against the setting of α and it performs well over a wide range of values for α and $\mathbb{E}[S]$.

12.6 Conclusion

We introduced the κ-gated service discipline for a polling model. It is a hybrid of the classical gated and exhausted disciplines, and consists of using κ_i gated service phases at Q_i before the server switches to the next queue. The aim of this discipline is to provide fairness (almost equal mean waiting times at the queues), while not giving up efficiency (weighted sum of mean waiting times). For the trade-off between these two we introduced the factor α. The κ_i's can then be optimized.

We showed how the mean visit times, the pseudo conservation law, the distribution of waiting times and the mean waiting times can be derived. We also derived the fluid limits. Further, using the fluid limits, we provided a heuristic to set the κ (not depending on α). In an extensive numerical study we showed that the heuristic performs well. Typically when α is given, one can find (e.g. by an exhaustive search) a better setting, but the heuristic setting is robust against the value of α, that is, for all α it performs well. So, the factor α typically does not play a significant role in the choice of κ.

We have chosen here to set κ so as to optimize the fairness and efficiency. However, the κ-gated discipline can be used for other performance characteristics on the mean waiting times as well. Instead of the efficiency, one could for example consider the sum $\sum_{i=1}^{N} c_i \mathbb{E}[W_i]$, where each queue $i = 1, \ldots, N$ is assigned a cost factor c_i. This could e.g. reflect a difference in the importance of the customers in each queue.

An interesting option for further research is the handling of the fractional κ_i's. Instead of rounding, one might assign a probability, say p_i, with which $\lfloor \kappa_i \rfloor$ phases are used, and $\lceil \kappa_i \rceil$ otherwise. This, however, might lead to a more complicated exact analysis. Another question is in which order the queues should be placed, as to minimize the variance in waiting times or in the $\gamma(\alpha)$.

discipline \ averages	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	$\mathbb{E}[W_3]$	$\mathbb{E}[W_4]$	$\mathbb{E}[W_5]$	Δ	Σ	$\gamma(0)$	$\gamma(\frac{1}{2})$	$\gamma(\frac{2}{3})$	$\gamma(\frac{5}{6})$
exhaustive	**19.9**	**20.6**	**20.6**	**20.6**	**20.7**	7.4	**0.0**	7.4	3.7	2.5	**1.2**
gated	26.6	26.0	26.0	26.1	26.1	6.5	8.2	6.5	7.4	7.6	7.9
elevator gg	39.9	39.9	39.9	39.9	39.9	**0.0**	20.2	**0.0**	10.1	13.5	16.8
κ-gat heur (round)	21.1	21.3	21.3	21.3	21.4	2.4	1.9	2.4	**2.2**	**2.1**	2.0
κ-gat heur (floor)	26.1	25.0	25.0	25.0	25.0	9.2	7.9	9.2	8.6	8.3	8.1
κ-gat heur (ceiling)	21.1	21.3	21.3	21.3	21.4	2.4	1.9	2.4	**2.2**	**2.1**	2.0

Table 12.8: Results of numerical study for $N = 5$: average values over 894 cases (as described in Table 12.3). Note: Ceiling identical to Round in all tested instances.

value of $\mathbb{E}[S]$ (inst.):	$[0.4, 1]$ (468)			$(1, 2]$ (585)			$(2, 3]$ (702)			$(3, 5.5]$ (585)			$(5.5, 10]$ (585)		
discipline \ averages	Δ	Σ	$\gamma(\frac{1}{2})$	Δ	Σ	$\gamma(\frac{1}{2})$	Δ	Σ	$\gamma(\frac{1}{2})$	Δ	Σ	$\gamma(\frac{1}{2})$	Δ	Σ	$\gamma(\frac{1}{2})$
exhaustive	20.8	0.0	10.4	22.1	0.0	11.1	24.0	0.0	12.0	28.7	0.0	14.4	31.6	0.0	15.8
gated	2.4	1.9	2.2	3.4	4.1	3.8	4.9	7.1	6.0	8.4	14.0	11.2	11.2	19.8	15.5
elevator gg	0.0	2.4	1.2	0.0	5.1	2.6	0.0	8.8	4.4	0.0	17.4	8.7	0.0	24.6	12.3
κ-gat heur (round)	0.3	0.8	0.6	0.4	1.6	2.0	0.6	2.7	1.7	1.0	5.4	3.2	1.2	7.7	4.5
κ-gat heur (floor)	1.6	1.2	1.4	2.2	2.5	2.4	3.1	4.3	3.7	5.3	8.4	6.9	7.0	12.0	9.5
κ-gat heur (ceiling)	0.9	0.6	1.8	1.2	1.2	1.2	1.6	2.2	1.9	2.5	4.3	3.4	3.1	6.0	4.6
κ-gat opt $\alpha = 0$	0.1	0.8	0.5	0.2	1.7	1.0	0.3	3.0	1.7	0.5	5.9	3.2	0.6	8.4	4.5
κ-gat opt $\alpha = \frac{1}{2}$	0.1	0.8	0.5	0.2	1.6	0.9	0.3	2.7	1.5	0.7	5.3	3.0	0.9	7.5	4.2
κ-gat opt $\alpha = \frac{2}{3}$	0.1	0.8	0.5	0.3	1.5	0.9	0.6	2.6	1.6	3.4	3.7	3.6	5.6	4.5	5.1
κ-gat opt $\alpha = \frac{5}{6}$	0.8	0.6	0.7	2.7	0.8	1.8	5.3	1.0	3.2	10.4	1.3	5.9	13.3	1.7	7.5
$\kappa = (2, 2)$	0.1	0.8	0.5	0.2	1.7	1.0	0.3	3.0	1.7	0.7	6.0	3.4	0.8	8.5	4.7

Table 12.9: Results for $N = 2$ split out according to $\mathbb{E}[S]$ (the mean total switch-over time during a cycle), where $\mathbb{E}[S] \in [0.4, 10]$ for the test bed of Table 12.3.

13

Gated/Exhaustive

We consider a polling system where the server cyclically serves the queues according to the following discipline: the server does one round of visits to the queues applying the gated service discipline at each of the queues, followed by one round of visits applying the exhaustive service discipline at each of the queues, and this alternating pattern repeats itself. We call this the *Gated/Exhaustive service discipline*. For this we derive (i) a Pseudo Conservation Law for the weighted sum of the mean waiting times, (ii) the mean steady-state waiting times using Mean Value Analysis, and (iii) queue length distributions making use of Multi-type Branching Processes. For (ii) and (iii) we apply the concept of smart customers (see Chapter 10).

13.1 Introduction

The classical polling system is a queueing system with multiple queues and one single server. The server cyclically visits all queues, where it serves the customers. Typically a so-called switch-over time is incurred when the server switches from one queue to another. There are many possible choices for deciding when the server should switch to the next queue. Those most often studied are the exhaustive service discipline (when the server arrives at a queue, it serves its customers until the queue has become empty) and the gated service discipline (when the server arrives at a queue, a gate closes and only the customers who are before the gate, i.e., who are already present, will be served in this server visit).

The present chapter considers the following variant of this classical model. First the server cyclically serves all queues according to the gated service discipline, and then the server cyclically serves all queues according to the exhaustive service discipline, and this alternating pattern repeats itself. We call this the *Gated/Exhaustive discipline*.

We present a detailed analysis of this model. We first aim for mean values, deriving both a *Pseudo Conservation Law* for the mean waiting times and exploiting the *Mean Value Analysis* (MVA) technique, that was recently [210] developed for polling systems, to obtain all mean waiting times. For the latter, we use the concept of *smart customers*,cf. Chapter 10, and basically doubling the number of queues in the system. Subsequently, we relate the joint queue length process to *Multi-type Branching Processes*, also using smart customers. In this way, we obtain the joint queue length distribution at server polling epochs.

Our work was partly motivated by the question whether an alternation of gated and exhaustive cycles might lead to *fairness* to the queues, in the sense of having almost identical mean waiting times. Fairness is a topic that has frequently played a role in the choice of service discipline in polling systems. In some recent studies [152, 185] of dynamic bandwidth allocation of Ethernet Passive Optical Networks (EPON), polling models have been considered with two-stage gated service: a gate closes behind the customers in a stage-1 buffer at the moment the server arrives, the customers in the stage-2 buffer are being served, and then those present in stage-1 move to the stage-2 buffer. This was seen to give rise to relatively small differences between mean waiting times at the various queues, but at the expense of longer delays. We conjectured that an alternation of gated and exhaustive cycles might also lead to small differences between mean waiting times (but with smaller delays than for two-stage gated), as various mean waiting time approximations (see e.g. [72]) have been based on the following facts: (i) for gated service, the mean waiting time at the ith queue is $(1+\rho_i)$ times the mean residual length of one server cycle for the ith queue, with ρ_i the traffic load at the ith queue; (ii) for exhaustive service, the mean waiting time is roughly $(1-\rho_i)$ times the mean residual length of one server cycle. Averaging with equal weights would yield roughly equal mean waiting times at all queues; our numerical results, however, will show that there can still be substantial differences between mean waiting times.

The structure of this chapter is as follows. In Section 13.2 we introduce the model, explaining the Gated/Exhaustive service discipline. In Section 13.3 we analyze the Gated/Exhaustive discipline. We derive the mean visit times of the server at each of the queues, and derive the Pseudo Conservation Law for the mean waiting times. Also, we apply the concept of smart customers to the Gated/Exhaustive discipline, and hence derive the (mean) waiting times, using Mean Value Analysis and Multi-type Branching Processes. Then, in Section 13.4, we numerically compare the mean waiting times of the Gated/Exhaustive discipline to (other mixes of) the exhaustive and gated discipline, and we end with a discussion of possible further work in Section 13.5. This chapter is based on [41].

13.2 Model and notation

We consider a polling model as described in Section 9.2, using a new service discipline: the *Gated/Exhaustive service discipline* (G/E discipline). This discipline works as follows. The server visits the queues in fixed cyclic order. A cycle consists of the visit of the server to each of the queues twice: once serving them according to the gated service discipline, and once to the exhaustive one. The first visit to Q_i is gated, denoted by Q_{i_G}, the second exhaustive, denoted by Q_{i_E}. Starting with the switch-over to Q_1, a cycle is typically given by:

$$S_1 - Q_{1_G} - S_2 - Q_{2_G} - \ldots - S_N - Q_{N_G} - S_1 - Q_{1_E} - S_2 - Q_{2_E} - \ldots - S_N - Q_{N_E}.$$

The cycle time, denoted by C, consists of the visit times to each of the queues twice, and all switch-over times occurred. A well known result [176] for the mean cycle time in a system where the queues are visited once in a cycle is $\mathbb{E}[C_{1visit}] = \mathbb{E}[S]/(1-\rho)$. As a cycle now contains two visits to each of the queues, we have

$$\mathbb{E}[C] = \frac{2\mathbb{E}[S]}{1-\rho}. \tag{13.2.1}$$

13.3 Analysis of G/E discipline

In this section we present the analysis of the G/E Discipline. First, we derive the mean visit times at each of the queues. Then, we give a pseudo conservation law for the weighted sum of the mean waiting times. Next, we show how the G/E Discipline fits into the concept of smart customers. Using that, we derive the mean waiting times using mean value analysis, and derive the waiting time distributions using multi-type branching processes.

