

CWI Tracts

Managing Editors

K.R. Apt (CWI, Amsterdam)
M. Hazewinkel (CWI, Amsterdam)
J.K. Lenstra (Eindhoven University of Technology)

Editorial Board

W. Albers (Enschede)
P.C. Baayen (Amsterdam)
R.C. Backhouse (Eindhoven)
E.M. de Jager (Amsterdam)
M.A. Kaashoek (Amsterdam)
M.S. Keane (Amsterdam)
H. Kwakernaak (Enschede)
J. van Leeuwen (Utrecht)
P.W.H. Lemmens (Utrecht)
M. van der Put (Groningen)
M. Rem (Eindhoven)
H.J. Sips (Delft)
M.N. Spijker (Leiden)
H.C. Tijms (Amsterdam)

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Telephone 31 - 20 592 9333, telex 12571 (macr nl),
telefax 31 - 20 592 4199

Stochastic scheduling and dynamic programming

G.M. Koole

1991 Mathematics Subject Classification: 90B22, 90C40, 90B35.
ISBN 90 6196 457 1
NUGI-code: 811

Copyright © 1995, Stichting Mathematisch Centrum, Amsterdam
Printed in the Netherlands

Preface

This is a revision of my Ph.D. thesis, which was written in the winter of 1991-92, based on four years of research at Leiden University. During that time I studied various routing and scheduling problems, for which I (partially) characterized the optimal policies using the same technique: dynamic programming.

Over the last three years I found several related articles of which I was previously unaware, some new interesting results appeared, and I strengthened a few results myself. Based on that I prepared this revision. The sections which changed the most are 1.8, 1.9, 2.4, 3.7, and appendix A.

I would like to thank three people who contributed greatly to my thesis: my advisor Arie Hordijk, for his guidance, my office-mate Floske Spieksma, and Carly Giezen, for her support outside the office.

Sophia Antipolis, June 1995

Ger Koole

Contents

Introduction	1
1. Models with Markov Arrival Processes	5
1.1. Markov Arrival Processes	5
1.2. Symmetric customer assignment model	6
1.3. Customer assignment model without waiting room	9
1.4. Discrete customer assignment model	10
1.5. Pathwise optimality	12
1.6. Customer assignment model with rejection	14
1.7. Series of parallel processors	15
1.8. Customer assignment model with workloads	17
1.9. Customer assignment model without information	20
1.10. Markov Arrival Processes with multiple customer classes and server vacations	21
1.11. Server assignment model with a single server	23
1.12. Server assignment model with multiple servers	29
2. Models with Markov Decision Arrival Processes	33
2.1. Markov Decision Arrival Processes	33
2.2. Symmetric customer assignment model	34
2.3. Tandems of customer assignment models	36
2.4. Networks of customer assignment models with workloads	42
2.5. Server assignment model with multiple servers	44
2.6. Tandems of server assignment models with a single server	48
2.7. Tandems of server assignment models with a single server and identical centers	53
3. Models with Dependent Markov Decision Arrival Processes	57
3.1. Dependent Markov Decision Arrival Processes	57
3.2. Asymmetric customer assignment model	58
3.3. Symmetric customer assignment models	61
3.4. Customer assignment models without waiting room	66
3.5. Customer assignment models with rejections	68
3.6. Server assignment model with multiple servers	71
3.7. Server assignment model with a single server and a finite source	72
4. Proofs of dynamic programming results	75
4.1. Proofs of chapter 1	75
4.2. Proofs of chapter 2	84

4.3. Proofs of chapter 3	88
5. Uniformization	103
5.1. Introduction	103
5.2. Uniformization with fixed parameter	105
5.3. Continuous-time Bellman equation	107
Appendices	113
A. The approximation of point processes	113
B. Phase-type distributions of DFR/IFR distributions	118
C. Majorization	121
D. Computational issues	124
References	127
Index	133

Introduction

The title of this monograph consists of two parts, *stochastic scheduling* and *dynamic programming*. The former refers to a class of models, the latter refers to the method used to find optimal policies for these models. The models studied here can be divided in two classes: those in which customers at arrival are to be assigned to one of a number of queues and those in which one or more servers are to be assigned to different customer classes or queues. Of great importance is the way in which customers arrive at the stations. Models with independent arrival streams are studied in chapter 1. Then we allow the arrival stream to depend on the numbers of customers in the queues in such a way that controllable networks can be modeled with it. These and other network results can be found in chapter 2. In chapter 3 we generalize the arrival process even more, for example to include finite source models. Many results of chapter 1 and 2 are special cases of the results of chapter 3. Chapter 4 contains the proofs of the dynamic programming results. Chapter 5 considers methods by which we can translate the discrete-time results of the chapters 1 to 3 to continuous-time results. We conclude with four appendices, respectively on weak convergence of arrival streams, on phase-type distributions with a monotone failure rate, on majorization, and on algorithms to compute optimal policies.

Summarizing, chapter 1 can be seen as an introduction, chapter 2 contains the network results, and in chapter 3 the dynamic programming results are handled in their greatest generality.

Chapter 1 starts by introducing the *Markov Arrival Process* (MAP), an arrival process based on a Markov process. In appendix A it is shown that the class of Markov Arrival Processes is dense in the class of all independent arrival processes. The MAP is taken as input to a model consisting of m parallel queues, with possibly finite buffers, each with their own exponential server. These types of models, in which each arriving customer is to be assigned to one of the queues, are called *customer assignment models*. When the service rates of all servers are equal, the policy that assigns arriving customers to the shortest non-full queue (the SQP) is optimal for a large class of cost functions, including the total number of customers. This is shown in section 1.2, by inductively proving properties of the discrete-time dynamic programming equation. A related model has no buffers at the servers, but different service rates. Here arriving customers should be sent to the fastest available server. For both models, there is a complete characterization of the allowable cost functions, to be found in appendix C.

The previous results are only interesting in continuous time, due to the way of modeling. Section 1.4 considers a simple symmetric discrete-time model with simultaneous events, in which the SQP is optimal. In section 1.5 we generalize the result of section 1.2 to pathwise optimality of the SQP. So far, all cost functions depend on the number of customers in the queues. We can also consider the number of departed customers. In section 1.6 it is shown that the SQP is again optimal, and that we can allow rejections. Section 1.7 deals with maintenance models closely related to the models of the earlier sections.

In sections 1.2 to 1.7 the information available to the controller is the numbers of customers in each queue. In section 1.8 the amount of work in each queue is known. Here the policy that assigns to the queue with the shortest workload (the SWP) is optimal. In section 1.9 there is no information at all, not even on previous assignments. It is shown that the optimal policy divides the arrivals equally among the queues. It is the only result in this chapter not obtained by dynamic programming.

Now we move to the *server assignment models*. First we generalize the MAP to be able to include server vacations and arrivals in multiple classes. In section 1.11 and 1.12 we deal with the following model. Customers arrive in m different classes, and all customers in the same class have an exponential service time with the same mean. There are one or more identical servers available, which have to be assigned to the customers present. Both models with a single and with multiple servers are studied, giving conditions on the cost functions for list policies to be optimal. As special cases we find the following well known results. In the single server case the μc -rule minimizes the weighted number of customers. In the multiple server case the makespan is minimized by the LEPT policy (LEPT stands for longest expected processing time first). In the single server case we generalize the results to IFR and DFR service time distributions.

In chapter 2 we consider controllable tandems and networks of centers, each center being of one of the types discussed in chapter 1. Consider the last center in a tandem system, in which the control in each center is allowed to depend on the state of the whole network. Then we cannot use the optimality results of chapter 1 to obtain the optimal policy in the last center, because the arrivals, through the control in the previous centers, depend on the state of that center. With a *Markov Decision Arrival Process* (MDAP) we deal with this type of dependency, by using it to model all but the last center with it. It is shown that the SQP, for the model of section 1.2, is still optimal for this type of arrival stream. An interesting question is what the optimal policy is in the first of two centers in tandem. Some results and counterexamples are given in section 2.3. We also analyze the model where the policies are allowed to depend on the workloads. It appears that the results are stronger than the results for the model based on the numbers of customers.

The results on the server assignment models are not as easily generalized to arrivals according to an MDAP. More precisely, the generalization holds only if the policy which is optimal in the case of an MAP processes the jobs in decreasing order of expected processing times. This means that LEPT also

minimizes the makespan for dependent arrivals, but in the single server case the μc -rule is only optimal if it coincides with LEPT. Counterexamples are given in the case that it does not. In the sections 2.6 and 2.7 we consider tandem systems with each center having a single server. Section 2.6 deals with heavy traffic results. In section 2.7 we assume that the service time distribution of each customer is the same in both centers. Then we have the striking result that each work-conserving policy minimizes the makespan.

Chapter 3 starts with generalizing the MDAP to a *Dependent Markov Decision Arrival Process* (DMDAP). Now we can also model a finite source. In section 3.2 a customer assignment model is studied with asymmetric service times. The following partial characterization of the optimal policy is given: if queue k has less customers and a faster server than queue l , then an arriving customer can better be assigned to queue k than to queue l . From this result the results of section 1.2 and 1.3 follow. In section 3.3 we study again symmetric models, but now with batch arrivals, and with non-routable arrivals and an assignable server. In section 3.4 we consider a model with asymmetric servers, multiple customer classes and no buffer space. Each customer has blocking costs, depending on its class. Various monotonicity results are proved. Then we move again to the server assignment models. Results for the multiple server case are generalized to partial availability of servers. Here we cannot model a finite source. We end the chapter by considering a model with a single server and a finite source.

Most results are obtained by proving structural properties for discrete-time models. Typically, we formulate the dynamic programming equation and prove certain inequalities by induction, provided that they hold for the cost functions. In most models we have an inequality giving the optimal policy, an inequality showing monotonicity, and, in the customer assignment models, an inequality showing symmetry of the costs, all in n steps. The decision points of the discrete-time model are the jump times of the original continuous-time model. In fact, the sojourn times of the embedded chain are all exponentially distributed with parameter α . By increasing this uniformization parameter we show in section 5.3 that the optimal policies in the continuous-time models have the same properties as the optimal policies in the discrete-time models. If the optimal policy is *myopic*, that is, the same decision rule is optimal for each horizon, then we can prove the continuous-time results by considering a fixed α . This is the subject of section 5.2. All models considered in chapter 1 have myopic optimal policies.

The main result of appendix A is already discussed. There multi-dimensional phase-type distributions are used, and it is shown that they are dense in the class of all distributions. In appendix B we deal with one-dimensional phase-type distributions. By the Markovian structure of our models, we cannot deal with general service time distributions. To prove results for (service time) distributions with monotone failure rates, we need a characterization for the approximating phase-type distributions. This is provided in appendix B.

As we said, our inductive results give conditions on cost functions. For several customer assignment models, complete characterizations of the sets of allowable cost functions are given in appendix C.

In some models where optimal policies could not be given, numerical experiments were done. Also to provide counterexamples computational methods were used. Appendix D deals with these methods.

Most models of chapter 1 can already be found in the literature. Existing results are generalized, for example to finite buffers and to more general cost functions. Detailed discussions of the existing literature can be found in the appropriate sections of chapter 1. The main generalizations of the chapters 2 and 3 are the dependent arrival processes. Chapter 5 adapts existing results for use in the models of chapters 1, 2 and 3. Also in the appendices several new results are presented.

Models with Markov Arrival Processes

1.1. Markov Arrival Processes

We start this chapter by introducing the arrival process.

1.1.1. Definition. (Markov Arrival Process) Let Λ be the countable state space of a Markov process with transition intensities λ_{xy} with $x, y \in \Lambda$. When this process moves from x to y with probability q_{xy} an arrival occurs. We call the triple (Λ, λ, q) a Markov Arrival Process (MAP).

Arrival processes with the arrivals on the jumps of a Markov process were first introduced by Rudemo [61]. For computational results we refer to an article by Neuts [51] and to chapter 5 of his latest book [53].

With the MAP the departure process of most queueing systems with exponentially distributed sojourn times can easily be modeled, which can then be used as input to another system. As an example, take the $M|M|1$ queue with a Poisson($\tilde{\lambda}$) arrival stream and service intensity μ . Construct the MAP (Λ, λ, q) , corresponding to the departures, as follows: take $\Lambda = \{0, 1, \dots\}$, $\lambda_{ii+1} = \tilde{\lambda}$ and $\lambda_{ii-1} = \mu$ if $i \geq 1$. All other transitions have intensity 0. Take $q_{ii+1} = 0$, $q_{ii-1} = 1$.

Now we show how to model a phase-type renewal process with an MAP.