13.3.1 Mean visit times

For the G/E discipline, we derive the expected duration of the visits at each of the queues. Denote by $\mathbb{E}[V_{i_G}]$ the expected duration of a visit period to Q_i when it is served gated, and by $\mathbb{E}[V_{i_E}]$ when it is served exhaustively, $i = 1, \ldots, N$. Denote by $\mathbb{E}[V_i]$ the expected duration of the visit periods to Q_i per cycle, so

$$\mathbb{E}[V_i] = \mathbb{E}[V_{i_G}] + \mathbb{E}[V_{i_E}].$$

As the server is working a fraction ρ_i of the time on Q_i, it follows from (13.2.1) that, for $i = 1, \ldots, N$:

$$\mathbb{E}[V_i] = \frac{2\,\mathbb{E}[S]\,\rho_i}{1-\rho}. \tag{13.3.1}$$

In order to determine the individual mean visit times $\mathbb{E}[V_{i_G}]$ and $\mathbb{E}[V_{i_E}]$ we set up a system of linear equations. For each of the $2N$ visits to a queue during one cycle, we have a single linear equation. This equation expresses the expected duration of that visit in terms of the other mean visit times.

For $\mathbb{E}[V_{i_G}]$ we make use of the fact that at the moment an exhaustive service to Q_i ends, there are no type i customers present in the system any more. After this, type i customers arrive at rate λ_i during the switch-over and visit times at other queues, until the start of the next gated service at Q_i. At that moment the type i customers present in the system are placed before a gate and these are the only ones to be served in this visit period to the queue. Now the mean duration of this visit time is the mean number of customers present at the start of the service, times the mean service time per customer. The mean number of customers present at the start of the service is equal to the arrival rate λ_i times the expected amount of time that has passed since the previous exhaustive visit to the queue. This gives:

$$\begin{aligned}
\mathbb{E}[V_{i_G}] &= \lambda_i \mathbb{E}[B_i]\Big(\mathbb{E}[S_{i+1}] + \mathbb{E}[V_{i+1_E}] + \mathbb{E}[S_{i+2}] + \ldots \\
&\qquad + \mathbb{E}[V_{N_E}] + \mathbb{E}[S_1] + \mathbb{E}[V_{1_G}] + \ldots + \mathbb{E}[S_i]\Big) \\
&= \rho_i \left(\mathbb{E}[S] + \sum_{k=i+1}^{N} \mathbb{E}[V_{k_E}] + \sum_{k=1}^{i-1} \mathbb{E}[V_{k_G}] \right),
\end{aligned} \tag{13.3.2}$$

for $i = 1, \ldots, N$, where an empty sum equals zero.

A similar expression can be found for $\mathbb{E}[V_{i_E}]$. Note that at the beginning of a gated service to Q_i there are no type i customers behind the gate, as all are placed before. Newly arriving type i customers are not served during this visit period any more, but have to wait until the server returns to the queue. So the mean number of customers present at the beginning of the exhaustive service to Q_i is equal to the arrival rate λ_i times the expected amount of time that has passed since the *beginning* of the previous gated service. But as the service to the queue is now exhaustive, the newly arriving customers during this visit time are still to be served during this visit. This can be interpreted as that every customer present at the start of the visit time induces a busy period. The expected duration of a busy period of one type i customer is $\mathbb{E}[B_i]/(1-\rho_i)$. This gives

$$\begin{aligned}\mathbb{E}[V_{i_E}] &= \lambda_i \frac{\mathbb{E}[B_i]}{1-\rho_i}\Big(\mathbb{E}[V_{i_G}] + \mathbb{E}[S_{i+1}] + \mathbb{E}[V_{i+1_G}] + \ldots \\ &\qquad + \mathbb{E}[V_{N_G}] + \mathbb{E}[S_1] + \mathbb{E}[V_{1_E}] + \ldots + \mathbb{E}[S_i]\Big) \\ &= \frac{\rho_i}{1-\rho_i}\left(\mathbb{E}[S] + \sum_{k=i}^{N} \mathbb{E}[V_{k_G}] + \sum_{k=1}^{i-1} \mathbb{E}[V_{k_E}]\right), \end{aligned} \tag{13.3.3}$$

for $i = 1, \ldots, N$.

Now (13.3.2) and (13.3.3) give a system of $2N$ linear equations in the $2N$ unknowns $\mathbb{E}[V_{i_G}]$ and $\mathbb{E}[V_{i_E}]$, $i = 1, \ldots, N$. Solving this gives explicit expressions for $\mathbb{E}[V_{i_G}]$ and $\mathbb{E}[V_{i_E}]$, $i = 1, \ldots, N$, in terms of $\mathbb{E}[S]$ and ρ_i.

In equilibrium, the rate at which type i customers enter the system is equal to the rate at which they leave the system. So for $i = 1, \ldots, N$:

$$\lambda_i \mathbb{E}[C] = \frac{\mathbb{E}[V_{i_E}] + \mathbb{E}[V_{i_G}]}{\mathbb{E}[B_i]}.$$

The left-hand side gives the mean number of type i customers that enters the system during a cycle; the right-hand side gives the mean number of type i customers that are served during a cycle. This observation is another way to derive (13.3.1).

13.3.2 Pseudo Conservation Law

Boxma and Groenendijk [34] derive a Pseudo Conservation Law (PCL) for the case of cyclic order polling systems (see Section 9.3.1):

$$\sum_{i=1}^{N} \rho_i \mathbb{E}[W_i] = \frac{\rho}{1-\rho}\sum_{i=1}^{N} \rho_i \mathbb{E}[B_i^{res}] + \rho \mathbb{E}[S^{res}] + \frac{\mathbb{E}[S]}{2(1-\rho)}\left(\rho^2 - \sum_{i=1}^{N} \rho_i^2\right) + \sum_{i=1}^{N} \mathbb{E}[M_i], \tag{13.3.4}$$

where $\mathbb{E}[M_i]$ is the mean amount of work in Q_i at a departure epoch of the server from Q_i. For the exhaustive discipline, $\mathbb{E}[M_i^E] = 0$, and for the gated discipline $\mathbb{E}[M_i^G] = \rho_i \mathbb{E}[V_i] = \rho_i^2 \mathbb{E}[S]/(1-\rho)$.

The PCL is based on a workload decomposition, where the only aspect that depends on the service discipline is the amount of work left by the server upon a service completion. Using (9.3.4), deriving the PCL for a certain discipline reduces to determining $\mathbb{E}[Y]$:

the expected amount of work in the system at an arbitrary epoch in a switch-over interval. It is a weighted sum of the $\mathbb{E}[Y_i]$, which is the mean amount of work at an arbitrary epoch in a switch-over interval when switching to Q_i, cf. (9.3.5). In [34], $\mathbb{E}[Y]$ is derived for the cases of purely exhaustive and purely gated services, and also for mixtures of these. We now derive $\mathbb{E}[Y]$ in case of the G/E discipline, hence determining the PCL for the G/E discipline.

Denote by $\mathbb{E}[Y_{i_G}]$ the mean amount of work at an arbitrary epoch in a switch-over interval to Q_i served gated, i.e. to Q_{i_G}, and by $\mathbb{E}[Y_{i_E}]$ to Q_i served exhaustively, i.e. to Q_{i_E}. As both the switch-over intervals have the same distribution, when looking at the system at an arbitrary epoch in a switch-over interval to Q_i, it is with equal probability a switch-over interval to Q_i served gated or to Q_i served exhaustively. Therefore, for $i = 1, \ldots, N$:

$$\mathbb{E}[Y_i] = \frac{1}{2}\mathbb{E}[Y_{i_G}] + \frac{1}{2}\mathbb{E}[Y_{i_E}],$$

and so, using (9.3.5):

$$\mathbb{E}[Y] = \frac{1}{2}\sum_{i=1}^{N} \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]}\mathbb{E}[Y_{i_G}] + \frac{1}{2}\sum_{i=1}^{N} \frac{\mathbb{E}[S_i]}{\mathbb{E}[S]}\mathbb{E}[Y_{i_E}]. \tag{13.3.5}$$

We look at the system at an arbitrary epoch in the switch-over interval S_i. Firstly, we consider the amount of work that arrived to the system during the time already passed in this switch-over time. As at Q_i work is arriving at rate $\rho_i = \lambda_i \mathbb{E}[B_i]$, the rate at which work is arriving to the system is $\rho = \sum_{i=1}^{N} \rho_i$. Now use that the expected amount of time already passed during the switch-over interval is equal to the expected residual switch-over time, which is $\mathbb{E}[S_i^{res}]$. This gives that the amount of work arrived to the system during the time already passed in this switch-over interval is equal to $\rho\mathbb{E}[S_i^{res}]$.

Secondly, at each of the Q_i work arrived at rate ρ_i during the period until the start of the switch-over time which is considered. If the last visit to Q_i was exhaustive, then this mean amount of work is equal to ρ_i times the mean duration of all intervals after the end of the visit to Q_i until the start of the switch-over interval considered. If the last visit to the queue was gated, then we also have to include the mean duration of this gated visiting time, as the type i customers arriving in this interval are still in the system. Deriving the expressions for $\mathbb{E}[Y_{i_G}]$ and $\mathbb{E}[Y_{i_E}]$ is now straightforward, but some tedious bookkeeping is needed to consider which switch-over intervals and visit times are concerned. We find, for $i = 1, \ldots, N$:

$$\begin{aligned}
\mathbb{E}[Y_{i_G}] &= \rho\, \mathbb{E}[S_i^{res}] + \sum_{k=1}^{i-1} \rho_k \left(\mathbb{E}[V_{k_G}] + \sum_{j=k+1}^{i-1} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_G}]) \right) \\
&\qquad + \sum_{k=i}^{N} \rho_k \left(\sum_{j=1}^{i-1} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_G}]) + \sum_{j=k+1}^{N} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_E}]) \right), \\
\mathbb{E}[Y_{i_E}] &= \rho\, \mathbb{E}[S_i^{res}] + \sum_{k=1}^{i-2} \rho_k \left(\sum_{j=k+1}^{i-1} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_E}]) \right) \\
&\qquad + \sum_{k=i}^{N} \rho_k \left(\mathbb{E}[V_{k_G}] + \sum_{j=1}^{i-1} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_E}]) + \sum_{j=k+1}^{N} (\mathbb{E}[S_j] + \mathbb{E}[V_{j_G}]) \right),
\end{aligned}$$

where an empty sum equals zero. Substituting these expressions into (13.3.5) gives the expression for $\mathbb{E}[Y]$ in terms of ρ_i and $\mathbb{E}[S_i]$. Using (9.3.4) we then find the PCL for a polling system with N queues in the G/E policy.

For the case of $N = 1$ queue, served according to the G/E discipline, this gives

$$\mathbb{E}[W_1] = \frac{\rho^2 \mathbb{E}[B_1{}^{res}]}{1-\rho} + \rho \mathbb{E}[S_1{}^{res}] + \frac{\mathbb{E}[S_1]}{2(1-\rho)} \left(\rho - \rho^2\right).$$

Note that for gated services the last term is $\frac{\mathbb{E}[S_1]}{2(1-\rho)} \left(2\rho^2\right)$, and for exhaustive services it vanishes.