Phase-type renewal processes. Assume we have a renewal process with independent interarrival times of phase-type, as discussed in Neuts [52]. Phase-type distributions are defined as follows. We have a Markov process with $m+1$ states, where state $m+1$ is absorbing, the other m states are transient. The transition intensity from state x to y is denoted by t_{xy} , α_x is the probability that the system starts in state x . The time until absorption is the phase-type distribution. Assume $\alpha_{m+1} = 0$, i.e. there is no atom at 0. To model this renewal process with an MAP (Λ, λ, q) , we have to take the parameters as follows: $\Lambda = \{1, \dots, m\}$, $\lambda_{xy} = t_{xy} + t_{xm+1}\alpha_y$ and $q_{xy} = (t_{xm+1}\alpha_y)/(t_{xy} + t_{xm+1}\alpha_y)$. We see that when the original state moves to $m+1$, the process is immediately restarted and moves to state y with probability α_y .

Also the Markov Modulated Poisson Process (MMPP) can be modeled with an MAP.

Markov Modulated Poisson Process. An MMPP is governed by a Markov process with state space $\tilde{\Lambda}$ and transition intensities $\tilde{\lambda}_{xy}$. When the system is in state x customers arrive with intensity μ_x . As this does not change the arrival process we can assume $\tilde{\lambda}_{xx} = 0$ for all x . This process can easily be modeled with an MAP (Λ, λ, q) : take

$$\Lambda = \tilde{\Lambda}, \quad \lambda_{xy} = \begin{cases} \mu_x & \text{if } x = y, \\ \tilde{\lambda}_{xy} & \text{otherwise} \end{cases} \quad \text{and} \quad q_{xy} = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

The MMPP is often used, both theoretically and practically, as it is easy to implement. However, models like the $M|M|1$ queue above can not be modeled with it. More details on the MMPP are given in Asmussen & Koole [3].

It can be shown that the class of MAPs is dense in the class of all arrival processes. This is shown in appendix A. The approximating MAPs used there have bounded rates in each state, i.e. $\sum_y \lambda_{xy} \leq \gamma$ for all x for some constant γ . By adding transitions from x to x with $q_{xx} = 0$ we can modify the MAP such that $\sum_y \lambda_{xy} = \gamma$ in each x . This is assumed throughout.

1.2. Symmetric customer assignment model

Now consider the following model. Customers arrive according to an MAP to a system consisting of m parallel queues. On arrival the customers have to be assigned to one of the queues. This assignment may depend on the state of the MAP reached at the arrival instant, and on the previous queue lengths. Queue j has a buffer of size B_j , including the customer being served. We write $B = (B_1, \dots, B_m)$. It is not allowed to assign a customer to a full queue, unless all queues are full. Each queue has a server which serves with rate μ . Our goal is to show that each arriving customer should be assigned to the shortest non-full queue, for various objective functions.

The total transition rate out of each state is bounded by $\gamma + m\mu$. Now the system can be seen to operate as follows. The time between two transitions is exponentially distributed with parameter $\alpha \geq \gamma + m\mu$. The transitions at the jump times have probabilities proportional to their rates. Central in our approach is the analysis of the Markov chain on the jump times, the embedded Markov chain. This method is called *uniformization*. (For more details, see chapter 5.)

At a jump, the probability of a transition from x to y in the arrival process is λ_{xy}/α . The probability of a departure at a queue is μ/α , the arrival probabilities remain q_{xy} . For notational simplicity we assume $\alpha = 1$, i.e. we use the same variables for the embedded discrete-time model as for the original continuous-time model. Note that a transition in the MAP and a departure at one of the queues cannot happen simultaneously. The state of our model will be notated as (x, i) , with $x \in \Lambda$ the state of the MAP and $i = (i_1, \dots, i_m)$ the state of the queues, i_j being the number of customers in queue j . Then, at each decision epoch, with probability λ_{xy} the arrival process moves from x to

y , giving an arrival with probability q_{xy} , and there is a (potential) departure of a customer at each queue with probability μ . With probability $1 - \gamma - m\mu$ a dummy transition occurs. Now define $v_{(x,i)}^n$ as the expected costs over n jumps of the embedded Markov chain, starting in state (x, i) . The $v_{(x,i)}^n$ can be computed recursively, using the following dynamic programming (dp) equation:

$$v_{(x,i)}^{n+1} = \sum_y \lambda_{xy} \left(q_{xy} \min_j \{ v_{(y,i+e_j)}^n \} + (1 - q_{xy}) v_{(y,i)}^n \right) + \sum_{j=1}^m \mu v_{(x,(i-e_j)+)}^n + (1 - \gamma - m\mu) v_{(x,i)}^n. \quad (1.2.1)$$

The minimization ranges over all j for which the queues are not full, i.e. for which $i_j < B_j$. If $i = B$, add action 0 with $e_0 = 0$.

Note that there are no immediate costs. The only costs are the v^0 , meaning that there are costs associated with the state reached in the end. Omitting the immediate costs does not restrict generality, but makes the analysis more elegant. Also note that relation (1.2.1) is not in the standard dynamic programming form because the action taken may depend on the current state of the arrival process y and not just on x . (In chapter 5 it is rewritten to bring it in the standard form.) The following lemma gives relations between the expected minimal costs in different states of the model.

1.2.1. Lemma. *If*

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \text{ for } i_{j_1} \leq i_{j_2}, \quad i + e_{j_1} + e_{j_2} \leq B, \quad (1.2.2)$$

$$w_{(x,i)} \leq w_{(x,i+e_{j_1})} \text{ for } i + e_{j_1} \leq B \quad (1.2.3)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \quad \text{for } i^* \text{ a permutation of } i, \quad i^* \leq B \quad (1.2.4)$$

hold for the cost function $w = v^0$, then they hold for all v^n .

For the proof we refer to the proof of corollary 3.3.2, because the model studied here is a special case of the model studied in section 3.3.

Note that the lemma gives conditions on the v^0 , the cost function. Let us interpret the equations. Equation (1.2.2) gives the optimal policy. If we have to decide between assigning a customer to queue j_1 or to j_2 we have to choose j_1 if there are less customers in that queue. Thus amongst the non-full queues, the shortest is selected. This policy is called the *Shortest Queue Policy* (SQP). The SQP tries to balance the number of customers in the queues.

Equation (1.2.3) and (1.2.4) are needed to prove (1.2.2). The former gives a general objective: fewer customers is better. The latter shows that the value function is symmetric, even though the buffer sizes can be different.

We assume that the costs are bounded, either from above or below. This ensures that, in the continuous-time model, the costs at time T , for all T , are

well defined. By (1.2.3), the costs are bounded from below by $v_{(x,0)}^n$ for fixed x , meaning that the assumption is not very restrictive. We assume it throughout this chapter. Now we can prove that the SQP minimizes the costs at time T , using corollary 5.2.2.

1.2.2. Theorem. *For all T , the SQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (1.2.2) to (1.2.4).*

It remains to study the cost functions that satisfy the conditions. An obvious cost function is $v_{(x,i)}^0 = i_1 + \dots + i_m = |i|$, meaning that the SQP minimizes the total number of customers in the system in expectation, both at the time horizon T and from 0 to T . Another cost function that satisfies the conditions is $v_{(x,i)}^0 = \max_j \{i_j\}$, the maximum queue length. Note that the dependence on the state of the arrival process can be quite general: if we associate costs c_x with state x , cost functions like $v_{(x,i)}^0 = c_x + |i|$ and $v_{(x,i)}^0 = c_x \max_j \{i_j\}$ for $c_x \geq 0$ are allowed. In fact, a necessary and sufficient condition is that for x fixed the costs must be weak Schur convex in i , as shown in appendix C. Not only $v_{(x,i)}^0 = |i|$ but also $v_{(x,i)}^0 = \mathbf{I}\{|i| > s\}$ is allowed for all s . ($\mathbf{I}\{\cdot\}$ is the indicator function.) This means that the SQP minimizes the probability that there are more than s customers in the system at T , i.e. the SQP *stochastically* minimizes the number of customers in the system. It is easy to see that if $v_{(x,i)}^0 = c_{(x,i)}$ satisfies (1.2.2) to (1.2.4), so does $v_{(x,i)}^0 = \mathbf{I}\{c_{(x,i)} > s\}$. This means that each cost function which is minimized by the SQP in expectation at T is minimized stochastically too. Summarizing, the SQP minimizes all Schur convex functions stochastically.

The first to prove the optimality of the SQP for minimizing the number of customers in the system, was Winston [82], in 1977. He assumed Poisson arrivals and infinite buffers. Weber [75] extended this to arbitrary arrivals, but his argument for service time distributions with an increasing failure rate was shown to be false in Sparaggis & Towsley [68]. Whitt [81] showed that the SQP is not optimal for a model with U-shaped failure rates. Proposition 8.3.2 of Walrand [74] gives a coupling proof for the exponential server case. Another proof of the pathwise optimality of the SQP is given in Hordijk & Koole [22]. We give yet another coupling proof based on dynamic programming in section 1.5. In Hordijk & Koole [21] finite buffers are introduced. There the number of departed customers is considered, rather than the number of customers in the system. Blocking is allowed. This model is discussed in section 1.6. The model of Towsley et al. [71] is exactly the model studied here. Johri [31] and recently Menich & Serfzo [46] weakened the conditions on the arrival and service rates. Similar conditions are studied in chapter 3. Finally, Sparaggis & Towsley [68] obtained the result for service times with an increasing likelihood ratio.

As said, in section 1.6 we consider a model in which the reward is related to the number of departed customers. Other customer assignment models can be found in section 1.3 to 1.9. In chapter 2 we generalize the present result to a model in which a certain dependency of the arrival process on the state of the queues is allowed. In chapter 3 we study models with different service rates for

different queues. There we give a partial characterization of the optimal policy. Together with this, assumptions on arrival and service rates are generalized.

1.3. Customer assignment model without waiting room

When we drop the condition that the service rates must be equal in each queue, we get an interesting problem. Numerical computations indicate that there is no optimal policy with a nice structure; for example the optimal policy depends, in the case of Poisson arrivals, on the arrival rate. In chapter 3 we give a partial characterization of the optimal policy using dynamic programming, and there we go into more details on the numerical results obtained by various researchers. Here we consider a special case where the optimal policy can be completely described, namely the case where there is, besides the customer in service, no space in the queues, i.e. $B = (1, \dots, 1)$. Queue j has a server with service rate μ_j , and we take $\mu_1 \geq \dots \geq \mu_m$ for convenience. We show that for various cost functions it is optimal to assign each arriving customer to the fastest available server. We call this policy the *Fastest Queue Policy* (FQP).

The first to address this problem was Seth, whose paper [64] appeared in the same year as Winston's seminal paper on the SQP [82], 1977. He analyzed the model with $m = 2$ servers and Poisson arrivals. Then there are only two policies to be considered, for which the stationary distribution is easily computed. The FQP minimizes the blocking probability. Derman et al. [17] generalize this result to multiple servers and general arrivals. Recent results for this type of model are discussed in section 3.4, where we consider a similar model with class-dependent blocking costs.

Seth [64] also gives a counterexample to the optimality of the FQP for non-exponential service times. A similar result is obtained by Cooper & Palakurthi [14]. These results show the sensitivity of this model to the shape of the service time distributions.

Now we derive the optimality of the FQP. As in the previous section, the model is uniformizable. Assume $\gamma + \mu_1 + \dots + \mu_m \leq 1$. The dynamic programming formulation is:

$$v_{(x,i)}^{n+1} = \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y,i+e_j)}^n\} + (1 - q_{xy}) v_{(y,i)}^n \right) + \sum_{j=1}^m \mu_j v_{(x,(i-e_j)^+)}^n + (1 - \gamma - \mu_1 - \dots - \mu_m) v_{(x,i)}^n. \quad (1.3.1)$$

The minimization ranges over all queues for which $i_j = 0$. Note the similarity with (1.2.1).

The following lemma gives the optimality of the FQP.

1.3.1. Lemma. If

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \text{ for } i_{j_1} = i_{j_2} = 0, j_1 < j_2 \quad (1.3.2)$$

and

$$w_{(x,i)} \leq w_{(x,i+e_{j_1})} \text{ for } i_{j_1} = 0 \quad (1.3.3)$$

hold for the cost function $w = v^0$, then they hold for all v^n .

Equation (1.3.2) gives the optimality of the FQP. For the proof we refer to the proof of the equivalent lemma for the more general model studied in section 3.4. There it is shown how the optimality of the FQP follows from a more general result on an asymmetric customer assignment problem. Also the symmetric model of section 1.2 is a special case of that model.

1.3.2. Theorem. *The FQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (1.3.2) and (1.3.3).*

Let us consider the cost functions satisfying the conditions. As in the previous section $v_{(x,i)}^0 = i_1 + \dots + i_m = |i|$ is allowed. Again, each allowable cost function is also minimized stochastically. This gives us, if we take $v_{(x,i)}^0 = \mathbf{I}\{|i| \geq m\}$, that the FQP minimizes the blocking probability at each T .

For the SQP we had a complete characterization of all allowable cost functions. Here something similar holds: the allowable cost functions are the set of functions increasing in an ordering, which is called the *partial sum ordering* in Chang et al. [12]. In appendix C the ordering is introduced and the equivalence is shown.

1.4. Discrete-time customer assignment model

So far we studied a continuous-time model by analyzing a discrete-time one. Of course, discrete-time models themselves are also interesting. Unfortunately, the model of lemma 1.2.1 is not very realistic in discrete time: arrivals and departures cannot happen simultaneously and therefore they are not independent. The optimality result for this model is more involved than for the model without simultaneous events. Therefore we analyze the following simple model. There are 2 identical parallel queues with infinite capacity, each with one server. When a customer is served during a time slot it leaves the system with probability μ , giving geometric service times with average $1/\mu$. The interarrival times are geometric with parameter λ . The state is denoted by (i, j) , with i (j) the number of customers in queue 1 (2).

The dynamic programming equation becomes:

$$\begin{aligned} v_{(i,j)}^{n+1} = & \lambda \min \left\{ \mu^2 v_{((i-1)+, (j-1)+)}^n + \mu(1-\mu) v_{((i-1)+, j)}^n + \right. \\ & (1-\mu) \mu v_{(i+1, (j-1)+)}^n + (1-\mu)^2 v_{(i+1, j)}^n, \\ & \mu^2 v_{(i-1)+, (j-1)+}^n + \mu(1-\mu) v_{(i-1)+, j+1}^n + \\ & \left. (1-\mu) \mu v_{(i, (j-1)+1)}^n + (1-\mu)^2 v_{(i, j+1)}^n \right\} + \\ & (1-\lambda) \left(\mu^2 v_{((i-1)+, (j-1)+)}^n + \mu(1-\mu) v_{((i-1)+, j)}^n + \right. \\ & \left. (1-\mu) \mu v_{(i, (j-1)+)}^n + (1-\mu)^2 v_{(i, j)}^n \right). \end{aligned} \quad (1.4.1)$$

Note that if a queue is empty there is no departure even if an arrival occurs at that queue. We have the same equations as in the model without simultaneous events:

1.4.1. Lemma. *If*

$$w_{(i+1,j)} \leq w_{(i,j+1)} \text{ for } i \leq j, \quad (1.4.2)$$

$$w_{(i,j)} \leq w_{(i+1,j)}, \quad (1.4.3)$$

and

$$w_{(i,j)} = w_{(j,i)} \quad (1.4.4)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof can be found in chapter 4. The optimal policy is not immediately clear from the equations. In the proof however it is shown that for $i \leq j$ we have $\mu^2 v_{((i-1)^+, (j-1)^+)}^n + \mu(1-\mu)v_{((i-1)^+, +1, j)}^n + (1-\mu)\mu v_{(i+1, (j-1)^+)}^n + (1-\mu)^2 v_{(i+1, j)}^n \leq \mu^2 v_{((i-1)^+, (j-1)^+, +1)}^n + \mu(1-\mu)v_{((i-1)^+, j+1)}^n + (1-\mu)\mu v_{(i, (j-1)^+, +1)}^n + (1-\mu)^2 v_{(i, j+1)}^n$, which are the terms in the minimization of the dynamic programming equation. Thus the SQP is optimal.

1.4.2. Theorem. *The SQP minimizes the costs at each n for all cost functions satisfying (1.4.2) to (1.4.4).*

The equations derived here are equivalent to (1.2.2) to (1.2.4), for $m = 2$. Thus the same cost functions are allowed here.

The generalization to more than 2 queues seems to be straightforward, although we did not check that in full detail. When we introduce buffers however, problems arise. For example, when some queues are full we have to specify the allowable actions and the actual point in time at which the arrival occurs; before or after the departure. After the departure seems from a modeling point of view the most interesting; this results in a model where we decide on the assignment after the departure of the customers. For example, if all queues are full, this means assigning to a queue where a departure occurs. We conjecture that also in this case the SQP is optimal. If the assignment occurs before the departures take place, then the SQP might not be optimal.

In the next section we return to the study of continuous-time models.

1.5. Pathwise optimality

In this section we want to prove the pathwise optimality of the SQP for the continuous-time model of section 1.2. There we showed that the SQP is stochastically optimal at T for all allowable cost functions. This is equivalent with saying that, for an arbitrary policy R , we can couple the realizations such that the costs at T are lower under the SQP. To prove pathwise optimality, we have to show that for coupled realizations the SQP has lower costs jointly across time. Again, we want to use dynamic programming for our result. However, in the dynamic programming recursion we compute expected costs: v^n are the expected costs after n transitions. We give a similar recursion with random variables.

In the previous sections it was sufficient to know the transition rates. Here however we need to know the stochastic behavior and specify the underlying probability spaces. In section 1.2 it is argued that our model is governed by two independent processes: one governing the jump times and one governing the transitions themselves. The jump process is the same as in section 1.2, we will not further specify it. The transitions are generated by independent uniformly distributed random variables. Assume the current state is (x, i) . Let U be the r.v. generating the transition at the current jump time. Let (j) be the index of the j th smallest component of i . If $i_{(j)} = i_{(j+1)}$, take $(j) < (j+1)$. For example, if $i = (2, 1, 0, 1)$, then $(1) = 3$, $(2) = 2$, $(3) = 4$ and $(4) = 1$. Note that $i_{(1)} \leq \dots \leq i_{(m)}$, the usual definition of $i_{(\cdot)}$. Assume that the states of the MAP are numbered. The system moves to $(y, i + e_j)$ if $U \in [\sum_{z < y} \lambda_{xz}, \sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy})$ and if action j was chosen in state (y, i) . The system moves to (y, i) if $U \in [\sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy}, \sum_{z \leq y} \lambda_{xz})$, and to $(x, (i - e_{(j)})^+)$ if $U \in [\gamma + (j-1)\mu, \gamma + j\mu)$. A dummy transition occurs if $U \in [\gamma + m\mu, 1]$. Note that the actual coupling can be found in the term on departures: in different states, departures at the j th longest queue in both models are coupled. Although the method of proof is different, this is the same coupling as in Walrand [74] and Hordijk & Koole [22].

Let U_n , $n \geq 1$, be i.i.d. random variables, uniformly distributed on $[0, 1]$. Choose random variables $V_{(x,i)}^0$, for all x and i on the same probability space, and define $V_{(x,i)}^n$, $n \geq 1$ by the following recursion:

$$V_{(x,i)}^{n+1} = \begin{cases} \min_j \{V_{(y,i+e_j)}^n\} & \text{if } U_{n+1} \in \left[\sum_{z < y} \lambda_{xz}, \sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy} \right), y \in \Lambda \\ V_{(y,i)}^n & \text{if } U_{n+1} \in \left[\sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy}, \sum_{z \leq y} \lambda_{xz} \right), y \in \Lambda \\ V_{(x,(i-e_{(j)})^+)}^n & \text{if } U_{n+1} \in [\gamma + (j-1)\mu, \gamma + j\mu), j = 1, \dots, m \\ V_{(x,i)}^n & \text{if } U_{n+1} \in [\gamma + m\mu, 1] \end{cases}$$

The allowable actions are the same as in section 1.2.

The minimization in the recursion is taken on each sample path. In general this minimum need not be attained by a unique action. In the next lemma we show that, in this case, it is attained by the same action in each state, namely that action that assigns to the shortest queue, which gives the optimality of the SQP for the recursion.

1.5.1. Lemma. *If*

$$W_{(x, i+e_{j_1})} \leq W_{(x, i+e_{j_2})} \quad \text{for } i_{j_1} \leq i_{j_2}, \quad i + e_{j_1} + e_{j_2} \leq B, \quad (1.5.1)$$

$$W_{(x, i)} \leq W_{(x, i+e_{j_1})} \quad \text{for } i + e_{j_1} \leq B, \quad (1.5.2)$$

and

$$W_{(x, i)} = W_{(x, i^*)} \quad \text{for } i^* \text{ a permutation of } i, \quad i^* \leq B \quad (1.5.3)$$

hold for the cost function $W = V^0$, then they hold for all V^n .

The proof of lemma 1.5.1 can be found in chapter 4. To understand the meaning of this lemma, condition on a realization of the jump times. Number the r.v.'s governing the transitions in reverse and condition also on them. Then the lemma tells us that the costs are minimized by the SQP. Note that the coupling is implicit in the recursion; for all policies the same U_n are used. Thus the lemma shows that the costs are lowest under the SQP for each realization. This gives of course the optimality at each T but also the optimality over the whole path.

1.5.2. Theorem. *The SQP minimizes the costs pathwise for all cost functions satisfying (1.5.1) to (1.5.3).*

The costs are allowed to be random variables. Apart from that the conditions are similar to the conditions of the previous sections.

Note that from the pathwise optimality it also follows that the sum of the waiting times of the first n customers is minimized stochastically by the SQP.

1.6. Customer assignment model with rejection

Here we study a model which is similar to that of section 1.2. However, the policies and the type of cost functions studied are different. Concerning the policies, it is allowed to send a customer to a full queue, meaning that it is rejected. By introducing an extra queue without waiting room in the buffer, we can add a rejection option in each state. The type of cost functions studied here is concerned with the number of customers that have already departed. This is the model studied in Hordijk & Koole [21], but we choose to prove it a little differently. In view of the objective it would be appropriate to have a model with rewards, but in order to agree with the other models we study costs. We add an extra variable to the state space (x, i) , which counts the number of departed customers, i.e. if a departure occurs at queue j the system moves from (x, i, k) to $(x, i - e_j, k + 1)$. The dynamic programming equation is

$$\begin{aligned} v_{(x,i,k)}^{n+1} = & \sum_y \lambda_{xy} \left(q_{xy} \min_j \{ v_{(y,i+e_j \wedge B,k)}^n \} + (1 - q_{xy}) v_{(y,i,k)}^n \right) + \\ & \mu \sum_{j=1}^m \left(\delta_{i_j} v_{(x,i-e_j,k+1)}^n + (1 - \delta_{i_j}) v_{(x,i,k)}^n \right) + (1 - \gamma - m\mu) v_{(x,i,k)}^n. \end{aligned} \quad (1.6.1)$$

Because of the rejection option the minimization ranges over all j . Of course, instead of adding the variable k , we could have taken immediate costs. This however would only have given results in expectation instead of stochastic results.

The analysis continues as usual:

1.6.1. Lemma. If

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i+e_{j_2},k)} \quad \text{for } i_{j_1} \leq i_{j_2}, \quad i + e_{j_1} + e_{j_2} \leq B, \quad (1.6.2)$$

$$w_{(x,i,k+1)} \leq w_{(x,i+e_{j_1},k)} \quad \text{for } i + e_{j_1} \leq B, \quad (1.6.3)$$

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i,k)} \quad \text{for } i + e_{j_1} \leq B \quad (1.6.4)$$

and

$$w_{(x,i,k)} = w_{(x,i^*,k)} \quad \text{for } i^* \text{ a permutation of } i, \quad i^* \leq B \quad (1.6.5)$$

hold for the cost function v^0 , then they hold for all v^n .

The present model is a special case of the model of section 3.5. Thus for the proof of the lemma we refer to the derivation in the beginning of that section. Equation (1.6.2) is by now well known; we should assign to the shortest queue. Equation (1.6.3) states that the costs are smaller when customers leave quickly. Equation (1.6.4) says that a full system is better. Note that it is the reverse of (1.2.3); it allows us to include rejection as an action without losing the optimality of the SQP. Equation (1.6.5) is again symmetry.

Of course all cost functions satisfying (1.6.2) to (1.6.5) are allowed, but cost functions depending only on i are not of interest here, because (1.6.3) and

(1.6.4) would give that the costs are constant in each state. Of interest here is $v_{(x,i,k)}^0 = -k$, meaning that, when starting in $(x, i, 0)$, the SQP maximizes the expected number of departed customers. Also $\mathbf{I}\{k \leq s\}$ is allowable for all s , giving the following theorem.

1.6.2. Theorem. *The SQP maximizes the number of departed customers between 0 and T stochastically.*

1.7. Series of parallel processors

A type of model related to the symmetric customer assignment model is the following, introduced by Katehakis & Melolidakis [33]. We have a series of m groups of components, group j consisting of B_j components. The system is up when at least one component in each group is functioning. New components arrive according to an MAP. The problem is how to assign the arriving components to the groups. Assigning a component to a group in which all components are functioning means that the component is lost. First we study a model in which all components are subject to failure, all with the same intensity. This is the model studied by Katehakis & Melolidakis. Then we consider the case where only the working components can fail.

For the first model we assume that B_j is finite for each j . Let $\gamma + (B_1 + \dots + B_m)\mu \leq 1$. The dynamic programming equation is:

$$v_{(x,i)}^{n+1} = \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y,i+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y,i)}^n \right) + \mu \sum_{j=1}^m i_j v_{(x,i-e_j)}^n + (1 - \gamma - (i_1 + \dots + i_m)\mu)v_{(x,i)}^n. \quad (1.7.1)$$

As in the last section the minimization ranges over all j .