13.3.3 Smart customers

For modeling the G/E policy, we make use of *smart customers* (cf. Chapter 10). It provides the option to let the arrival rate at a queue depends on the position of the server. We exploit this concept for routing customers in a model with $2N$ queues. Denote the arrival rate of customers at Q_i when the server is working at Q_j by λ_{ij}, and the rate when the server is switching from Q_{j-1} to Q_j, by μ_{ij}.

In order to distinguish between the visits to a given queue in the gated service part of the cycle or in the exhaustive one, we number the queues as if there were $2N$ queues:

$$Q_1, Q_2, \ldots, Q_N, Q_{N+1}, \ldots, Q_{2N}.$$

For $i = 1, 2, \ldots, N$ we have that Q_i represents a gated visit to Q_i, and for $i = N+1, N+2, \ldots, 2N$ we have that Q_{N+i} represents an exhaustive visit to Q_i. A cycle of the server is given by:

$$S_1 - Q_1 - S_2 - Q_2 - \ldots - S_N - Q_N - S_1 - Q_{N+1} - S_2 - Q_{N+2} - \ldots - S_N - Q_{2N}.$$

The important observation here is that Q_i and Q_{N+i} are actually the same queue. When a type i customer arrives it should be directed to the appropriate queue. This can be achieved by a proper choice for the λ_{ij} and μ_{ij}.

First we look at the gated part of the cycle. When the server has not been working on Q_i yet, we direct arriving type i customers to queue Q_i. If the server has already served Q_i, then arriving type i customers are directed to the queue served exhaustively, i.e. to Q_{N+i}. If the server is working at Q_i, we have that arriving type i customers are not served in this service interval anymore, as the service discipline is gated. They will be served when the server returns to this queue, and so the customers are also directed to Q_{N+i}.

For the exhaustive part of the cycle we have almost the same reasoning. If the server has not yet been working on Q_{N+i}, arriving type i customers are directed to this queue. If the server has already served Q_{N+i}, they are sent to Q_i. But when the server is working at Q_{N+i}, newly arriving customers are in this case served in this service interval, as the service discipline is exhaustive. So they are sent to Q_{N+i}.

We can summarize the above as follows. Let the set $J_i = \{i+1, \ldots, N+i\}$, then

$$\lambda_{ij} = \begin{cases} \lambda_i & \text{for } i = 1, \ldots, N, \ j \notin J_i \cup \{i\}, \\ \lambda_i & \text{for } i = N+1, \ldots, 2N, \ j \in J_i \cup \{i\}, \\ 0 & \text{otherwise}, \end{cases} \tag{13.3.6}$$

and

$$\mu_{ij} = \begin{cases} \lambda_i & \text{for } i = 1, \dots, N, \ j \notin J_i, \\ \lambda_i & \text{for } i = N+1, \dots, 2N, \ j \in J_i, \\ 0 & \text{otherwise.} \end{cases} \tag{13.3.7}$$

13.3.4 Mean Value Analysis

In this section we derive the mean steady-state waiting times using *Mean Value Analysis* (MVA) with smart customers. The main idea of MVA is outlined in Section 9.3.2, and adapted in Chapter 10 for the concept of smart customers. In this section, we show how the G/E discipline fits into the framework of smart customers.

As in Section 13.3.3, we consider a model with $2N$ queues, where Q_i and Q_{i+N} are basically the same queue. A cycle consists of the visits to all of these $2N$ queues, and switch-over times incurred. We number the switch-over times and visit times to these queues from 1 to $4N$, starting with the switch-over to Q_1 served gated. This gives

$$\begin{array}{ccccccc} S_1 & Q_{1_G} & S_2 & Q_{2_G} & \dots & S_N & Q_{N_G} \\ 1 & 2 & 3 & 4 & \dots & 2N-1 & 2N \end{array}$$

$$\begin{array}{ccccccc} S_1 & Q_{1_E} & S_2 & Q_{2_E} & \dots & S_N & Q_{N_E} \\ 2N+1 & 2N+2 & 2N+3 & 2N+4 & \dots & 4N-1 & 4N \end{array}$$

Now we define period j, $j = 1, \dots, 4N$, as either the switch-over time or visit time numbered correspondingly. By q_j we denote the fraction of time the system is in period j, $j = 1, \dots, 4N$. Let interval (i, j) consist of the periods $i, i+1, \dots, i+j-1$. For the mean cycle length we have $\mathbb{E}[C] = 2\mathbb{E}[S]/(1-\rho)$, cf. (13.2.1), and the mean visit times to the queues $\mathbb{E}[V_{i_G}]$ and $\mathbb{E}[V_{i_E}]$ are derived in Section 13.3.1. Recall that the switch-over times to Q_i served gated and served exhaustively are probabilistically identical.

System of equations. We derive a system of equations in order to determine $\mathbb{E}[W_i]$. Let $\mathbb{E}[W_{i_G}]$ denote the mean waiting time of a type i customer receiving gated service, and $\mathbb{E}[W_{i_E}]$ denote the mean waiting time for one receiving exhaustive service. For these we have, for $i = 1, \dots, N$,

$$\mathbb{E}[W_i] = q_{G,i}\,\mathbb{E}[W_{i_G}] + q_{E,i}\,\mathbb{E}[W_{i_E}], \tag{13.3.8}$$

where $q_{G,i}$ denotes the fraction of type i customers that will receive gated service, and $q_{E,i}$ the fraction that will receive exhaustive service. Clearly $q_{E,i} = 1 - q_{G,i}$.

Let $\mathbb{E}[L_{i_G}]$ and $\mathbb{E}[L_{i_E}]$ be the mean number of waiting type i customers in the system that will receive gated, respectively exhaustive service. Then, for $i = 1, \dots, N$,

$$\mathbb{E}[L_i] = \mathbb{E}[L_{i_G}] + \mathbb{E}[L_{i_E}].$$

There are type i customers in the system waiting for exhaustive service during the periods $2i, \dots, 2N+2i$, and type i customers waiting for gated service during the periods $2N+2i+1, \dots, 4N, 1, \dots, 2i$. These two intervals are almost complementary; only during period $2i$, which is the visit to Q_{i_G}, both types of customers can be simultaneously present in the system: the ones that will receive (gated) service during this period and the ones that have arrived in this period, but who have to wait until the next (exhaustive) service to the queue. Hence we obtain, by PASTA, the following expressions, $i = 1, \dots, N$,

$$q_{G,i} = \sum_{j=2N+2i+1}^{2i-1} q_j, \quad q_{E,i} = \sum_{j=2i}^{2N+2i} q_j, \tag{13.3.9}$$

where the summation for $q_{G,i}$ should be understood to be cyclical, i.e., over all $j \in \{2N + 2i + 1, \dots, 4N, 1, \dots, 2i - 1\}$.

Denote by λ_{i_G} and λ_{i_E} the arrival rates of type i customers that will be served gated, respectively exhaustively. So, for $i = 1, \dots, N$,

$$\lambda_{i_G} = \lambda_i \, q_{G,i}, \quad \lambda_{i_E} = \lambda_i \, q_{E,i}.$$

Little's Law gives the following relations, for $i = 1, \dots, N$,

$$\mathbb{E}[L_{i_G}] = \lambda_{i_G} \mathbb{E}[W_{i_G}], \quad \mathbb{E}[L_{i_E}] = \lambda_{i_E} \mathbb{E}[W_{i_E}].$$

By $\mathbb{E}[L_{ij}]$ we denote the mean number of type i customers waiting in the queue during period j, for $i = 1, \dots, N$ and $j = 1, \dots, 4N$. Hence, we have, for $i = 1, \dots, N$,

$$\mathbb{E}[L_i] = \sum_{j=1}^{4N} q_j \, \mathbb{E}[L_{ij}].$$

During a gated service to Q_i, which is period $2i$, we distinguish between type i customers that will still receive service during this period (the ones before the gate), and type i customers that have to wait until the next visit of the server to the queue. These last ones are those type i customers that arrived *during* this period, and so, as the service discipline is gated, will not receive service any more in this period. By $\bar{L}_{i,2i}$ we denote the ones that will receive service, and by $\tilde{L}_{i,2i}$ the ones that have to wait. We have $L_{i,2i} = \bar{L}_{i,2i} + \tilde{L}_{i,2i}$. Analogously to (13.3.9), this gives, for $i = 1, \dots, N$,

$$\mathbb{E}[L_{i_G}] = q_{2i} \mathbb{E}[\bar{L}_{i,2i}] + \sum_{j=2N+2i+1}^{2i-1} q_j \, \mathbb{E}[L_{ij}],$$

$$\mathbb{E}[L_{i_E}] = q_{2i} \mathbb{E}[\tilde{L}_{i,2i}] + \sum_{j=2i+1}^{2N+2i} q_j \, \mathbb{E}[L_{ij}],$$

where the summation for $\mathbb{E}[L_{i_G}]$ should again be understood to be cyclical.

By making use of the PASTA property we obtain for the mean waiting time of a gated type i customer,

$$\mathbb{E}[W_{i_G}] = \frac{\mathbb{E}[L_{i_G}] - q_{2i} \mathbb{E}[\bar{L}_{i,2i}]}{q_{G,i}} \mathbb{E}[B_i] + \mathbb{E}[R_{2N+2i+1,2N-1}], \tag{13.3.10}$$

where R_{ij} denotes the residual time of the periods $i, i+1, \dots, i+j-1$. This expression can be interpreted as follows. A type i customer that will receive gated service, has to arrive in the periods $2N + 2i + 1, \dots, 4N, 1, \dots, 2i - 1$, consisting of $2N - 1$

periods, and in which the system is a fraction $q_{G,i}$ of the time. Arriving customers have to wait for the services of all customers already present in the queue, and for the time it takes before the server starts working on Q_i again. The latter time has mean duration $\mathbb{E}[R_{2N+2i+1,2N-1}]$. On arrival of a type i customer, there are on average $\sum_{j=2N+2i+1}^{2i-1} q_j\,\mathbb{E}[L_{ij}] = \mathbb{E}[L_{i_G}] - q_{2i}\mathbb{E}[\bar{L}_{i,2i}]$ type i customers already present in the system, all having mean service time $\mathbb{E}[B_i]$.

For the exhaustive type i customers we similarly find, for $i = 1, \ldots, N$:

$$\mathbb{E}[W_{i_E}] = \frac{\mathbb{E}[L_{i_E}]}{q_{E,i}}\,\mathbb{E}[B_i] + \frac{q_{2i} + \ldots + q_{2i+2N-1}}{q_{E,i}}\,\mathbb{E}[R_{2i,2N-1}] + \frac{q_{2i+2N}}{q_{E,i}}\,\mathbb{E}[B_i^{res}]. \quad (13.3.11)$$

This gives a system of equations for $\mathbb{E}[W_i]$, $\mathbb{E}[W_{i_G}]$, $\mathbb{E}[W_{i_E}]$, $\mathbb{E}[L_i]$, $\mathbb{E}[L_{i_G}]$, $\mathbb{E}[L_{i_E}]$ which can be solved, provided $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$ are known. The required equations for $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$ are derived in the next section.