1.7.1. Lemma. *If*

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \quad \text{for } i_{j_1} \leq i_{j_2}, \quad i + e_{j_1} + e_{j_2} \leq B, \quad (1.7.2)$$

$$w_{(x,i+e_{j_1})} \leq w_{(x,i)} \quad \text{for } i + e_{j_1} \leq B \quad (1.7.3)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \quad \text{for } i^* \text{ a permutation of } i, \quad i^* \leq B \quad (1.7.4)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof can be found in chapter 4. It is interesting to note that for the proof of (1.7.2) we do not need (1.7.3), because of the fact that each component is handled in exactly the same way. Therefore we need (1.7.3) only to see that the optimal policy does not reject arriving components. If sending a component to a full group were not allowed, as in section 1.2, we could omit (1.7.3). In lemma 1.2.1 this cannot be done, as we need (1.2.3) in the proof of (1.2.2).

Equation (1.7.3) is the reverse of (1.2.3), and is again due to the service mechanism.

1.7.2. Theorem. *The SQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (1.7.2) to (1.7.4).*

One function we are interested in is $v_{(x,i)}^0 = \mathbf{I}\{\exists j \text{ with } i_j = 0\}$. This cost function is indeed allowable, giving that the SQP minimizes the probability that the system is down. If the system is up when there are k out of n groups functioning, rather than all n groups, we can take $v_{(x,i)}^0 = 1$ if there are more than k non-empty groups in i , and 0 otherwise. This cost function is also allowable, thus the SQP maximizes also in this k -out-of- n system the probability that the system is working. A related cost function is $\mathbf{I}\{\exists j \text{ with } i_j < k\}$. This is also an allowable choice, corresponding to a system in which each group much have at least k working components. These results were also obtained by Katehakis & Melolidakis [33].

Now we consider a similar model, not studied in [33], in which only the m components required for the system to function can fail. This means that no component fails if the system is down. If we want to maximize the probability that the system is up at T , the SQP might not be optimal, as the following example shows. Take $m = 2$, $\mu = \lambda = 1$ and $T = 2$. With the computational method described in appendix D, which amounts to computing the dynamic programming equations for a large uniformization parameter, we computed the optimal policy. It followed that it is optimal in state $(0, 1)$ to assign new components to group 2. Customers arriving after 0.967 are assigned to group 1.

However, if we look at the expected time the system is up from 0 to T the SQP is optimal. To show this, we have to introduce immediate costs. We prefer to incur all costs together at T , in a way similar to the model of the previous section. Therefore we add an extra component to the state space, which is raised by 1 each time a component fails. The dynamic programming equation is:

$$v_{(x,i,k)}^{n+1} = \begin{cases} \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y,i+e_j \wedge B,k)}^n\} + (1 - q_{xy}) v_{(y,i,k)}^n \right) + \\ \mu \sum_{j=1}^m v_{(x,i-e_j,k+1)}^n + (1 - \gamma - m\mu) v_{(x,i,k)}^n & \text{if } i_j > 0 \text{ for all } j, \\ \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y,i+e_j \wedge B,k)}^n\} + (1 - q_{xy}) v_{(y,i,k)}^n \right) + \\ (1 - \gamma) v_{(x,i,k)}^n & \text{if } i_j = 0 \text{ for some } j. \end{cases}$$

Again, the minimization ranges over all j .

1.7.3. Lemma. *If*

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i+e_{j_2},k)} \quad \text{for } i_{j_1} \leq i_{j_2}, \quad i + e_{j_1} + e_{j_2} \leq B, \quad (1.7.5)$$

$$\sum_{j_1=1}^m w_{(x,i-e_{j_1},k+1)} \leq m w_{(x,i,k)} \quad \text{for } i \geq e, \quad (1.7.6)$$

$$w_{(x, i+e_{j_1}, k)} \leq w_{(x, i, k)} \quad \text{for } i + e_{j_1} \leq B, \quad (1.7.7)$$

$$w_{(x, i, k+1)} \leq w_{(x, i, k)} \quad (1.7.8)$$

and

$$w_{(x, i, k)} = w_{(x, i^*, k)} \quad \text{for } i^* \text{ a permutation of } i, i^* \leq B \quad (1.7.9)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof can be found in chapter 4. As in the first model of this section, we only need (1.7.7) to know that we should not use the rejection option. As in the result of the previous section, we can take $v_{(x, i, k)}^0 = -k$, giving that the SQP maximizes the expected number of failed components. However, we are interested in the time the system is up. But, components only fail if the system is up, with rate $m\mu$. Thus the policy that maximizes the number of departures, also maximizes the time that the system is up.

1.7.4. Theorem. *The SQP maximizes the expected time that the system is up between 0 and T .*

As in section 1.6, the second and third equation give, for cost functions only depending on i , constant costs. Thus $v_{(x, i, k)}^0 = -k$ is the only cost function of interest.

In Koole [39] the same model is studied, but there the queue to which an arrival is assigned is determined at the time of the previous arrival. This models the repair at the spot by a repairman, and results in a model with a specific form of delayed information. Similar results as for the current model are derived.

1.8. Customer assignment model with workloads

The information available to the controller in the model of section 1.2 are the numbers of customers in the queues. Here we study a model in which the amount of work in the queues, the workload, is known. The characteristics of the model are as follows. The service times of all customers are identically independently distributed, the controller assigns not knowing the actual service times, and the servers all work at the same constant speed c . Daley [15] showed that a variant of the SQP, the *Shortest Workload Policy* (SWP), minimizes the total workload at each T . In fact, he shows with forward induction that the workload under the SWP is weakly submajorized by the workload of each policy, giving the stochastic optimality for each Schur convex cost function. (Appendix C deals with majorization.) Foss [18] obtains the same result. Also Wolff [84] shows that the SWP minimizes the workload, although he only compares the SWP with policies that are not allowed to depend on the workload.

We also prove the optimality of the SWP, again with dynamic programming. However, as decision points we do not take the jumps of a Poisson

process but the actual arrival instants. Thus, technically speaking, we condition on the arrival process. We do this to avoid technical problems: when the sojourn time between two events is constant, the amount of work done in each queue is also a constant, thus simplifying the analysis. In general, many models with arrivals according to MAP's can also be handled by taking arrivals at deterministic times. Exceptions are the second model of the previous section and several models of chapter 3, the reason being that even for arrivals according to MAP's the optimal policies are not myopic. We chose to use the MAP as much as possible, to link on with the forthcoming chapters.

Now we prove the optimality of the SWP at T . Let s_n be the sojourn time between the n th and $(n+1)$ th arrival, counted backward from the time horizon, let the amount of work done by a busy server in this time be $u_n = cs_n$, and assume that P is the distribution function of the service times. With i we denote the vector of workloads, $i \in \mathbb{R}_+^m$. We have

$$v_i^{n+1} = \min_j \left\{ \int_0^\infty v_{(i+te_j - u_n e)^+}^n dP(t) \right\} \quad (1.8.1)$$

1.8.1. Lemma. *If*

$$\int w_{i+te_{j_1}} dP(t) \leq \int w_{i+te_{j_2}} dP(t) \text{ for } i_{j_1} \leq i_{j_2}, \quad (1.8.2)$$

$$w_i \leq w_{i+te_{j_1}} \text{ for } t \geq 0, \quad (1.8.3)$$

and

$$w_i = w_{i^*} \text{ for } i^* \text{ a permutation of } i \quad (1.8.4)$$

hold for the cost function $w = v^0$, then they hold for all v^n .

Note the resemblance to lemma 1.2.1. The proof can be found in chapter 4. In section 3.3 we give a different proof of the optimality of the SWP; there we see it as the limiting case of the SQP model with batch arrivals.

Equation (1.8.2) without the integration, i.e. $w_{i+te_{j_1}} \leq w_{i+te_{j_2}}$ for all t , is not true; this means that it is essential that the controller does not know the actual service times of the arriving customers. To construct an example illustrating this, take $m = 2$, $u_0 = 2$ and $v_{(i_1, i_2)}^0 = i_1 + i_2$, which indeed satisfies the conditions of lemma 1.8.1. Let the service time be equal to 2 a.s. Then it is easily seen that, if we take $i = (0, 1)$, $t = 1$, $j_1 = 1$ and $j_2 = 2$, then $v_{i+te_{j_1}}^1 = v_{(1,1)}^1 = 1 > 0 = v_{(0,2)}^1 = v_{i+te_{j_2}}^1$.

1.8.2. Theorem. *The SWP minimizes the costs (stochastically) at T for all cost functions satisfying (1.8.2) to (1.8.4).*

The cost functions considered here are functions of \mathbb{R}_+^m . It follows directly that again all Schur convex functions satisfy the inequalities. See appendix C for an overview of these functions. If we require the inequalities to hold for all service time distributions P , then the Schur convex functions are exactly the

allowable cost functions, which can be shown in the same way as theorem C.1. Note that the statement in the penultimate paragraph of p. 304 in Daley [15], on the functions that respect weak majorization, is not correct: for example indicator functions of allowable cost functions are in general not convex.

For the SQP we were able to prove pathwise optimality. Here however, as stated in Wolff [84], we have the striking result that the SWP minimizes the total workload stochastically but not pathwise. To construct a counterexample to the pathwise optimality, take a model with initial workload $i = (1, 2)$ and speed $c = 1$. For the service time B we have $\mathbb{P}(B = 1) = \mathbb{P}(B = 2) = \frac{1}{2}$. The first customer arrives at $t = 0$, the second at $t = 1$. No more arrivals occur before $t = 4$. When we fix the policy used, there are four different realizations up to $t = 3$, each with probability $\frac{1}{4}$. To get a pathwise ordering, we have to combine the realizations for the SWP and an arbitrary policy R such that the SWP is better for all t . Take R such that we start with assigning to the longest queue, but the second customer is assigned to the shortest. Denote with b_i (\hat{b}_i) the service time of the i th arriving customer in the model that uses the SWP (R). At $t = 1$ the amount of work is $1 + b_1 + b_2$ ($1 + \hat{b}_1 + \hat{b}_2$). Therefore we have to couple $b_1 = b_2 = 1$ with $\hat{b}_1 = \hat{b}_2 = 1$. Now we show that if $\hat{b}_1 = 1$ and $\hat{b}_2 = 2$, then there is no choice of b_1 and b_2 which is pathwise better. Take first $b_1 = 1$ and $b_2 = 2$. Then, at $t = 3$, the system ruled by R is empty, but not the model under the SWP. For both eventualities with $b_1 = 2$ we have that the amount of work just after the first arrival is larger under the SWP.

Note that if we are allowed to let the coupling depend on t in this example, we find the optimality of the SWP. This is equivalent to saying that the SWP is stochastically optimal in this example, which follows also from theorem 1.8.2.

In the models of the next chapter where customers move through a network, it is of interest to consider the number of departed customers instead of the workloads. This model was studied by Wolff [83]. First he remarks that the SWP is stochastically equivalent to a single $M|M|m$ queue with FCFS discipline. Then he shows that FCFS is better than any policy in the model with parallel queues, using a coupling argument. In the coupling argument service times are given to the customers the moment they start service. This means that the controller is allowed to assign knowing the number of customers in each queue, and the remaining service times of the customers presently in service. A policy in this class is the SQP, but not the SWP or other policies depending on the workloads. This result is generalized to the class of all policies which do not depend on the service time of the arriving customer in Koole [36]. These results are all pathwise.

1.9. Customer assignment model without information

In the previous section we have shown that the SWP minimizes the total amount of work in the system stochastically, at any T . For exponential service times, we have seen in section 1.2 that the number of customers is minimized by the SQP. In the latter model the workloads are not known to the controller, i.e. in the model where the SQP is optimal, the *queue-length model*, the controller has to decide based on different information than in the *workload model* where the SWP is optimal. Note that because the SQP minimizes the number of customers stochastically it also stochastically minimizes the amount of work still to be done in the class of allowable policies. An interesting question is if either the SWP or the SQP is better with respect to minimizing the number of customers in the system. This question is answered by Wolff [83]. As mentioned in the previous section, he shows that the SWP is better than all policies that do not depend on the workload, amongst which is the SQP.

Besides the number of customers or the workload we have two more obvious models with a different amount of information. The first is where you have no information at all. For exponential service times and an initially empty system we show at the end of this section that each arriving customer should be assigned to each queue with probability $\frac{1}{m}$ to minimize the number of customers, and thus the total workload. We call this policy the *Equal Splitting Policy* (ESP). When we know the previous assignments but not the state of the system the *Cyclic Assignment Policy* (CAP) minimizes the number of customers; proposition 8.3.4 of Walrand [74] has a simple proof for the case with exponential service times, a proof for IFR service times (see appendix B for a definition of IFR) can be found in Liu & Towsley [42].

From standard results in Markov Decision Theory, we know that even if the class of policies in the models depending on the queue lengths are allowed to depend on the whole history, the SQP remains optimal. This means that the workload under the SQP is smaller than under the CAP (and the ESP). It is clear that the ESP is worse than the CAP. Thus if we list the policies in increasing order of expected workload, we have: SWP, SQP, CAP, and ESP.

We end this section with showing that the ESP minimizes the number of customers in the system, when there is no information available. For results for cost function related to the workloads, we refer to Chang et al. [11] and Chang [10]. A full proof of the result is given in Koole [38]. As this proof is based on forward instead of backward recursion, we will only sketch it.

We confine ourselves to two queues. Consider first a single model, with assignment vector $(p, 1 - p)$ with $p \geq \frac{1}{2}$. Let $Q^p(n) = (Q_1^p(n), Q_2^p(n))$ be the queue lengths directly after the n th event (which can be an arrival or a (potential) departure from one of the queues), and initial state $Q^p(0)$. Define for all $i, j, s \in \mathbb{N}_0$

$$A(i, j, s) = \{(x, y) \in \mathbb{N}_0^2 \mid x \leq i, y \leq j, x + y \leq s\}.$$

Now let

$$P_n^p(i, j, s) = \mathbb{P}((Q_1^p(n), Q_2^p(n)) \in A(i, j, s)).$$

Take $P_0^p(i, j, s) = 1$ for all i, j, s , which corresponds to starting with an empty system. Then it can be proven, with forward induction, that

$$P_n^p(i + j, i + k, s) \geq P_n^p(i, i + j + k, s) \quad (1.9.1)$$

for all $i, j, k, s, n \geq 0$. In a way this shows that if queue 1 has a higher assignment probability than queue 2, then this will result in a stochastically larger queue length. This interpretation becomes clear if we take $k = 0$. Having shown (1.9.1), we can compare two systems with assignment probabilities $q \geq p \geq \frac{1}{2}$. Again with forward induction it can be shown that, for all $i, j, s, n \geq 0$,

$$P_n^p(i, i + j, s) \geq P_n^q(i, i + j, s).$$

For $i \geq s$ this states that the probability of having less than s customers in the system at any time is maximized by the ESP. In [38] it is shown how this result can be made pathwise, and how it can be generalized to an arbitrary number of queues.

Let us compare the method of proof for the above result with dynamic programming. In general, dp determines the optimal action in each state. In the current setting, due to the information structure, distributions on states would serve as states. Equation (1.9.1) shows that certain distributions do not occur, and in those that can occur it is advantageous to have a more balanced assignment.

If we were to apply dp to the model without state information (i.e., with distributions as states) then we would find the CAP as optimal policy. Although the CAP uses no state information, it uses the previous assignments to determine the current. Note that Bernoulli policies use no information at all.

1.10. MAP's with multiple customer classes and server vacations

In the models we study after this section, we have multiple customer classes and server vacations. Therefore we add a mark to each arrival generated by the MAP to model the class of an arriving customer or the availability of a server. Let q_{xy}^k be the probability of an arrival in class k , given a transition from x to y . Then an arrival with mark k , $1 \leq k \leq m$, denotes the arrival of a customer in class k . In some of our models servers can go on vacation at random times. There are s servers. With an arrival in class k , $m + 1 \leq k \leq m + s$, an event for server $k - m$ is meant; if the server is working he goes on vacation and vice versa. We assume $\sum_{k=1}^{s+m} q_{xy}^k \leq 1$. Simultaneous arrivals cannot occur. To give a complete description of the current state of the system we have to specify the state of the arrival process, of the servers and of the queues. Thus, besides the state of the arrival process x and the state of the queues i we have to add

a variable to the state of the system denoting the availability of the servers. Because we are interested in optimally assigning the available servers, but not in controlling the number of servers, it is convenient to make this variable part of the arrival process. Thus, add a vector $z = (z_1, \dots, z_s)$ of 0-1 variables to the state of the MAP. Server k is available if and only if $z_k = 1$. Concerning the arrivals of customers, we want to address questions like: when is the first time that the system becomes empty after N arrivals? To deal with this type of question, we also would like to identify the state of the arrival process with the numbers of arrived customers. To do so, also add a variable $n = (n_1, \dots, n_m)$ to the state of the MAP, where n_k is the number of customers that have arrived in class k . Assume we have an MAP (Λ, λ, q) . The transition intensities of the new arrival process $(\tilde{\Lambda}, \tilde{\lambda}, \tilde{q})$ with state space $\tilde{\Lambda} = \{(x, z, n)\}$ become:

$$\begin{aligned}\tilde{\lambda}_{(x,z,n)(y,z,n+e_k)} &= \lambda_{xy} q_{xy}^k, \quad 1 \leq k \leq m \\ \tilde{q}_{(x,z,n)(y,z,n+e_k)}^l &= \mathbf{I}\{l = k\}, \quad 1 \leq k \leq m \\ \tilde{\lambda}_{(x,z,n)(y,z^*,n)} &= \lambda_{xy} q_{xy}^{m+k}, \quad z_j = z_j^*, \quad j \neq k, \quad z_k^* = (1 - z_k)^+ \\ \tilde{q}_{(x,z,n)(y,z^*,n)}^{m+k} &= \begin{cases} 1 & \text{if } z_j = z_j^*, \quad j \neq k, \quad z_k^* = (1 - z_k)^+ \\ 0 & \text{otherwise} \end{cases} \\ \tilde{\lambda}_{(x,z,n)(y,z,n)} &= \lambda_{xy} \left(1 - \sum_{k=1}^{m+s} q_{xy}^k\right) \\ \tilde{q}_{(x,z,n)(y,z,n)}^k &= 0\end{aligned}$$

The arrival process just defined is again an MAP. Thus we have the following equivalent definition:

1.10.1. Definition. (Markov Arrival Process) Let Λ be the countable state space of a Markov process with transition intensities λ_{xy} with $x, y \in \Lambda$. When this process moves from x to y , with probability q_{xy}^k an arrival in class $1 \leq k \leq m$ occurs, and with probability q_{xy}^{m+k} an event with server $1 \leq k \leq s$ occurs. There are sets $\Lambda_1^s, \dots, \Lambda_s^s \subset \Lambda$ such that server k is available if and only if $x \in \Lambda_k^s$, and sets $\Lambda_{1n}^a, \dots, \Lambda_{mn}^a \subset \Lambda$, $n \in \mathbb{N}$, such that if $x \in \Lambda_{kn}^a$ then there have been n or more arrivals of class k . We call the triple (Λ, λ, q) an MAP.

Section 1.1 handled MAP's with only one customer class and without server vacations. We showed there how to model various types of arrival processes. If the arrival streams in different classes are independent of each other we can take the superposition of the m processes (i.e., the process with as state space the product space, in which each component is independent of the others), with the arrivals in process j having marks j . This is again an MAP.

The result in appendix A on the approximation of arrival processes is on marked arrival streams, thus the weak convergence of MAPs to general arrival processes holds for the present model too.

1.11. Server assignment model with a single server

In this section we study a model in which a single server is to be assigned to one of m customer classes. Each customer in class j has an exponential service time with intensity μ_j . At each decision epoch the server can be reassigned. Customers arrive in the m classes according to an MAP. Server vacations are not interesting because we have only one server, therefore we do not model them. This model has been studied extensively, mainly for linear costs, i.e. a cost function in which every customer of class j adds c_j to the costs. It is well known that the customers should be served in decreasing order of $\mu_j c_j$, according to the μc -rule. This result can be found in Baras et al. [4] and Buyukkoc et al. [8], the last paper using a very simple interchange argument. Here we also show that the μc -rule is optimal, using dynamic programming. The μc -rule minimizes the costs stochastically only in the special case that the service rates and the costs are both decreasing. An interesting related model is that of Righter & Shanthikumar [57]. They have DFR service time distributions and consider the number of successful departures. With p_j the probability that a departure in queue j is successful, they show that the μp -rule is optimal. This result holds stochastically, in all cases. Later on in this section we also consider DFR service times, showing that the μc -rule, with μ_j the current failure rate of a customer, is still optimal.

Take $\mu = \max_j \mu_j$. We uniformize, and we assume therefore that $\gamma + \mu \leq 1$. We consider two models, one in which idleness of the server is allowed and one in which it is not allowed. We have as the dynamic programming equation:

$$\begin{aligned} v_{(x,i)}^{n+1} = \min_l \left\{ \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y,i+e_j)}^n + (1 - \sum_{j=1}^m q_{xy}^j) v_{(y,i)}^n \right) + \right. \\ \left. \mu_l v_{(x,i-e_l)}^n + (1 - \gamma - \mu_l) v_{(x,i)}^n \right\} = \\ \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y,i+e_j)}^n + (1 - \sum_{j=1}^m q_{xy}^j) v_{(y,i)}^n \right) + \\ \min_l \left\{ \mu_l v_{(x,i-e_l)}^n + (\mu - \mu_l) v_{(x,i)}^n \right\} + (1 - \gamma - \mu) v_{(x,i)}^n. \end{aligned}$$

The minimization ranges over all l with $i_l > 0$. If idleness is allowed, action 0 (with $\mu_0 = 0$) has to be added to the actions. Now we have the following lemma:

1.11.1. Lemma. *If idleness is not allowed or is suboptimal in each state and*

$$\begin{aligned} \mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i)} \leq \mu_{j_2} w_{(x,i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x,i)} \quad (1.11.1) \\ \text{for } j_1 < j_2 \text{ and } i_{j_1}, i_{j_2} > 0 \end{aligned}$$

hold for the cost function v^0 , then they hold for all v^n .

Thus, if idleness is not allowed, we have the optimality of the policy that assigns to the non-empty queue with smallest index. We call this policy the *Smallest Index Policy* (SIP). Later on in this section we study a more general model. Therefore lemma 1.11.1 follows from lemma 1.11.5. When idleness is allowed we have to add monotonicity to obtain the suboptimality of idleness.

1.11.2. Lemma. *If*

$$w_{(x, i - e_{j_1})} \leq w_{(x, i)} \text{ for } i_{j_1} > 0 \quad (1.11.2)$$

holds for the cost function v^0 , then it holds for all v^n .

Note that (1.11.2) is a special case of (1.11.1), for the cases that $|i| \geq 2$, by giving action 0 the lowest priority. For the proof, we refer to the proof of lemma 1.11.6. We can have two separate lemmas because we do not need (1.11.2) in the proof of (1.11.1). The same approach does not work in most customer assignment models because monotonicity is needed to prove the inequality giving the structure of the optimal policy. The same holds for the multiple server model of the next section. We summarize our results for the continuous-time model.

1.11.3. Theorem. *The SIP minimizes the costs at T for all cost functions satisfying (1.11.1), when idleness is not allowed.*

1.11.4. Theorem. *The SIP minimizes the costs at T for all cost functions satisfying (1.11.1) and (1.11.2), when idleness is allowed.*

Remark. In section 1.2 we assumed that all cost functions are bounded. In the model of theorem 1.11.3 however, it is natural to consider cost functions of which both the positive and negative parts are unbounded. For example, if $m = 2$, $v_{(x, i)}^0 = i_1 - i_2$ is an allowable cost function. In this case finiteness of the costs at T can be shown when the costs are ν -bounded, as is proved in the first part of the proof of theorem 5.3.2.

Also in the case of an infinite planning horizon, there are complications. Due to the unboundedness of the costs we cannot use the results for negative dynamic programming, as suggested in chapter 5. In the case of Poisson arrivals ν -geometric recurrence can be shown, giving average and *Blackwell* optimality of the SIP. The ν -geometric recurrence of the discrete-time model is shown by Spieksma [70]. Her results are used in Dekker & Hordijk [16] to verify their conditions for Blackwell optimality in the continuous-time semi-Markov model.

Now we study the cost functions. In general we have the following characterization. Define $\Delta_j v_{(x, i)}^0 = v_{(x, i + e_j)}^0 - v_{(x, i)}^0$. Then (1.11.1) is equivalent to

$$\mu_{j_1} \Delta_{j_1} v_{(x, i - e_{j_1})}^0 \geq \mu_{j_2} \Delta_{j_2} v_{(x, i - e_{j_2})}^0 \text{ if } j_1 < j_2 \text{ and } i_{j_1}, i_{j_2} > 0 \quad (1.11.3)$$

and (1.11.2) to

$$\Delta_j v_{(x,i)}^0 \geq 0 \text{ for all } j. \quad (1.11.4)$$

A simple cost function satisfying both conditions is the following: $v_{(x,i)}^0 = \mathbf{I}\{|i| > 0\}$. As we minimize the expected cost for this function, we minimize the probability that there are any customers present at time n . When there are no arrivals, this coincides with minimizing the makespan stochastically. Of course this is of no interest here, as all work conserving policies minimize the makespan. The analysis of this type of cost function is of interest to the multiple server case.