Residual periods and conditional queue lengths. We now derive a set of equations relating $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$. At the end of an exhaustive service to Q_i, this queue is empty. From this moment on, the number of type i customers in the system increases at rate λ_i. As the residual duration of a period is in distribution equal to the amount of time already elapsed, so are their means. Hence

$$\lambda_i\,\mathbb{E}[R_{2N+2i+1,j}] = \frac{\sum_{k=1}^{j} q_{2N+2i+k}\mathbb{E}[L_{i,2N+2i+k}]}{\sum_{l=1}^{j} q_{2N+2i+l}},$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, 2N-1$.

The same idea applies to gated service. But now, at the beginning of a gated service to Q_i, there are no type i customers behind the gate, and from this moment on their number starts to increase at rate λ_i. This gives

$$\lambda_i\,\mathbb{E}[R_{2i,j}] = \frac{q_{2i}\mathbb{E}[\tilde{L}_{i,21}] + \sum_{k=1}^{j-1} q_{2i+k}\mathbb{E}[L_{i,2i+k}]}{\sum_{l=0}^{j-1} q_{2i+l}},$$

for $i = 1, \ldots, N$ and $j = 1, \ldots, 2N$, where an empty sum equals zero.

Another way to express $\mathbb{E}[R_{ij}]$ is to determine the expected residual duration of an (i,j) interval based on the mean queue lengths at an arbitrary epoch during this interval. First we consider the case $j = 1$. For i odd, a residual $(i,1)$ period is just a residual switch-over time, so

$$\mathbb{E}[R_{i,1}] = \mathbb{E}[S_i^{res}],\ \mathbb{E}[R_{2N+i,1}] = \mathbb{E}[S_i^{res}], \quad i = 1, 3, \ldots, 2N-1.$$

The residual time of a visit time at Q_i served gated satisfies

$$\mathbb{E}[R_{2i,1}] = \mathbb{E}[B_i^{res}] + \mathbb{E}[\bar{L}_{i,2i}]\mathbb{E}[B_i],$$

for $i = 1, \ldots, N$. First we have to wait for the residual service time of the customer in service, and then for the service of the $\bar{L}_{i,2i}$ customers in front of the gate. In case Q_i

is served exhaustively, we have to wait for the busy periods induced by the customers present, yielding for $i = 1, \ldots, N$:

$$\mathbb{E}[R_{2N+2i,1}] = \frac{\mathbb{E}[B_i^{res}]}{1-\rho_i} + \mathbb{E}[L_{i,2N+2i}]\frac{\mathbb{E}[B_i]}{1-\rho_i}.$$

We now consider $j = 2$, in which case it is convenient to introduce $q_{i,j}$ defined as the sum of $q_i, \ldots, q_{i+j-1}$. With probability $q_{i+1}/q_{i,2}$ the residual $(i,2)$ period is equal to the residual $(i+1,1)$ period. With probability $q_i/q_{i,2}$ it is equal to the residual $(i,1)$ period plus either a switch-over time (if i is even) or plus the busy period incurred by the number of customers present in the system and that of those arriving during the residual $(i,1)$ period (if i is odd). This yields, for $i = 1, \ldots, N$,

$$\begin{aligned}
\mathbb{E}[R_{2i-1,2}] &= \frac{q_{2i-1}}{q_{2i-1,2}}\left(\mathbb{E}[S_i^{res}](1+\rho_i) + \mathbb{E}[L_{i,2i-1}]\mathbb{E}[B_i]\right) + \frac{q_{2i}}{q_{2i-1,2}}\mathbb{E}[R_{2i,1}],\\
\mathbb{E}[R_{2i,2}] &= \frac{q_{2i}}{q_{2i,2}}\left(\mathbb{E}[R_{2i,1}] + \mathbb{E}[S_{i+1}]\right) + \frac{q_{2i+1}}{q_{2i,2}}\mathbb{E}[R_{2i+1,1}],\\
\mathbb{E}[R_{2N+2i-1,2}] &= \frac{q_{2N+2i-1}}{q_{2N+2i-1,2}}\left(\frac{\mathbb{E}[S_i^{res}]}{1-\rho_i} + \mathbb{E}[L_{i,2N+2i-1}]\frac{\mathbb{E}[B_i]}{1-\rho_i}\right) + \frac{q_{2N+2i}}{q_{2N+2i-1,2}}\mathbb{E}[R_{2N+2i,1}],\\
\mathbb{E}[R_{2N+2i,2}] &= \frac{q_{2N+2i}}{q_{2N+2i,2}}\left(\mathbb{E}[R_{2N+2i,1}] + \mathbb{E}[S_{i+1}]\right) + \frac{q_{2N+2i+1}}{q_{2N+2i,2}}\mathbb{E}[R_{2N+2i+1,1}],
\end{aligned}$$

where $2N + 2N + 1$ is assumed to equal 1 as the system is cyclic. This can be readily extended to $j > 2$: with probability $q_{i+1,j-1}/q_{i,j}$ the residual (i,j) period is equal to the residual $(i+1, j-1)$ period, and otherwise, it is equal to the residual $(i,1)$ period plus an $(i+1, j-1)$ period, the mean length of which is determined by the mean queue lengths during period i. The resulting expressions are rather lengthy, and therefore omitted.

We thus obtain sufficiently many equations to determine the unknowns $\mathbb{E}[R_{ij}]$ and $\mathbb{E}[L_{ij}]$, hence $\mathbb{E}[W_{i_G}]$ and $\mathbb{E}[W_{i_E}]$ follow from (13.3.10) and (13.3.11), respectively. Then, using (13.3.8), the mean waiting times $\mathbb{E}[W_i]$ follow.

13.3.5 Multi-type Branching Processes

In Resing [160] it is shown that for polling systems with gated or exhaustive service disciplines, the joint queue length process at the beginning of a visit time at a fixed queue, is a *Multi-type Branching Process* (MTBP), see Section 9.3.3. This gives expressions for the generating functions of the joint queue length distributions at these times.

If the service discipline in each queue satisfies the *branching property*, cf. [160, Property 1], then the queue length process at polling instants of a fixed queue is a multi-type branching process (MTBP) with immigration in each state. The gated, exhaustive, and G/E service disciplines do satisfy this branching property. Hence, if the server arrives at Q_i and it finds k_i customers there, during the visit of the server each of these k_i customers is replaced in an i.i.d. way by a random population, having probability generating function (pgf) $h_i(z_1, z_2, \ldots, z_N)$. For the gated and exhaustive service discipline, respectively, h_i is given in (9.3.9), and in (9.3.8), respectively, and the immigration pgf $g(z_1, z_2, \ldots, z_N)$ is as given in (9.3.11).

For the G/E discipline, we have to adapt the h_i and g. Replacing λ_j by λ_{ji} (see Section 13.3.3) in the h_i, we find:

$$(G) \qquad h_i^G(z_1, z_2, \dots, z_N) = \beta_i \left(\sum_{j=1}^{N} \lambda_{ji}(1 - z_j) \right), \tag{13.3.12}$$

$$(E) \qquad h_i^E(z_1, z_2, \dots, z_N) = \theta_i \left(\sum_{j \neq i} \lambda_{ji}(1 - z_j) \right), \tag{13.3.13}$$

and replacing λ_k by μ_{ki} in g gives:

$$g(z_1, z_2, \dots, z_N) = \prod_{i=1}^{N} \sigma_i \left(\sum_{k=1}^{i} \mu_{ki}(1 - z_k) + \sum_{k=i+1}^{N} \mu_{ki}(1 - f^{(k)}(z_1, z_2, \dots, z_N)) \right). \tag{13.3.14}$$

As the service disciplines are either gated or exhaustive, they do satisfy the Branching Property, with pgf h_i^G or h_i^E respectively, and immigration process $g(z_1, z_2, \dots, z_N)$.

In order to analyze the G/E discipline, we can now follow the same procedure as described in Section 9.3.3, with $2N$ queues and the λ_{ij} and μ_{ij} as given in (13.3.6) and (13.3.7) respectively. Using h_i^G and h_i^E of (13.3.12) and (13.3.13) respectively, we can by (9.3.10) calculate the offspring pgfs $f^{(i)}(z_1, z_2, \dots, z_{2N})$ for $i = 1, \dots, 2N$. In combination with $g(z_1, z_2, \dots, z_{2N})$ of (13.3.14), the pgf of the stationary distribution $\pi(j_1, j_2, \dots, j_{2N})$ follows by (9.3.12). This is the pgf of the number of customers present in the system at the moment that the server starts working on Q_1 according to the gated discipline. By renumbering the queues, we can in the same way find expressions for the moment that the server starts working on Q_i, $i = 2, \dots, 2N$, i.e. to Q_i, $i = 1, \dots, N$ served either gated or exhaustively. Also, we can use (9.3.14), in combination with (9.3.13), to find the LST of the waiting time distribution.

13.4 Comparison of gated and exhaustive strategies

We compare gated and exhaustive strategies for a system with two queues where $\lambda_1 = 0.6$, $\lambda_2 = 0.2$, and $\mathbb{E}[S_i] = 1$, $\mathbb{E}[S_i^{res}] = 1$, $\mathbb{E}[B_i] = 1$, $\mathbb{E}[B_i^{res}] = 1$, for $i = 1, 2$. This is the same example as in [210], in which, using MVA, mean waiting times are derived in case both queues are served purely gated, purely exhaustively and mixed gated and exhaustively. The performance of these strategies is shown in Table 13.1, together with the results for the G/E strategy, starting the cycle either at Q_1 or Q_2. Namely, as the G/E strategy is not symmetric, this leads to different mean waiting times, although the weighted sum (i.e., the mean amount of work in the system) is the same for both cases.

From Table 13.1 we see that the weighted sum of the expected waiting times is minimal, when both queues are served exhaustively, as is to be expected, since in this discipline the server does not unnecessarily switch to the other queue when there is still work at the current queue. Serving both queues gated gives the highest weighted sum of mean waiting times, but this strategy is more fair to the queues, as the difference in the mean waiting times is smaller. The weighted sum of the mean waiting times for the two strategies, where one queue is served exhaustively and the other gated, is bigger than for

Strategy Q_1 Q_2 Q_1 Q_2	$\mathbb{E}[W_1]$	$\mathbb{E}[W_2]$	$\rho_1\mathbb{E}[W_1]+$ $\rho_2\mathbb{E}[W_2]$	$\lvert\mathbb{E}[W_1]-$ $\mathbb{E}[W_2]\rvert$
E E E E	5.50	11.50	5.6	6.00
G G G G	12.77	9.69	9.6	3.08
E G E G	5.04	14.88	6.0	9.84
G E G E	6.64	13.12	6.6	6.48
G G E E	6.96	12.11	6.6	5.15
E G G E	6.84	12.47	6.6	5.63

Table 13.1: Comparison of the mean waiting times for a polling system with 2 queues.

purely exhaustive, and these strategies are less fair. On the other hand, the G/E strategy, starting the cycle at Q_1, is more fair than purely exhaustive, and the weighted sum of the mean waiting times is only a little larger. Starting the G/E strategy at Q_2 does not lead to more fairness.