A cost function of interest here is the following: $v_{(x,i)}^0 = \sum_{j=1}^m c_j i_j$. It is easy to see that this function satisfies (1.11.3) if and only if $\mu_1 c_1 \geq \dots \geq \mu_m c_m$. This means that the μc -rule minimizes the costs in expectation. If idleness is allowed we have to add $c_j \geq 0$ to make the cost function satisfy (1.11.4). In the customer assignment models studied previously in this chapter every cost function that was minimized in expectation was also minimized stochastically. Here this is not the case. To analyze the stochastic optimality, first assume $c_j \geq 0$. We distinguish three cases.

1. $\mu_1 \geq \dots \geq \mu_m$, $c_1 \geq \dots \geq c_m \geq 0$. Now $\mathbf{I}\{\sum_{j=1}^m c_j i_j > k\}$ satisfies the conditions too. Indeed, we have $v_{(x,i-e_{j_1})}^0 \leq v_{(x,i-e_{j_2})}^0 \leq v_{(x,i)}^0$. Therefore $\Delta_{j_1} v_{(x,i-e_{j_1})}^0 \geq \Delta_{j_2} v_{(x,i-e_{j_2})}^0$. Together with $\mu_{j_1} \geq \mu_{j_2}$ we have (1.11.3).
2. There are j_1 and j_2 such that $j_1 < j_2$ and $\mu_{j_1} < \mu_{j_2}$. For example, take $m = 2$, no arrivals, $i = (1, 1)$, $\mu_1 = 1$, $c_1 = 5$, $\mu_2 = 2$ and $c_2 = 2$. The μc -rule prescribes class 1, however, if we want to minimize $\mathbf{IP}(i_1 c_1 + i_2 c_2 \geq 6)$ for some T , we should start with class 2.
3. There are j_1 and j_2 such that $j_1 < j_2$ and $c_{j_1} < c_{j_2}$. For example, take $m = 2$, no arrivals, $i = (1, 1)$, $\mu_1 = 4$, $c_1 = 1$, $\mu_2 = 1$ and $c_2 = 3$. Again the μc -rule prescribes class 1, but we should choose class 2 to minimize $\mathbf{IP}(i_1 c_1 + i_2 c_2 \geq 2)$ for some T .

Thus the μc -rule is stochastically optimal only if $\mu_1 \geq \dots \geq \mu_m$ and $c_1 \geq \dots \geq c_m$. We call the service rates and costs in this case *agreeable*. When there are no arrivals, the stochastic optimality also follows from Righter & Shanthikumar [58], by taking, in their notation, $f_j(C_j) = c_j \mathbf{I}\{C_j > T\}$.

If we do not allow idleness, i.e. when the holding costs can be negative, the condition for stochastic optimality is as follows. Take m_1 such that $c_1 \geq \dots \geq c_{m_1} \geq 0 \geq c_{m_1+1} \geq \dots \geq c_m$. If $\mu_1 \geq \dots \geq \mu_{m_1}$ and if $\mu_{m_1+1} \leq \dots \leq \mu_m$, then $\mathbf{I}\{\sum_{j=1}^m c_j i_j > k\}$ satisfies (1.11.3).

Several other interesting cost functions, like the expected weighted number of late customers and the expected weighted sum of customer tardiness, also satisfy the conditions on the cost functions. See Chang et al. [12] for details.

We change the model as follows. When a customer in queue j is served it leaves the system with rate μ_j , and joins queue $f(j)$ with rate $\mu - \mu_j$. We assume that $f(j) \geq j - 1$. If $f(j) = j$ for all j we have the same model as above. This service mechanism can be formulated in terms of successful departures. If

$p_j = \mu_j/\mu$, then p_j is the probability that a departure is successful. The value function becomes

$$v_{(x,i)}^{n+1} = \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y,i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xy}^j\right) v_{(y,i)}^n \right) + \min_l \left\{ \mu_l v_{(x,i-e_l)}^n + (\mu - \mu_l) v_{(x,i-e_l+e_{f(l)})}^n \right\} + (1 - \gamma - \mu) v_{(x,i)}^n. \quad (1.11.5)$$

Again, the minimization ranges over all l with $i_l > 0$. If idleness is allowed, action 0 (with $\mu_0 = 0$, $f(0) = 0$ and $e_0 = 0$) has to be added to the actions.

1.11.5. Lemma. *If idleness is not allowed or not optimal in each state and*

$$\begin{aligned} \mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i-e_{j_1}+e_{f(j_1)})} &\leq \\ \mu_{j_2} w_{(x,i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x,i-e_{j_2}+e_{f(j_2)})} &\text{ for } j_1 < j_2 \text{ and } i_{j_1}, i_{j_2} > 0 \end{aligned} \quad (1.11.6)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof can be found in chapter 4. Similar results for monotonicity hold:

1.11.6. Lemma. *If*

$$\mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i-e_{j_1}+e_{f(j_1)})} \leq \mu w_{(x,i)} \text{ for } i_{j_1} > 0 \quad (1.11.7)$$

holds for the cost function v^0 , then it holds for all v^n .

The proof can be found in chapter 4. Again we have:

1.11.7. Theorem. *The SIP minimizes the costs at T for all cost functions satisfying (1.11.6), when idleness is not allowed. When idleness is allowed, (1.11.7) should be added.*

Similar results are obtained in section 3 of Nain [49]. Actually, he allows random routing of unsuccessfully served customers, but, as in the present model, only to higher numbered queues. We chose not to model random routing so as to keep the notation simple.

An interesting case we can model is that of a single class of DFR service times. We use the characterization of DFR distributions by phase-type distributions as shown in appendix B. There the transition intensity in each phase is taken to be equal. After k phases of service a customer finishes service with probability α_k or receives one or more additional phases of service with probability $1 - \alpha_k$. If a DFR distribution is approximated by phase-type distributions in this way, then the α_k are non-increasing. It does not restrict generality to take $\alpha_k = \alpha_l$ for $k > l$, with l a constant (in appendix B $l = m^2$).

Consider the following server assignment model. Take $m = l$, $p_j = \alpha_j$, $f(j) = j + 1$ if $j < m$, and $f(m) = m$. The costs are linear with $c_j = 1$ for

all j . This cost function satisfies (1.11.6), because the α_j are decreasing. Thus the expected number of customer in the system is minimized in expectation by serving the customer with the highest failure rate. Using the same argument as for the model without routing of customers, it follows that this result holds also stochastically. Using a limiting argument, this gives that the number of customers in an $G|DFR|1$ queue is minimized at T by the policy that serves the customer with the least attained service time (the LAST policy).

Note that customers are generally not served until they leave the system, but only until they change phase. For the limiting case this gives processor sharing as the service discipline for all customers who have received the same amount of service.

Taking $p_j = \alpha_{m-j}$, $f(j) = j - 1$ and $c_j = -1$ shows that MAST (most attained service time) maximizes the number of customers in the system. Although the p_k are increasing, also $v_{(x,i)}^0 = \mathbf{I}\{|i| \leq s\}$ satisfies the conditions, and thus the result holds also stochastically. Note that MAST is equivalent to FCFS.

For IFR service times it is shown in appendix B that the α_k are non-decreasing. In this case the above results are reversed.

1.11.8. Theorem. *LAST (FCFS) stochastically minimizes the number of customers at T in a $G|G|1$ queue in the case of DFR (IFR) service times; LAST (FCFS) stochastically maximizes the number of customers at T in a $G|G|1$ queue in the case of IFR (DFR) service times.*

All these results can also be found in Righter & Shanthikumar [57].

We continue with generalizing the above results to models with multiple customer classes. To avoid certain technicalities we assume that each class has either positive holding costs and a DFR service time distribution, or negative holding costs and a IFR service time distribution. From the construction it will be clear how to deal with the other two cases; in these cases however the condition that $f(j) \geq j - 1$ can easily be violated.

Thus assume first that each class has its own DFR service times, class n having l_n phases, $n = 1, \dots, r$, r being the number of classes. The success probability of phase k of class n is α_k^n , the holding costs are $c_n^* > 0$, independent of the phase. We make a distinction between classes and queues. Now take for each class and possible phase a queue, i.e. $l_1 + \dots + l_r$ queues, with j_{nk} the number of customers in the queue corresponding to the n th customer class and the k th phase. Of course, we take $f(j_{nk}) = j_{nk+1}$ if $k < l_n$ and $f(j_{nl_n}) = j_{nl_n}$, $p_{nk} = \alpha_k^n$, and $c_{nk} = c_n^*$. Order the queues in decreasing value of $p_{nk}c_{nk}$. Then the μc -rule is optimal. We use a limiting argument to get the result for the case of general DFR service times. The optimal policy serves the customer with the highest product of holding cost and failure rate. We call this policy again the μc -rule.

As indicated, it is also possible to have customer classes with IFR service time distributions and negative holding costs (requiring that idling is not allowed). In this case all DFR customers are first served, possibly using processor

sharing. Then (as long as there are no arrivals) the customers with IFR service times are served.

1.11.9. Theorem. *The μc -rule minimizes the expected holding costs at T if customers have either DFR or IFR service time distributions, provided the holding costs for customers with DFR (IFR) service time distributions are positive (negative). Idleness is allowed if there are no customers with IFR service times.*

For a stochastic result we need that $\alpha_{k_1}^{n_1} \geq \alpha_{k_2}^{n_2}$ for all k_1 and k_2 if $c_{n_1} > c_{n_2}$. In the limiting case this means that the failure rate of class n_1 is always higher than the failure rate of class n_2 . This is the case if we have a family of random processing times with decreasing failure rate (see for example section 4.2 of Weiss [79]).

The results of this section are a superset of those in Koole [37]: there it is assumed that $f(j) \geq j$ instead of $f(j) \geq j - 1$.

Remark. Equation (1.11.5) can be written as $v_{(x,i)}^{n+1} = \gamma T_1 v_{(x,i)}^n + \mu T_2 v_{(x,i)}^n$, with $T_1 v_{(x,i)}^n = (\sum_y \lambda_{xy}(\dots))/\gamma$ and $T_2 v_{(x,i)}^n = (\min_l \{\dots\})/\mu$, if we assume that $\gamma + \mu = 1$. Here T_1 and T_2 themselves can be seen as dp operators. In chapter 5 it is shown how convex combinations of dp operators result from a continuous time model.

A discrete time model however would typically consist of a departure and an arrival event in succession, resulting in a dp equation of the form $w_{(x,i)}^{n+1} = T_2 T_1 w_{(x,i)}^n$.

The proof of the lemmas 1.11.5 and 1.11.6 basically consists of showing that the equations propagate for T_1 and T_2 . Of course, this implies that the lemmas hold as well for w^n , proving the optimality of the SIP for the discrete time model. A direct proof of this result can be found in Weishaupt [78].

The generalization to other models, as the one with multiple servers (studied in the next section) or the customer assignment models studied earlier, are less direct because in these models there are events which have to be dealt with simultaneously. A more systematic study of different types of value function based on operators as T_1 and T_2 here can be found in Altman & Koole [2].

1.12. Server assignment model with multiple servers

In this section we study again the first model of the previous section, but with multiple servers. Server vacations are interesting here and therefore we model them as well. It was shown by Bruno et al. [7] that the policy that assigns the available servers to the jobs with lowest service intensities, i.e. the jobs with the largest expected processing time, minimizes the expected makespan. The optimal policy is called LEPT. Weber [76] generalized this to stochastic optimality. Giving conditions on the cost functions for LEPT to be optimal, for arrivals according to an MAP and arbitrary server vacations, is the main subject of this section. The cost function corresponding to the makespan will indeed appear to be allowable. Independently, similar results were derived by Chang et al. [12].

We assume that $\mu_1 \leq \dots \leq \mu_m$. As in section 1.10, s is the number of servers, and $s(x)$ is the number of servers currently available, i.e. $s(x)$ is determined by x , the state of the MAP. We assume $\gamma + s\mu_m \leq 1$, i.e. we have uniformized the model. Take $\mu = \mu_m$. An assignment action in (x, i) consists of the $s(x)$ class numbers to which the available servers are assigned. We introduce again class 0, $\mu_0 = 0$. If a server is assigned to class 0 it idles. Now we can assume that there will be no more servers assigned to a class then there are customers in that class. These actions are called admissible. The dynamic programming equation is:

$$\begin{aligned}
 v_{(x,i)}^{n+1} = & \min_{l_1, \dots, l_{s(x)}} \left\{ \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y,i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xy}^j\right) v_{(y,i)}^n \right) + \right. \\
 & \sum_{k=1}^{s(x)} \left(\mu_{l_k} v_{(x,i-e_{l_k})}^n + (\mu - \mu_{l_k}) v_{(x,i)}^n \right) + \left. \left(1 - \gamma - s(x)\mu\right) v_{(x,i)}^n \right\} = \\
 & \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y,i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xy}^j\right) v_{(y,i)}^n \right) + \\
 & \min_{l_1, \dots, l_{s(x)}} \left\{ \sum_{k=1}^{s(x)} \left(\mu_{l_k} v_{(x,i-e_{l_k})}^n + (\mu - \mu_{l_k}) v_{(x,i)}^n \right) \right\} + (1 - \gamma - s(x)\mu) v_{(x,i)}^n.
 \end{aligned} \tag{1.12.1}$$

To make the action unique we can assume $l_1 \leq \dots \leq l_{s(x)}$.