The given strategies try to achieve fairness in waiting times at the expense of (slightly) higher waiting times. The price paid, however, seems to be far less than that in the case of the two-stage gated service discipline [185]; this strategy achieves more fairness than purely gated service, but the mean waiting times increase by roughly an expected cycle time.

Based on the intuition as explained in the introduction, we initially expected that the G/E discipline would lead to small differences in mean waiting times, and hence more fairness. This clearly turns out not to be the case, probably because the mean visit times at the queues differ quite a lot: in this example we have $\{\mathbb{E}[V_{1_G}], \mathbb{E}[V_{2_G}], \mathbb{E}[V_{1_E}], \mathbb{E}[V_{2_E}]\} = \{3, 1, 9, 3\}$. Hence, roughly three quarter of the time the system is in the exhaustive part of the cycle, which explains why the mean waiting times of the exhaustive case dominate the ones for the G/E case. Further research should provide more insight in the potential of achieving fairness by making cycles of one *or more* gated visits to the queues, followed by one or more cycles of exhaustive visits.

13.5 Conclusion and discussion

In this work we introduced the Gated/Exhaustive service discipline for polling systems. We derived a Pseudo Conservation Law for the mean waiting times, and used Mean Value Analysis and Multi-type Branching Processes to derive the (mean) waiting times at each of the queues, using the concept of Smart Customers. We numerically compared the mean waiting times in the Gated/Exhaustive discipline to number of mixtures of gated and exhaustive strategies, for the case of two queues. This disproved the idea that the Gated/Exhaustive discipline would lead to almost identical mean waiting times at all queues.

A possible topic for further study is to devise polling systems that do lead to better equalized mean waiting times. In Chapter 12 we introduced the κ-gated discipline for this reason. Also, one could think of the following:

(i) Other mixes of gated and exhaustive services, e.g., gated and exhaustive cycles in a ratio of $k_G : k_E$ for some k_G and k_E to be determined. We could vary the order in which the cycles are applied, e.g. $G-E-G-E-G$ repetitively, or we could take different ratios for each of the queues.

(ii) A mixed strategy of exhaustive and gated services, where the one chosen depends on a coin flip. There are multiple ways to do this. One way is to flip a coin at the beginning of a cycle and let this determine whether we do the entire round gated services or exhaustive services. Another way would be to decide this at each queue separately, at the moment the server arrives. For both cases, we could also let these probabilities depend on whether a gated or an exhaustive service was previously applied to the cycle respectively the queue, in that way letting the order of strategies become a Markov chain.

(iii) The fractional gated policy or fractional exhaustive policy [128]. In these strategies, for each of the customers it is decided whether or not it will be served during this visit of the server to the queue.

(iv) The Gated/Exhaustive policy applied to non-cyclic polling systems, like systems with fixed polling tables [17, 35] (e.g. Q_1, Q_2, Q_1, Q_3 repetitively). As such a model fits into the framework of branching type models [160] and that of smart customers, the analysis for a given polling table could be done along the same lines as the analysis discussed in this chapter.

The parameters in the above mentioned strategies could be chosen in such a way as to equalize the mean waiting times, by minimizing the difference between the largest and the smallest mean waiting time; they could also be chosen such that they optimize some other performance measure.

BIBLIOGRAPHY

[1] I.J.B.F. Adan, V.G. Kulkarni, and A.C.C. van Wijk. Server farms. In preparation, 2012.

[2] P. Alfredsson and J. Verrijdt. Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science*, 45(10):1416–1431, 1999.

[3] E. Altman, A. Khamisy, and U. Yechiali. On elevator polling with globally gated regime. *Queueing Systems*, 11(1):85–90, 1992.

[4] E.M. Alvarez, M.C. van der Heijden, and W.H.M. Zijm. The selective use of emergency shipments for service-contract differentiation. Beta working paper WP 322, Beta Research School for Operations Management and Logistics, 2010.

[5] T.W. Archibald, D. Black, and K.D. Glazebrook. An index heuristic for transshipment decisions in multi-location inventory systems based on a pairwise decomposition. *European Journal of Operational Research*, 192(1):69–78, 2007.

[6] T.W. Archibald, S.A.E. Sassen, and L.C. Thomas. An optimal policy for a two depot inventory problem with stock transfer. *Management Science*, 43(2):173–183, 1997.

[7] H. Arslan, S.C. Graves, and T.A. Roemer. A single-product inventory model for multiple demand classes. *Management Science*, 53(9):1486–1500, 2007.

[8] J.R. Artalejo, A. Economou, and M.J. Lopez-Herrero. Analysis of a multiserver queue with setup times. *Queueing Systems*, 51(1):53–76, 2005.

[9] B. Avi-Itzhak, H. Levy, and D. Raz. Quantifying fairness in queuing systems. *Probability in the Engineering and Informational Sciences*, 22(04):495–517, 2008.

[10] S. Axsäter. Modelling emergency lateral transshipments in inventory systems. *Management Science*, 36(11):1329–1338, 1990.

[11] S. Axsäter. A new decision rule for lateral transshipments in inventory systems. *Management Science*, 49(9):1168–1179, 2003.

[12] S. Axsäter. Evaluation of unidirectional lateral transshipments and substitutions in inventory systems. *European Journal of Operational Research*, 149(2):438–447, 2003.

[13] S. Axsäter. *Inventory control*. Number 90 in International Series in Operations Research & Management Science. Springer Verlag, 2006.

[14] S. Axsäter, C. Howard, and J. Marklund. A distribution inventory model with transshipments from a support warehouse. Working paper, Lund University, 2010.

[15] S. Axsäter, M. Kleijn, and T.G. de Kok. Stock rationing in a continuous review two-echelon inventory model. *Annals of Operations Research*, 126(1):177–194, 2004.

[16] A. Ayanso, M. Diaby, and S.K. Nair. Inventory rationing via drop-shipping in internet retailing: a sensitivity analysis. *European Journal of Operational Research*, 171(1):135–152, 2006.

[17] J. Baker and I. Rubin. Polling with a general-service order table. *IEEE Transactions on Communications*, 35(3):283–288, 1987.

[18] C.E. Bell. Optimal operation of an $M/M/2$ queue with removable servers. *Operations Research*, 28(5):1189–1204, 1980.

[19] S. Benjaafar and M. ElHafsi. Production and inventory control of a single product assemble-to-order system with multiple customer classes. *Management Science*, 52(12):1896–1912, 2006.

[20] S. Benjaafar and M. ElHafsi. A production-inventory system with both patient and impatient demand classes. *IEEE Transactions on Automation Science and Engineering*, 9(1):148–159, 2012.

[21] S. Benjaafar, M. ElHafsi, and T. Huang. Optimal control of a production-inventory system with both backorders and lost sales. *Naval Research Logistics*, 57(3):252–265, 2010.

[22] J.P.C. Blanc and R.D. van der Mei. Optimization of polling systems with Bernoulli schedules. *Performance Evaluation*, 22(2):139–158, 1995.

[23] M.A.A. Boon. A polling model with reneging at polling instants. *Annals of Operations Research*, To appear, DOI: 10.1007/s10479-010-0758-2, 2010.

[24] M.A.A. Boon and I.J.B.F. Adan. Mixed gated/exhaustive service in a polling model with priorities. *Queueing Systems*, 63(1):383–399, 2009.

[25] 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.

[26] 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. *Discrete Event Dynamic Systems*, 20(4):511–536, 2010.

[27] M.A.A. Boon, R.D. van der Mei, and E.M.M. Winands. Applications of polling systems. *Surveys in Operations Research and Management Science*, 16(2):67–82, 2011.

[28] M.A.A. Boon, A.C.C. van Wijk, I.J.B.F. Adan, and O.J. Boxma. A polling model with smart customers. *Queueing Systems*, 66(3):1–36, 2010.

[29] M.A.A Boon, E.M.M. Winands, I.J.B.F. Adan, and A.C.C. van Wijk. Closed-form waiting time approximations for polling systems. *Performance Evaluation*, 68(3):290–306, 2011.

[30] S.C. Borst and O.J. Boxma. Polling models with and without switchover times. *Operations Research*, 45(4):536–543, 1997.

[31] O.J. Boxma. Workloads and waiting times in single-server systems with multiple customer classes. *Queueing Systems*, 5(1):185–214, 1989.

[32] O.J. Boxma. Polling systems. In K.R. Apt, A. Schrijver, and N.M. Temme, editors, *From universal morphisms to megabytes: a Baayen space odyssey. Liber amicorum for P.C. Baayen*, pages 215–230. CWI, Amsterdam, 1994.

[33] O.J. Boxma, J. Bruin, and B.H. Fralix. Sojourn times in polling systems with various service disciplines. *Performance Evaluation*, 66(11):621–639, 2009.

[34] O.J. Boxma and W.P. Groenendijk. Pseudo-conservation laws in cyclic-service systems. *Journal of Applied Probability*, 24(4):949–964, 1987.

[35] O.J. Boxma, W.P. Groenendijk, and J.A. Weststrate. A pseudoconservation law for service systems with a polling table. *IEEE Transactions on Communications*, 38(10):1865–1870, 1990.

[36] O.J. Boxma, J. Ivanovs, K. Kosiński, and M. Mandjes. Lévy-driven polling systems and continuous-state branching processes. Eurandom report 2009-026, Eindhoven University of Technology, 2009.

[37] O.J. Boxma and M. Kelbert. Stochastic bounds for a polling system. *Annals of Operations Research*, 48(3):295–310, 1994.

[38] O.J. Boxma, H. Levy, and J.A. Weststrate. Efficient visit frequencies for polling tables: minimization of waiting cost. *Queueing Systems*, 9(1):133–162, 1991.

[39] O.J. Boxma, H. Levy, and J.A. Weststrate. Efficient visit orders for polling systems. *Performance Evaluation*, 18(2):103–123, 1993.

[40] 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(3):187–208, 1992.

[41] O.J. Boxma, A.C.C. van Wijk, and I.J.B.F. Adan. Polling systems with a gated/exhaustive discipline. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools (ValueTools 2008)*. ICST, 2008.

[42] G.A.J.F. Brouns. Optimal control of routing to two parallel finite capacity queues or two parallel Erlang loss systems with dedicated and flexible arrivals. SPOR report 03, Eindhoven University of Technology, 2002.

[43] Ö. Bulut and M.M. Fadıloğlu. Production control and stock rationing for a make-to-stock system with parallel production channels. *IIE Transactions*, 43(6):432–450, 2011.

[44] K.E. Caggiano, P.L. Jackson, J.A. Muckstadt, and J.A. Rappold. Efficient computation of time-based customer service levels in a multi-item, multi-echelon supply chain: A practical approach for inventory optimization. *European Journal of Operational Research*, 199(3):744–749, 2009.

[45] S. Carr and I. Duenyas. Optimal admission control and sequencing in a make-to-stock/make-to-order production system. *Operations Research*, 48(5):709–720, 2000.

[46] K.D. Cattani and G.C. Souza. Inventory rationing and shipment flexibility alternatives for direct market firms. *Production and Operations Management*, 11(4):441–457, 2002.

[47] S. Chen, J. Xu, and Y. Feng. A partial characterization of the optimal ordering/rationing policy for a periodic review system with two demand classes and backordering. *Naval Research Logistics*, 57(4):330–341, 2010.

[48] E. Chew, L. Lee, and S. Liu. Dynamic rationing and ordering policies for multiple demand classes. *OR Spectrum*, To appear, DOI: 10.1007/s00291-011-0239-2:1–25, 2011.

[49] W.K. Ching. Markov-modulated Poisson processes for multi-location inventory problems. *International Journal of Production Economics*, 53(2):217–223, 1997.

[50] W.K. Ching, R.H. Chan, and X.Y. Zhou. Circulant preconditioners for Markov-modulated Poisson processes and their applications to manufacturing systems. *SIAM Journal on Matrix Analysis and Applications*, 18(2):464–481, 1997.

[51] E.B. Çil, E.L. Örmeci, and F. Karaesmen. Effects of system parameters on the optimal policy structure in a class of queueing control problems. *Queueing Systems*, 61(4):273–304, 2009.

[52] A.J. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490, 1960.

[53] M.A. Cohen, N. Agrawal, and V. Agrawal. Winning in the aftermarket. *Harvard Business Review*, 84(5):129–138, 2006.

[54] M.A. Cohen, P.R. Kleindorfer, and H.L. Lee. Service constrained (s, S) inventory systems with priority demand classes and lost sales. *Management Science*, 4(34):482–499, 1988.

[55] M.A. Cohen and H.L. Lee. Out of touch with customer needs? Spare parts and after sales service. *Sloan Management Review*, 31(2):55–66, 1990.

[56] R.B. Cooper. Queues served in cyclic order: Waiting times. *Bell System Technical Journal*, 49(3):399–413, 1970.

[57] R.B. Cooper and G. Murray. Queues served in cyclic order. *Bell System Technical Journal*, 48(3):675–689, 1969.

[58] C. Das. Supply and redistribution rules for two-location inventory systems: One-period analysis. *Management Science*, 21(7):765–776, 1975.

[59] F. de Véricourt, F. Karaesmen, and Y. Dallery. Assessing the benefits of different stock-allocation policies for a make-to-stock production system. *Manufacturing & Service Operations Management*, 3(2):105–121, 2001.

[60] F. de Véricourt, F. Karaesmen, and Y. Dallery. Optimal stock allocation for a capacitated supply system. *Management Science*, 48(11):1486–1501, 2002.

[61] R. Dekker, R.M. Hill, M.J. Kleijn, and R.H. Teunter. On the $(S-1, S)$ lost sales inventory model with priority demand classes. *Naval Research Logistics*, 49(6):593–610, 2002.

[62] R. Dekker, M.J. Kleijn, and P.J. De Rooij. A spare parts stocking policy based on equipment criticality. *International Journal of Production Economics*, 56–57:69–77, 1998.

[63] V. Deshpande, M.A. Cohen, and K. Donohue. A threshold inventory rationing policy for service-differentiated demand classes. *Management Science*, 49(6):683–703, 2003.

[64] J.L. Dorsman, R.D. van der Mei, and E.M.M. Winands. A new method for deriving waiting-time approximations in polling systems with renewal arrivals. *Stochastic Models*, 27(2):318–332, 2011.

[65] J.L. Dorsman, R.D. van der Mei, and E.M.M Winands. Polling systems with batch service. *OR Spectrum*, To appear, DOI: 10.1007/s00291-011-0275-y, 2011.

[66] S. Duran, T. Liu, D. Simchi-Levi, and J.L. Swann. Optimal production and inventory policies of priority and price-differentiated customers. *IIE Transactions*, 39(9):845–861, 2007.

[67] R. Ehrhardt. (s, S) Policies for a dynamic inventory model with stochastic lead times. *Operations Research*, 32(1):121–132, 1984.

[68] M. Eisenberg. Queues with periodic service and changeover time. *Operations Research*, 20(2):440–451, 1972.

[69] P. Enders, I.J.B.F. Adan, A. Scheller-Wolf, and G.J. van Houtum. Inventory rationing for a system with heterogeneous customer classes. Working paper 2008-E2, Tepper School of Business, Carnegie Mellon University, 2008.

[70] D. Erlenkotter. Ford Whitman Harris and the economic order quantity model. *Operations Research*, 38(6):937–946, 1990.

[71] R.V. Evans. Sales and restocking policies in a single item inventory system. *Management Science*, 14(7):463–472, 1968.

[72] D. Everitt. Simple approximations for token rings. *IEEE Transactions on Communications*, 34(7):719–721, 1986.

[73] P.T. Evers. Heuristics for assessing emergency transshipments. *European Journal of Operational Research*, 129(2):311–316, 2001.

[74] M.M. Fadıloğlu and Ö. Bulut. An embedded markov chain approach to the analysis of inventory systems with backordering under rationing. Technical report, Bilkent University, Turkey, 2005.

[75] M.M. Fadıloğlu and Ö. Bulut. A dynamic rationing policy for continuous-review inventory systems. *European Journal of Operational Research*, 202(3):675–685, 2010.

[76] G. Fayolle, P.J.B. King, and I. Mitrani. The solution of certain two-dimensional Markov models. *Advances in Applied Probability*, 14(2):295–308, 1982.

[77] E.A. Feinberg and X. Zhang. Switching on and off the full capacity of an $M/M/\infty$ queue. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 7678–7683, 2011.

[78] M. Ferguson and Y. Aminetzah. Exact results for nonsymmetric token ring systems. *IEEE Transactions on Communications*, 33(3):223–231, 1985.

[79] M.J. Fischer, C.M. Harris, and J. Xie. An interpolation approximation for expected wait in a time-limited polling system. *Computers & Operations Research*, 27(4):353–366, 2000.

[80] W. Fischer and K. Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18(2):149–171, 1993.

[81] K.C. Frank, R.Q. Zhang, and I. Duenyas. Optimal policies for inventory systems with priority demand classes. *Operations Research*, 51(6):993–1002, 2003.

[82] C. Fricker and M.R. Jaibi. Monotonicity and stability of periodic polling models. *Queueing Systems*, 15(1):211–238, 1994.

[83] S.W. Fuhrmann. Performance analysis of a class of cyclic schedules. *Bell Laboratories Technical Memorandom*, 81–59531–1, 1981.

[84] S.W. Fuhrmann and R.B. Cooper. Stochastic decompositions in the $M/G/1$ queue with generalized vacations. *Operations Research*, 33(5):1117–1129, 1985.

[85] A. Gandhi, V. Gupta, M. Harchol-Balter, and M.A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67(11):1155–1171, 2010.

[86] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *Proceedings of the 2009 Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, pages 157–168. ACM, 2009.

[87] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5(2):79–141, 2003.

[88] J.P. Gayon, F. de Véricourt, and F. Karaesmen. Stock rationing in an $M/E_r/1$ multi-class make-to-stock queue with backorders. *IIE Transactions*, 41(12):1096–1109, 2009.

[89] Y. Gong and R. de Koster. A polling-based dynamic order picking system for online retailers. *IIE Transactions*, 40(11):1070–1082, 2008.

[90] J. Grahovac and A. Chakravarty. Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items. *Management Science*, 47(4):579–594, 2001.

[91] C.M. Grinstead and J.L. Snell. *Introduction to probability*. American Mathematical Society, 1997.

[92] W.P. Groenendijk. Waiting-time approximations for cyclic-service systems with mixed service strategies. In *Proceedings of the 12th International Teletraffic Congress (ITC)*, pages 1434–1441, 1989.

[93] D. Gross. Centralized inventory control in multilocation supply systems. In H.E. Scarf, D.M. Gilford, and M.W. Shelly, editors, *Multistage Inventory Models and Techniques*, pages 47–84. Stanford University Press, Stanford, CA, 1963.

[94] A.Y. Ha. Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, 43(8):1093–1103, 1997.

[95] A.Y. Ha. Stock-rationing policy for a make-to-stock production system with two priority classes and backordering. *Naval Research Logistics*, 44(5):457–472, 1997.

[96] A.Y. Ha. Stock rationing in an $M/E_k/1$ make-to-stock queue. *Management Science*, 46(1):77–87, 2000.

[97] G. Hadley and T.M. Whitin. *Analysis of inventory systems*. Prentice Hall, 1963.

[98] B. Hajek. Extremal splittings of point processes. *Mathematics of Operations Research*, 10(4):543–556, 1985.

[99] F.W. Harris. How many parts to make at once. *Factory, The Magazine of Management*, 10(2):135–136, 152, 1913.

[100] F.W. Harris. How many parts to make at once. *Operations Research*, 38(6):947–950, 1990.

[101] H.C. Haynsworth and B.A. Price. A model for use in the rationing of inventory during lead time. *Naval Research Logistics*, 36(4):491–506, 1989.

[102] H. Heffes and D. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications*, 4(6):856–868, 1986.

[103] Y.T. Herer and M. Tzur. The dynamic transshipment problem. *Naval Research Logistics*, 48(5):386–408, 2001.

[104] Y.T. Herer, M. Tzur, and E. Yücesan. The multilocation transshipment problem. *IIE Transactions*, 38(3):185–200, 2006.

[105] C. Howard, I.C. Reijnen, J. Marklund, and T. Tan. Using pipeline information in a multi-echelon spare parts inventory system. Working paper, Eindhoven University of Technology, 2010.

[106] X. Hu, I. Duenyas, and R. Kapuscinski. Optimal joint inventory and transshipment control under uncertain capacity. *Operations Research*, 56(4):881–897, 2008.

[107] M. Jain and A. Jain. Working vacations queueing model with multiple types of server breakdowns. *Applied Mathematical Modelling*, 34(1):1–13, 2010.

[108] A. Kaplan. Stock rationing. *Management Science*, 15(5):260–267, 1969.

[109] J. Keilson and L.D. Servi. The distributional form of Little's law and the Fuhrmann-Cooper decomposition. *Operations Research Letters*, 9(4):239–247, 1990.

[110] Y.L. Koçağa and A. Şen. Spare parts inventory management with demand lead times and rationing. *IIE Transactions*, 39(9):879–898, 2007.

[111] A.G. Konheim and H. Levy. Efficient analysis of polling systems. In *Proceedings of the Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'92)*, pages 2325–2331. IEEE, 1992.

[112] A.G. Konheim, H. Levy, and M.M. Srinivasan. Descendant set: an efficient approach for the analysis of polling systems. *IEEE Transactions on Communications*, 42(2/3/4):1245–1253, 1994.

[113] A.G. Konheim, I. Meilijson, and A. Melkman. Processor-sharing of two parallel lines. *Journal of Applied Probability*, 18(4):952–956, 1981.

[114] A.G. Konheim and B. Meister. Waiting lines and times in a system with polling. *Journal of the ACM*, 21(3):470–490, 1974.

[115] G. Koole. Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*, 30(3):323–339, 1998.

[116] G. Koole. Monotonicity in Markov reward and decision chains: Theory and applications. *Foundations and Trends in Stochastic Systems*, 1(1):1–76, 2006.