1.12.1. Lemma. If

$$\begin{aligned}
 \mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i)} &\leq \mu_{j_2} w_{(x,i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x,i)} \tag{1.12.2} \\
 &\text{for } j_1 < j_2 \text{ both admissible and } |i| \geq 2
 \end{aligned}$$

and

$$w_{(x,i-e_{j_1})} \leq w_{(x,i)} \text{ for } i_{j_1} > 0 \tag{1.12.3}$$

hold for the cost function v^0 , then they hold for all v^n .

If the number of customers is $s(x) + 1$ or more and there are admissible actions $j_1, \dots, j_{s(x)}$ and $j_1^*, j_2, \dots, j_{s(x)}$ with $j_1 < j_1^*$, then

$$\sum_{k=1}^{s(x)} \left(\mu_{j_k} v_{(x, i-e_{j_k})}^n + (\mu - \mu_{j_k}) v_{(x, i)}^n \right) \leq$$

$$\mu_{j_1^*} v_{(x, i-e_{j_1^*})}^n + (\mu - \mu_{j_1^*}) v_{(x, i)}^n + \sum_{k=2}^{s(x)} \left(\mu_{j_k} v_{(x, i-e_{j_k})}^n + (\mu - \mu_{j_k}) v_{(x, i)}^n \right)$$

is equivalent to (1.12.2). This means that (1.12.2) and (1.12.3) gives us the optimal policy. Equation (1.12.3) says that, if possible, no server should idle. By (1.12.2) we know that, when there are more than s customers, we should serve the group of customers with indexes as small as possible. Thus the SIP, which is here equal to LEPT, is optimal.

As contrasted with the single server case, we need (1.12.3) in the proof of (1.12.2). The model here is a special case of the model of section 3.6, thus for the proof we refer to the proof of lemma 3.6.1.

1.12.2. Theorem. *The SIP minimizes the costs at T for all cost functions satisfying (1.12.2) and (1.12.3).*

Note that the inequalities (1.12.2) and (1.12.3) are the same as (1.11.1) and (1.11.2). Thus, (1.11.3) and (1.11.4) characterize again the allowable cost functions. However, we have the extra condition $\mu_1 \leq \dots \leq \mu_m$. This means, in the case of linear costs, that the μc -rule is optimal in the multiple server model if $\mu_1 \leq \dots \leq \mu_m$ and $\mu_1 c_1 \geq \dots \geq \mu_m c_m$. To satisfy the monotonicity we assume $c_m \geq 0$. Note that if $\mu_1 = 0$ then (1.12.2) and (1.12.3) give that the costs in each state must be equal.

Now we go into the details of cost functions of the type $v_{(x, i)}^0 = \mathbf{I}\{|i| > 0\}$. As said in the previous section, we conclude that the probability that there are any customers present at T is minimized by LEPT.

We can modify the system such that it remains empty once it becomes empty, by taking $v_{(x, 0)}^{n+1} = \sum_y \lambda_{xy} v_{(y, 0)}^n + (1 - \sum_y \lambda_{xy}) v_{(x, 0)}^n$. Lemma 1.12.1 still holds for this model. In section 3.6 another approach with the same result is taken. Now we can study the probability that the system becomes empty before T . This means that the SIP minimizes the length of the busy period.

As shown in section 1.1, we can model the departure process of most queueing systems with an MAP. This way we can model tandem systems, of which the center with state i is the last in line, although we cannot let the actions taken in the first centers depend on the state of the last center, as this would introduce a dependence on the last center. Tandem models with this type of dependence are the subject of the next chapter. For tandem systems without this dependence, we might be interested in the moment the whole

system becomes empty. Now we have to take $v_{(x,i)}^0 = \mathbf{I}\{|i| > 0 \text{ or } |x| > 0\}$, where x is the vector denoting the state of all centers but the last one. This gives similar results as above, but now for emptiness of the whole system.

As argued in section 1.10, we can take sets $\tilde{\Lambda}_k \subset \Lambda$, denoting the set of states for which the number of arrivals in all classes at reaching that state is k or more. By taking $v_{(x,i)}^0 = \mathbf{I}\{|i| > 0 \text{ or } x \notin \tilde{\Lambda}_k\}$ we can study the first time after the k th arrival at which the system becomes empty. If there are no arrivals after the k th we have the makespan in the release date model of Weber [76] and Chang et al. [12]. Note that the conditions on the cost functions in Chang et al. [12] are the same as the conditions here. The generalization of this section consists of a more general arrival process.

When considering linear costs, we cannot take $c_1 = \dots = c_m = 1$ unless $\mu_1 = \dots = \mu_m$. This is not strange, because it is intuitively clear that LEPT does not minimize the number of customers at any T . The perhaps more logical candidate for optimality, the policy that serves customers with high service rates first (the SEPT policy), is not optimal either. This we show with the following example.

Take the following model: $s = 2$, $m = 2$, $\mu_1 = 2$ and $\mu_2 = 1$. There are no arrivals, and we start with $i_1 = 2$ and $i_2 = 1$. The objective function is the expected number of customers at T . The possible work-conserving policies are LEPT which starts serving a class 1 and a class 2 customer at time 0 and SEPT which starts with both class 1 customers. In the continuous-time model it is easy to compute the expected number of departed customers L at T using the following formula, with α_1 and α_2 the service rates of the customers served first, α_3 the rate of the other customer (note that $\alpha_3 < \alpha_1 + \alpha_2$):

$$\begin{aligned}
 L = & \int_0^T (\alpha_1 + \alpha_2) e^{-(\alpha_1 + \alpha_2)t} dt + \\
 & \frac{\alpha_1}{\alpha_1 + \alpha_2} \int_0^T (\alpha_1 + \alpha_2) e^{-(\alpha_1 + \alpha_2)t} (1 - e^{-\alpha_2(T-t)}) dt + \\
 & \frac{\alpha_2}{\alpha_1 + \alpha_2} \int_0^T (\alpha_1 + \alpha_2) e^{-(\alpha_1 + \alpha_2)t} (1 - e^{-\alpha_1(T-t)}) dt + \\
 & \int_0^T (\alpha_1 + \alpha_2) e^{-(\alpha_1 + \alpha_2)t} (1 - e^{-\alpha_3(T-t)}) dt = \\
 & 1 - e^{-(\alpha_1 + \alpha_2)T} + \\
 & 1 + e^{-(\alpha_1 + \alpha_2)T} - e^{-\alpha_1 T} - e^{-\alpha_2 T} + \\
 & 1 - e^{-(\alpha_1 + \alpha_2)T} - \frac{\alpha_1 + \alpha_2}{\alpha_1 + \alpha_2 - \alpha_3} e^{-\alpha_3 T} (1 - e^{-(\alpha_1 + \alpha_2 - \alpha_3)T}).
 \end{aligned}$$

The first line in the last expression is the probability that the first departure takes place before T . The second line is equal to $(1 - e^{-\alpha_1 T})(1 - e^{-\alpha_2 T})$, the

probability that both first scheduled jobs finish. The last term is concerned with the customer scheduled last.

Using a small computer program we computed L for LEPT ($\alpha_1 = 1$, $\alpha_2 = \alpha_3 = 2$) and the SEPT ($\alpha_1 = \alpha_2 = 2$, $\alpha_3 = 1$). For T small SEPT is better at 0 (for $T = 0.1$ we have $L = 0.380 \approx 4T$ for SEPT and $L = 0.302 \approx 3T$ for LEPT as can be expected from the infinitesimal properties). However, for T larger, LEPT is better (for $T = 3$ we have 2.929 for SEPT vs. 2.941 for LEPT). Thus there is no myopic optimal policy. Typically, the optimal policy is equal to LEPT at time 0 (if T is large enough), and change to SEPT as time goes on: if we are at $T - \varepsilon$ with ε small and still no customer have left SEPT is optimal. It is well known, see e.g. Weber [76] and Chang et al. [13], that if we replace the number of customers at T by the integral from 0 to T of the number of customers, i.e. if we consider flowtime, then SEPT is stochastically optimal.

Models with Markov Decision Arrival Processes

2.1. Markov Decision Arrival Processes

In the previous chapter we studied models with arrivals which were modeled by an MAP. In an MAP the arrival times depend only on x , the state of the MAP, and not on i , the state of the queues. In this chapter we generalize the MAP to allow for a certain type of dependency on the state of the queues. Of course, this dependency cannot be taken completely general. Take for example a customer assignment model in which arrivals occur more frequently if the queues are balanced. Then it is clear that it might be optimal to assign an arriving customer to the longest queue to suppress future arrivals. Therefore we model the dependence using actions in the arrival process, while keeping, for a fixed action, the transition intensities independent of the state of the queues. This leads to the following definitions. First we describe the arrival process without multiple customer classes or server vacations.

2.1.1. Definition. (Markov Decision Arrival Process) Let Λ be the countable state space of a Markov decision process with transition intensities λ_{xay} with $x, y \in \Lambda$ and $a \in A(x)$, the set of actions in x . When this process moves from x to y , while action a was chosen, then with probability q_{xay} an arrival occurs. We call the quadruple (Λ, A, λ, q) a Markov Decision Arrival Process (MDAP).

Note the similarity with the definition of the MAP in section 1.1: if we take $|A(x)| = 1$ for all x , we have an MAP. We use definition 2.1.1 in the sections on the customer assignment models. In the server assignment models we need again arrivals in multiple classes and server vacations. The equivalent of definition 1.10.1 is:

2.1.2. Definition. (Markov Decision Arrival Process) Let Λ be the countable state space of a Markov decision process with transition intensities λ_{xay} with $x, y \in \Lambda$ and $a \in A(x)$, the set of actions in x . When this process moves from x to y , while action a was chosen, then with probability q_{xay}^k an arrival in class $1 \leq k \leq m$ occurs, and with probability q_{xay}^{m+k} an event with server $1 \leq k \leq s$ occurs. There are sets $\Lambda_1^s, \dots, \Lambda_s^s$ such that server k is available if and only if $x \in \Lambda_k^s$, and sets $\Lambda_{1n}^a, \dots, \Lambda_{mn}^a$, $n \in \mathbb{N}$, such that if $x \in \Lambda_{kn}^a$ then there have been n or more arrivals of class k . We call the quadruple (Λ, A, λ, q) a Markov Decision Arrival Process (MDAP).

In the next section we start by illustrating the use of an MDAP for customer assignment models. Again we assume that the transition intensities in each state are equal, i.e. $\sum_y \lambda_{xay} = \gamma$ for all $x \in \Lambda$ and $a \in A(x)$.

2.2. Symmetric customer assignment model

We consider the model of section 1.2, but with arrivals according to an MDAP. Thus, we have m queues, with buffer sizes $B = (B_1, \dots, B_m)$ and service intensity μ . The results for this model are quite similar to the results of section 1.2, the only difference being the arrival process. To obtain optimality results both at T and from 0 to T we now need to introduce immediate costs $c_{(x,i)}$. See section 5.3 for more details. Before illustrating the use of the MDAP, we give the dynamic programming equation. Note the similarity with (1.2.1).

$$v_{(x,i)}^{n+1} = c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay} \min_j \{v_{(y,i+e_j)}^n\} + (1 - q_{xay}) v_{(y,i)}^n \right) \right\} + \sum_{j=1}^m \mu v_{(x,(i-e_j)+)}^n + (1 - \gamma - m\mu) v_{(x,i)}^n. \quad (2.2.1)$$

The second minimization ranges again over all j for which the queues are not full, i.e. for which $i_j < B_j$.