[117] A.A. Kranenburg. *Spare parts inventory control under system availability constraints*. PhD thesis, PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2006.

[118] A.A. Kranenburg and G.J. van Houtum. A multi-item spare parts inventory model with customer differentiation. Beta working paper, Beta Research School for Operations Management and Logistics, 2004.

[119] A.A. Kranenburg and G.J. van Houtum. Cost optimization in the $(S-1, S)$ lost sales inventory model with multiple demand classes. *Operations Research Letters*, 35(4):493–502, 2007.

[120] A.A. Kranenburg and G.J. van Houtum. A new partial pooling structure for spare parts networks. *European Journal of Operational Research*, 199(3):908–921, 2009.

[121] K.S. Krishnan and V.R.K. Rao. Inventory control in N warehouses. *Journal of Industrial Engineering*, 16(3):212–215, 1965.

[122] A. Kuczura. The interrupted Poisson process as an overflow process. *Bell System Technical Journal*, 52(3):437–448, 1973.

[123] A. Kukreja, C.P. Schmidt, and D.M. Miller. Stocking decisions for low-usage items in a multilocation inventory system. *Management Science*, 47(10):1371–1383, 2001.

[124] E. Kutanoglu. Insights into inventory sharing in service parts logistics systems with time-based service levels. *Computers & Industrial Engineering*, 54(3):341–358, 2008.

[125] E. Kutanoglu and M. Mahajan. An inventory sharing and allocation method for a multi-location service parts logistics network with time-based service levels. *European Journal of Operational Research*, 194(3):728–742, 2009.

[126] H.L. Lee. A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science*, 33(10):1302–1316, 1987.

[127] J.E. Lee and Y. Hong. A stock rationing policy in a (s,S)-controlled stochastic production system with 2-phase Coxian processing times and lost sales. *International Journal of Production Economics*, 83(3):299–307, 2003.

[128] H. Levy. Optimization of polling systems: the fractional exhaustive service method. Technical report, Report No. 120/88, Tel-Aviv University, Israel, 1988.

[129] H. Levy. Delay computation and dynamic behavior of non-symmetric polling systems. *Performance Evaluation*, 10(1):35–51, 1989.

[130] H. Levy and M. Sidi. Polling systems: applications, modeling, and optimization. *IEEE Transactions on Communications*, 38(10):1750–1760, 1990.

[131] S.A. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23(4):687–710, 1975.

[132] J. Liu and C.G. Lee. Evaluation of inventory policies with unidirectional substitutions. *European Journal of Operational Research*, 182(1):145–163, 2007.

[133] F.V. Lu and R.F. Serfozo. $M/M/1$ queueing decision processes with monotone hysteretic optimal policies. *Operations Research*, 32(5):1116–1132, 1984.

[134] C. Mack. The efficiency of N machines uni-directionally patrolled by one operative when walking time is constant and repair times are variable. *Journal of the Royal Statistical Society Series B*, 19(1):173–178, 1957.

[135] C. Mack, T. Murphy, and N.L. Webb. The efficiency of N machines uni-directionally patrolled by one operative when walking time and repair times are constants. *Journal of the Royal Statistical Society Series B*, 19(1):166–172, 1957.

[136] D.M. Markowitz, M.I. Reiman, and L.M. Wein. The stochastic economic lot scheduling problem: heavy traffic analysis of dynamic cyclic policies. *Operations Research*, 48(1):136–154, 2000.

[137] D.M. Markowitz and L.M. Wein. Heavy traffic analysis of dynamic cyclic policies: a unified treatment of the single machine scheduling problem. *Operations Research*, 49(2):246–270, 2001.

[138] K.S. Meier-Hellstern. The analysis of a queue arising in overflow models. *IEEE Transactions on Communications*, 37(4):367–372, 1989.

[139] P. Melchiors. Rationing policies for an inventory model with several demand classes and stochastic lead times. Technical report, University of Aarhus, Denmark, 2001.

[140] P. Melchiors. Restricted time-remembering policies for the inventory rationing problem. *International Journal of Production Economics*, 81–82:461–468, 2003.

[141] P. Melchiors, R. Dekker, and M.J. Kleijn. Inventory rationing in an (s, Q) inventory model with lost sales and two demand classes. *Journal of the Operational Research Society*, 51(1):111–122, 2000.

[142] R. Menich and R.F. Serfozo. Optimality of routing and servicing in dependent parallel processing systems. *Queueing Systems*, 9(4):403–418, 1991.

[143] B.L. Miller. A queueing reward system with several customer classes. *Management Science*, 16(3):234–245, 1969.

[144] S. Minner, E.A. Silver, and D.J. Robb. An improved heuristic for deciding on emergency transshipments. *European Journal of Operational Research*, 148(2):384–400, 2003.

[145] K.T. Möllering and U.W. Thonemann. An optimal critical level policy for inventory systems with two demand classes. *Naval Research Logistics*, 55(7):632–642, 2008.

[146] K.T. Möllering and U.W. Thonemann. An optimal constant level rationing policy under service level constraints. *OR Spectrum*, 32(2):319–341, 2010.

[147] I. Moon and S. Kang. Rationing policies for some inventory systems. *Journal of the Operational Research Society*, 49(5):509–518, 1998.

[148] S. Nahmias and W.S. Demmy. Operating characteristics of an inventory system with rationing. *Management Science*, 27(11):1236–1245, 1981.

[149] T.L. Olsen and R.D. van der Mei. Polling systems with periodic server routing in heavy traffic: renewal arrivals. *Operations Research Letters*, 33(1):17–25, 2005.

[150] F. Olsson. Optimal policies for inventory systems with lateral transshipments. *International Journal of Production Economics*, 118(1):175–184, 2009.

[151] F. Olsson. An inventory model with unidirectional lateral transshipments. *European Journal of Operational Research*, 200(3):725–732, 2010.

[152] C.G. Park, D.H. Han, B. Kim, and H.S. Jun. Queueing analysis of symmetric polling algorithm for DBA scheme in an EPON. In B.D. Choi, editor, *Proceedings 1st Korea-Netherlands Joint Conference on Queueing Theory and its Applications to Telecommunication Systems*, pages 147–154, Korea University, Seoul, 2005.

[153] C. Paterson, G. Kiesmüller, R. Teunter, and K. Glazebrook. Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2):125–136, 2011.

[154] E.L. Porteus. *Foundations of stochastic inventory theory*. Stanford Business Books, 2002.

[155] M.L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc. New York, NY, USA, 1994.

[156] G. Rabinowitz, A. Mehrez, C.-W. Chu, and B.E. Patuwo. A partial backorder control for continuous review (r, Q) inventory system with Poisson demand and constant lead time. *Computers & Operations Research*, 22(7):689–700, 1995.

[157] I.C. Reijnen, T. Tan, and G.J. van Houtum. Inventory planning for spare parts networks with delivery time requirements. Beta working paper, Beta Research School for Operations Management and Logistics, 2009.

[158] M.I. Reiman and B. Simon. An interpolation approximation for queueing systems with Poisson input. *Operations Research*, 36(3):454–469, 1988.

[159] M.I. Reiman and L.M. Wein. Dynamic scheduling of a two-class queue with setups. *Operations Research*, 46(4):532–547, 1998.

[160] J.A.C. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13(4):409–426, 1993.

[161] P. Rijk. Multi-item, multi-location stock control with capacity constraints for the fieldstock of service parts at Océ. Master's thesis, Eindhoven University of Technology, 2007.

[162] L.W. Robinson. Optimal and approximate policies in multiperiod, multilocation inventory models with transshipment. *Operations Research*, 38(2):278–295, 1990.

[163] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.

[164] J.G. Shanthikumar. On stochastic decomposition in $M/G/1$ type queues with generalized server vacations. *Operations Research*, 36(4):566–569, 1988.

[165] C.C. Sherbrooke. Multiechelon inventory systems with lateral supply. *Naval Research Logistics*, 39(1):29–40, 1992.

[166] A.W. Shogan. A single server queue with arrival rate dependent on server breakdowns. *Naval Research Logistics Quarterly*, 26(3):487–497, 1979.

[167] R. Shoham and U. Yechiali. Elevator-type polling systems. *ACM Sigmetrics Performance Evaluation Review*, 20(1):255–257, 1992.

[168] B. Simon. A simple relationship between light and heavy traffic limits. *Operations Research*, 40(Supplement 2):S342–S345, 1992.

[169] D.R. Smith and W. Whitt. Resource sharing for efficiency in traffic systems. *Bell System Technical Journal*, 60(1):39–55, 1981.

[170] M.J. Sobel and R.Q. Zhang. Inventory policies for systems with stochastic and deterministic demand. *Operations Research*, 49(1):157–162, 2001.

[171] S. Stidham Jr. Optimal control of admission to a queueing system. *IEEE Transactions on Automatic Control*, 30(8):705–713, 1985.

[172] S. Stidham Jr and R.R. Weber. A survey of Markov decision models for control of networks of queues. *Queueing Systems*, 13(1):291–314, 1993.

[173] D.S. Szarkowicz and T.W. Knowles. Optimal control of an $M/M/s$ queueing system. *Operations Research*, 33(3):644–660, 1985.

[174] G. Tagaras. Effects of pooling on the optimization and service levels of two-location inventory systems. *IIE Transactions*, 21(3):250–257, 1989.

[175] G. Tagaras and M.A. Cohen. Pooling in two-location inventory systems with non-negligible replenishment lead times. *Management Science*, 38(8):1067–1083, 1992.

[176] H. Takagi. *Analysis of polling systems*. MIT Press Cambridge, MA, USA, 1986.

[177] H. Takagi. Queuing analysis of polling models. *ACM Computing Surveys*, 20(1):5–28, 1988.

[178] Y. Tang, D.S. Xu, and W.H. Zhou. Inventory rationing in a capacitated system with backorders and lost sales. In *2007 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1579–1583. IEEE, 2007.

[179] H. Tempelmeier. Supply chain inventory optimization with two customer classes in discrete time. *European Journal of Operational Research*, 174(1):600–621, 2006.

[180] R.H. Teunter and W.K. Klein Haneveld. Dynamic inventory rationing strategies for inventory systems with two demand classes, Poisson demand and backordering. *European Journal of Operational Research*, 190(1):156–178, 2008.

[181] H.G.H. Tiemessen, M. Fleischmann, G.J. van Houtum, E. Pratsini, and J.A.E.E. van Nunen. Dynamic demand fulfillment in spare parts networks with multiple customer classes. Working paper, Eindhoven University of Technology, 2011.

[182] H.C. Tijms. *Stochastic models: an algorithmic approach*. Wiley New York, 1994.

[183] D.M. Topkis. Optimal ordering and rationing policies in a nonstationary dynamic inventory model with n demand classes. *Management Science*, 15(3):160–176, 1968.

[184] D. Towsley, P.D. Sparaggis, and C.G. Cassandras. Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control*, 37(9):1446–1451, 1992.

[185] R.D. van der Mei and J.A.C. Resing. Analysis of polling systems with two-stage gated service: fairness versus efficiency. In L. Mason, T. Drwiega, and J. Yan, editors, *Proceedings of the 20th international teletraffic conference on Managing traffic performance in converged networks (ITC2007)*, pages 544–555, 2007.

[186] R.D. van der Mei and A. Roubos. Polling models with multi-phase gated service. *Annals of Operations Research*, To appear, DOI: 10.1007/s10479-011-0921-4, 2011.

[187] R.D. van der Mei and E.M.M. Winands. A note on polling models with renewal arrivals and nonzero switch-over times. *Operations Research Letters*, 36(4):500–505, 2008.

[188] W. van der Weij, S. Bhulai, and R. van der Mei. Optimal scheduling policies for the limited processor sharing queue. Technical report, VU University Amsterdam, 2009.

[189] N.M. van Dijk and E. van der Sluis. To pool or not to pool in call centers. *Production and Operations Management*, 17(3):296–305, 2008.

[190] W. van Jaarsveld and R. Dekker. Finding optimal policies in the $(S-1,S)$ lost sales inventory model with multiple demand classes. *Report Econometric Institute*, EI 2009–14, 2009.

[191] A.C.C. van Wijk. Optimal control of a head-of-line processor sharing model with regular and opportunity customers. In *Proceedings of the 8th Conference on Stochastic Models of Manufacturing and Service Operations (SMMSO), Kusadasi, Turkey*, pages 215–222, 2011.

[192] A.C.C. van Wijk, I.J.B.F. Adan, O.J. Boxma, and A.C. Wierman. Fairness and efficiency for polling models with the κ-gated service discipline. *Performance Evaluation*, To appear, 2012.

[193] A.C.C. van Wijk, I.J.B.F. Adan, and G.J. van Houtum. Optimal lateral transshipment policy for a two location inventory problem. Eurandom report 2009-027, Eindhoven University of Technology, 2009.

[194] A.C.C. van Wijk, I.J.B.F. Adan, and G.J. van Houtum. Optimal policy for a multi-location inventory system with a quick response warehouse. Eurandom report 2011-013, Eindhoven University of Technology, 2011.

[195] A.C.C. van Wijk, I.J.B.F. Adan, and G.J. van Houtum. Approximate evaluation of multi-location inventory models with lateral transshipments and hold back levels. *European Journal of Operational Research*, 218(3):624–635, 2012.

[196] A.F. Veinott Jr. Optimal policy in a dynamic, single product, nonstationary inventory model with several demand classes. *Operations Research*, 13(5):761–778, 1965.

[197] A.F. Veinott Jr. The status of mathematical inventory theory. *Management Science*, 12(11):745–777, 1966.

[198] O. Vicil and P. Jackson. An exact optimal solution to a threshold inventory rationing model for two priority demand classes. Technical report, Cornell University Operations Research and Industrial Engineering, 2006.

[199] O. Vicil and P. Jackson. An exact optimal solution to a threshold inventory rationing model for multiple priority demand classes. Technical report, Cornell University Operations Research and Industrial Engineering, 2006.

[200] V.M. Vishnevskii and O.V. Semenova. Mathematical methods to study the polling systems. *Automation and Remote Control*, 67(2):173–220, 2006.

[201] Y. Wang, M.A. Cohen, and Y.S. Zheng. Differentiating customer service on the basis of delivery lead-times. *IIE Transactions*, 34(11):979–989, 2002.

[202] K.M. Wasserman and N. Bambos. Optimal server allocation to parallel queues with finite-capacity buffers. *Probability in the Engineering and Informational Sciences*, 10(02):279–285, 1996.

[203] R.R. Weber and S. Stidham. Optimal control of service rates in networks of queues. *Advances in Applied Probability*, 19(1):202–218, 1987.

[204] K.E. Wee and M. Dada. Optimal policies for transshipping inventory in a retail network. *Management Science*, 51(10):1519–1533, 2005.

[205] J.A. Weststrate. *Analysis and optimization of polling models*. PhD thesis, Tilburg University, Tilburg, The Netherlands, 1992.

[206] W. Whitt. An interpolation approximation for the mean workload in a $GI/G/1$ queue. *Operations Research*, 37(6):936–952, 1989.

[207] E.M.M. Winands. On polling systems with large setups. *Operations Research Letters*, 35(5):584–590, 2007.

[208] E.M.M. Winands. *Polling, production & priorities*. PhD thesis, Eindhoven University of Technology, 2007.

[209] E.M.M. Winands. Branching-type polling systems with large setups. *OR Spectrum*, 33(1):77–97, 2011.

[210] E.M.M. Winands, I.J.B.F. Adan, and G.J. van Houtum. Mean value analysis for polling systems. *Queueing Systems*, 54(1):35–44, 2006.

[211] R.W. Wolff. *Stochastic modeling and the theory of queues*. Prentice hall, Englewood Cliffs, N.J., 1989.

[212] H. Wong, G.J. van Houtum, D. Cattrysse, and D.V. Oudheusden. Multi-item spare parts systems with lateral transshipments and waiting time constraints. *European Journal of Operational Research*, 171(3):1071–1093, 2006.

[213] H. Wong, G.J. van Houtum, D. Cattrysse, and D. Van Oudheusden. Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints. *Journal of the Operational Research Society*, 56(12):1419–1430, 2005.

[214] J. Xu, S. Chen, B. Lin, and R. Bhatnagar. Optimal production and rationing policies of a make-to-stock production system with batch demand and backordering. *Operations Research Letters*, 38(3):231–235, 2010.

[215] K. Xu, P.T. Evers, and M.C. Fu. Estimating customer service in a two-location continuous review inventory model with emergency transshipments. *European Journal of Operational Research*, 145(3):569–584, 2003.

[216] H. Zhao, V. Deshpande, and J.K. Ryan. Inventory sharing and rationing in decentralized dealer networks. *Management Science*, 51(4):531–547, 2005.

[217] H. Zhao, V. Deshpande, and J.K. Ryan. Emergency transshipment in decentralized dealer networks: When to send and accept transshipment requests. *Naval Research Logistics*, 53(6):547–567, 2006.

[218] H. Zhao, J.K. Ryan, and V. Deshpande. Optimal dynamic production and inventory transshipment policies for a two-location make-to-stock system. *Operations Research*, 56(2):400–410, 2008.

[219] W. Zhou, C.Y. Lee, and D. Wu. Optimal control of a capacitated inventory system with multiple demand classes. *Naval Research Logistics*, 58(1):43–58, 2011.

[220] Y. Zhou and X. Zhao. A two-demand-class inventory system with lost-sales and backorders. *Operations Research Letters*, 38(4):261–266, 2010.

[221] Y. Zhou and X. Zhao. Optimal policies of an inventory system with multiple demand classes. *Tsinghua Science & Technology*, 15(5):498–508, 2010.

[222] P.H. Zipkin. *Foundations of inventory management*. McGraw-Hill/Irwin, New York, 2000.

SUMMARY

Pooling and Polling: Creation of Pooling in Inventory and Queueing Models

The subject of the present monograph is the 'Creation of Pooling in Inventory and Queueing Models'. This research consists of the study of sharing a scarce resource (such as inventory, server capacity, or production capacity) between multiple customer classes. This is called pooling, where the goal is to achieve cost or waiting time reductions. For the inventory and queueing models studied, both theoretical, scientific insights are generated, as well as strategies which are applicable in practice.

This monograph consists of two parts: *pooling* and *polling*. In the first part, pooling is applied to multi-location inventory models. It is studied how cost reduction can be achieved by the use of stock transfers between local warehouses, so-called lateral transshipments. In this way, stock is pooled between the warehouses. The setting is motivated by a spare parts inventory network, where critical components of technically advanced machines are kept on stock, to reduce down time durations. We create insights into the question when lateral transshipments lead to cost reductions, by studying several models.

Firstly, a system with two stock points is studied, for which we completely characterize the structure of the optimal policy, using dynamic programming. For this, we formulate the model as a Markov decision process. We also derived conditions under which simple, easy to implement, policies are always optimal, such as a hold back policy and a complete pooling policy. Furthermore, we identified the parameter settings under which cost savings can be achieved. Secondly, we characterize the optimal policy structure for a multi-location model where only one stock point issues lateral transshipments, a so-called quick response warehouse. Thirdly, we apply the insights generated to the general multi-location model with lateral transshipments. We propose the use of a hold back policy, and construct a new approximation algorithm for deriving the performance characteristics. It is based on the use of interrupted Poisson processes. The algorithm is shown to be very accurate, and can be used for the optimization of the hold back levels, the parameters of this class of policies. Also, we study related inventory models, where a single stock point servers multiple customers classes.

Furthermore, in the first part, the pooling of server capacity is studied. For a two queue model where the head-of-line processor sharing discipline is applied, we derive the optimal control policy for dividing the servers attention, as well as for accepting customers. Also, a server farm with an infinite number of servers is studied, where servers can be turned off after a service completion in order to save costs. We characterize the

optimal policy for this model.

In the second part of the thesis, polling models are studied, which are queueing systems where multiple queues are served by a single server. An application is the production of multiple types of products on a single machine. In this way, the production capacity is pooled between the product types. For the classical polling model, we derive a closed-form approximation for the mean waiting time at each of the queues. The approximation is based on the interpolation of light and heavy traffic results. Also, we study a system with so-called smart customers, where the arrival rate at a queue depends on the position of the server. Finally, we invent two new service disciplines (the gated/exhaustive and the κ-gated discipline) for polling models, designed to yield 'fairness and efficiency' in the mean waiting times. That is, they result in almost equal mean waiting times at each of the queues, without increasing the weighted sum of the mean waiting times too much.

Curriculum vitae

Alexandra Cornelia Catharina (Sandra) van Wijk was born in Leuven, Belgium, on May 23, 1985. Since 1987 she has been living in The Netherlands. In 2002 she finished pre-university education at Jacob Roelandslyceum, Boxtel, and started her studies in Applied Mathematics at Eindhoven University of Technology. She obtained her BSc degree in 2006, with a thesis on waiting times at traffic lights, while already enrolled in the Master Industrial and Applied Mathematics. She obtained her MSc degree with a specialization in Statistics, Probability and Operations Research in 2007, after an internship at Philips Research Eindhoven, studying the entropy of hidden Markov models. Moreover, she conducted an internship in the operations research group of the Mathematics department, studying polling models.

On November 1, 2007, she started her PhD project at the same university, in a multidisciplinary cooperation between the Mathematics department (Operations Research group) and the Industrial Engineering department (Operations, Planning, Accounting and Control group), while also working at research institute Eurandom (Queueing and Performance Analysis track). The main topic of her research was 'the creation of pooling in inventory and queueing models'. She worked under the supervision of Onno Boxma, Ivo Adan, Ton de Kok, and Geert-Jan van Houtum. In 2012, she spent three months abroad at the Computer Science department of Carnegie Mellon University in Pittsburgh (USA), working together with Mor Harchol-Balter and Alan Scheller-Wolf.

Next to her studies, she worked as a teaching assistant at the university, teaching various courses for the Mathematics department, and she teaches high school students in preparation for their final exams. Moreover, she is an enthusiastic korfball player, winning the European student championship korfball in 2007.

The PhD project ends with the realization of this thesis, which Sandra defends on April 24, 2012.