The MDAP is especially designed to model the arrivals at the last center of a tandem network. To show this, assume there are \tilde{m} (m) queues in the first (second) center, with state $(\tilde{i}_1, \dots, \tilde{i}_{\tilde{m}})$ ((i_1, \dots, i_m)), service intensities $\tilde{\mu}$ (μ) and buffer sizes \tilde{B} (B). The arrival process at the first center is Poisson with rate $\tilde{\lambda}$. Assignment actions are taken in both centers, and these actions are allowed to depend on the whole state of the system. Then the dynamic programming recursion is:

$$v_{(\tilde{i},i)}^{n+1} = c_{(\tilde{i},i)} + \min_a \left\{ \tilde{\lambda} v_{(\tilde{i}+e_a \wedge \tilde{B},i)}^n + \sum_{j_1=1}^{\tilde{m}} \left(\delta_{\tilde{i}_{j_1}} \tilde{\mu} \min_j \{v_{(\tilde{i}-e_{j_1},i+e_j \wedge B)}^n\} + (1 - \delta_{\tilde{i}_{j_1}}) \tilde{\mu} v_{(\tilde{i},i)}^n \right) \right\} + \sum_{j=1}^m \mu v_{(\tilde{i},(i-e_j)+)}^n + (1 - \tilde{\lambda} - \tilde{m}\tilde{\mu} - m\mu) v_{(\tilde{i},i)}^n.$$

Now, if we take $\lambda_{\tilde{i},a,\tilde{i}+e_a} = \tilde{\lambda}$ and $q_{\tilde{i},a,\tilde{i}+e_a} = 0$, $\lambda_{\tilde{i},a,\tilde{i}-e_j} = \tilde{\mu}$ and $q_{\tilde{i},a,\tilde{i}-e_j} = 1$ if $\tilde{i}_j > 0$, $\lambda_{\tilde{i}a\tilde{i}} = \gamma - \tilde{\lambda} - \tilde{\mu} \sum_{j=1}^{\tilde{m}} \delta_{\tilde{i}_j}$ and $q_{\tilde{i}a\tilde{i}} = 0$, and all other transition rates 0, then this recursive equation has the form of (2.2.1). Thus we have modeled the first center as an MDAP.

It is easy to see that, instead of a tandem system, we can model any network in which i is the state of a center without feedback to the network.

We return to the general model with an MDAP. As in the case with an MAP, we have the following result:

2.2.1. Lemma. *If*

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \text{ for } i_{j_1} \leq i_{j_2}, i + e_{j_1} + e_{j_2} \leq B, \quad (2.2.2)$$

$$w_{(x,i)} \leq w_{(x,i+e_{j_1})} \text{ for } i + e_{j_1} \leq B \quad (2.2.3)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \text{ for } i^* \text{ a permutation of } i, i^* \leq B \quad (2.2.4)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

For the proof we refer to the more general model of section 3.3. The result says again that an SQP is optimal, for suitable cost functions. An SQP, because SQP refers only to the assignment of the customers to the queues, and not to the action in the MDAP. For the tandem model described above it follows that it is optimal in the second center to employ the SQP, if the first center is also controlled optimally. How this first center should be controlled is the subject of the next section.

Because the optimal actions in the MDAP can depend on n , we cannot use the method of section 5.2, but we need a limiting argument. To use this, we have some minor restrictions on the cost functions. All cost functions considered here satisfy these conditions. Note that some of our cost functions, like $|i|$, are unbounded. Still they satisfy the conditions, which are given in assumption 5.3.1. Throughout this chapter we assume that this assumption holds.

2.2.2. Theorem. *For all T , an SQP minimizes the costs at T (and from 0 to T) for all cost functions satisfying (2.2.2) to (2.2.4).*

The conditions on the cost functions are exactly the same as in section 1.2, thus we refer to that section for a discussion of the allowable cost functions. Regarding stochastic optimality however, results are not as easy, as the optimal policy depends on the horizon. Of course, if v^0 is allowable, an SQP minimizes $\mathbf{I}\{v_{(x,i)}^0 > s\}$ for each value of s , but for different values of s different SQP's can be optimal. Examples showing this are easily given. Thus there is no single policy that is better than all policies R and all values of s . It is an open question if there is for every fixed policy R an SQP which is better for all s .

2.3. Tandems of customer assignment models

In this section we consider the tandem system introduced in the previous section, with $\tilde{\mu} = \mu$ and $\tilde{B} = B = \infty$. Customers arrive at the first center according to a Poisson process with intensity λ . In the previous section we saw that in the second center the SQP should be used. Here we study the optimal assignment in the first center. We use the same notation. First we show that the SQP is not optimal in the first center.

Consider a system in which there is only one arrival at time 0. We compute the expected flowtime (which is the sum of the departure times). As the initial state we take $\tilde{i} = (1, 0)$, $i = (5, 5)$. Thus we have to decide whether to route the arriving customer to queue 2 or to queue 1 at the first center. Take $\mu = 1$. Let us denote the expected flowtime if we start with (\tilde{i}, i) with $f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix})$. These numbers can be calculated with the recursive formulae

$$\left\{ \begin{array}{l} f(\begin{smallmatrix} 00 \\ 00 \end{smallmatrix}) = 0, \\ f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) = (\tilde{i}_1 + \tilde{i}_2 + i_1 + i_2 + \delta_{\tilde{i}_1} f(\begin{smallmatrix} \tilde{i}_1 - 1 & i_1 + 1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) + \delta_{\tilde{i}_2} f(\begin{smallmatrix} \tilde{i}_1 & i_1 + 1 \\ \tilde{i}_2 - 1 & i_2 \end{smallmatrix}) + \\ \quad \delta_{i_1} f(\begin{smallmatrix} \tilde{i}_1 & i_1 - 1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) + \delta_{i_2} f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 - 1 \end{smallmatrix})) / (\delta_{\tilde{i}_1} + \delta_{\tilde{i}_2} + \delta_{i_1} + \delta_{i_2}) \text{ if } i_1 \leq i_2, \\ f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) = (\tilde{i}_1 + \tilde{i}_2 + i_1 + i_2 + \delta_{\tilde{i}_1} f(\begin{smallmatrix} \tilde{i}_1 - 1 & i_1 \\ \tilde{i}_2 & i_2 + 1 \end{smallmatrix}) + \delta_{\tilde{i}_2} f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 - 1 & i_2 + 1 \end{smallmatrix}) + \\ \quad \delta_{i_1} f(\begin{smallmatrix} \tilde{i}_1 & i_1 - 1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) + \delta_{i_2} f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 - 1 \end{smallmatrix})) / (\delta_{\tilde{i}_1} + \delta_{\tilde{i}_2} + \delta_{i_1} + \delta_{i_2}) \text{ if } i_1 \geq i_2, \end{array} \right.$$

We found that $f(\begin{smallmatrix} 25 \\ 05 \end{smallmatrix}) = 41.63 < 41.67 = f(\begin{smallmatrix} 15 \\ 15 \end{smallmatrix})$. Because the flowtime is the integral of the number of customers over time, there are T 's for which the number of customers at T is not minimized by the SQP in both centers. This is because if it were, the expected number of customers would be smaller under the SQP for all T , and so the flowtime would also be smaller.

Define $f^T(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix})$ as the expected flowtime up to T , i.e. the expected number of customers integrated from 0 to T . Add an extra superscript A to denote the model with Poisson(λ) arrivals. It is easily seen that $f(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) - f^T(\begin{smallmatrix} \tilde{i}_1 & i_1 \\ \tilde{i}_2 & i_2 \end{smallmatrix}) \rightarrow 0$, as T increases. Take T such that this difference is smaller than 0.01. Take λ small enough such that the expected flowtime of the arrivals before T is smaller than 0.01. Then we have

$$f^{TA}(\begin{smallmatrix} 25 \\ 05 \end{smallmatrix}) \leq f^T(\begin{smallmatrix} 25 \\ 05 \end{smallmatrix}) + 0.01 \leq f(\begin{smallmatrix} 25 \\ 05 \end{smallmatrix}) + 0.01 < f(\begin{smallmatrix} 15 \\ 15 \end{smallmatrix}) - 0.01 \leq f^T(\begin{smallmatrix} 15 \\ 15 \end{smallmatrix}) \leq f^{TA}(\begin{smallmatrix} 15 \\ 15 \end{smallmatrix}),$$

where the first inequality follows by the choice of λ , and the fourth follows by the choice of T . This shows that, for λ sufficiently small, there are states in which routing according to the SQP is suboptimal.

An intuitive explanation of this phenomenon is easily given. When both queues of the second center are heavily loaded, it pays to delay arriving customers, which allows one to see how the center evolves in time. This can be done by assigning customers arriving at the first center to the longest queue.

To study the optimal policy for more realistic values of λ than considered in the last section we did various numerical calculations on the two center model.

Again we fixed the service rate μ to 1 and varied the arrival rate λ . Because we used successive approximation (see appendix D for a practical discussion of computational algorithms) we had to introduce buffers (in each queue equal) to make the state space finite. We also varied these buffer sizes to study the influence of the finite buffers on our model. To minimize blocking influence we assumed that no service takes place at the first center if the second center is full. Note that this type of arrival process cannot be modeled with an MDAP, due to the blocking protocol. We computed the optimal policy in the first center for discounted and average costs. Our results are summarized in the two tables below. First we consider discounted costs.

In Hordijk & Koole [22] we took as immediate reward the expected number of departed customers. The advantage of taking this reward is that the optimal policy does not seem to depend on the buffer sizes. Because we have worked so far with the total number of customers we do the same in the present calculations. However, this means a stronger buffer influence, especially when $\lambda \geq 2$. Therefore we only considered $\lambda < 2$ here. Because $\mu = 1$, it follows from Kingman [34] that a single center model operated by the SQP has a stationary distribution under this assumption. (See Adan et al. [1] for a recent result and references on computational issues regarding the SQP.) Thus, if we take B large enough, we expect to have little buffer influence.

The results for B varying from 20 to 45 are shown in table 2.3.1. For each combination of β , the discount factor, and λ the table contains the maximum relative difference between the optimal policy and the SQP, and the state where this maximum is attained. These numbers are calculated with the formula $\max_i \{(v_i^\beta(\text{SQP}) - v_i^\beta)/v_i^\beta\}$, where v_i^β ($v_i^\beta(\text{SQP})$) are the costs under the optimal policy (the SQP) and the minimization is taken over all possible states. It is clear from the table that the SQP is nearly optimal. In some cases the difference decreases as B increases, thus in these cases the SQP might be optimal for $B = \infty$. In these states we increased B , if possible, until the relative difference was smaller than 10^{-15} . In the cases with $\lambda = 1.5$ and $\beta = 0.75$, and $\lambda = 1.9$ and $\beta = 0.5$ and 0.75 we were, due to computational difficulties, not able to increase B any further.

	$\beta = 0.01$	0.1	0.25	0.5	0.75
$\lambda = 0.1$	$<10^{-15}$	$1.5 \cdot 10^{-13}, (\begin{smallmatrix} 0 & 14 \\ 1 & 14 \end{smallmatrix})$	$1.7 \cdot 10^{-9}, (\begin{smallmatrix} 0 & 10 \\ 1 & 10 \end{smallmatrix})$	$5.9 \cdot 10^{-7}, (\begin{smallmatrix} 0 & 8 \\ 1 & 8 \end{smallmatrix})$	$1.8 \cdot 10^{-5}, (\begin{smallmatrix} 0 & 9 \\ 2 & 9 \end{smallmatrix})$
0.25	$<10^{-15}$	$1.6 \cdot 10^{-13}, (\begin{smallmatrix} 0 & 14 \\ 1 & 14 \end{smallmatrix})$	$2.2 \cdot 10^{-9}, (\begin{smallmatrix} 0 & 10 \\ 1 & 10 \end{smallmatrix})$	$8.6 \cdot 10^{-7}, (\begin{smallmatrix} 0 & 9 \\ 1 & 9 \end{smallmatrix})$	$2.6 \cdot 10^{-5}, (\begin{smallmatrix} 0 & 9 \\ 2 & 9 \end{smallmatrix})$
0.5	$<10^{-15}$	$4.5 \cdot 10^{-14}, (\begin{smallmatrix} 0 & 15 \\ 1 & 15 \end{smallmatrix})$	$1.1 \cdot 10^{-9}, (\begin{smallmatrix} 0 & 11 \\ 1 & 11 \end{smallmatrix})$	$6.3 \cdot 10^{-7}, (\begin{smallmatrix} 0 & 9 \\ 1 & 9 \end{smallmatrix})$	$2.2 \cdot 10^{-5}, (\begin{smallmatrix} 0 & 10 \\ 2 & 10 \end{smallmatrix})$
1	$<10^{-15}$	$<10^{-15}$	$4.2 \cdot 10^{-11}, (\begin{smallmatrix} 0 & 13 \\ 1 & 13 \end{smallmatrix})$	$5.1 \cdot 10^{-8}, (\begin{smallmatrix} 0 & 11 \\ 1 & 11 \end{smallmatrix})$	$1.9 \cdot 10^{-6}, (\begin{smallmatrix} 0 & 11 \\ 1 & 11 \end{smallmatrix})$
1.5	$<10^{-15}$	$<10^{-15}$	$2.2 \cdot 10^{-14}, (\begin{smallmatrix} 0 & 18 \\ 1 & 18 \end{smallmatrix})$	$8.4 \cdot 10^{-11}, (\begin{smallmatrix} 0 & 15 \\ 1 & 15 \end{smallmatrix})$	$<10^{-11}$
1.9	$<10^{-15}$	$<10^{-15}$	$<10^{-15}$	$<10^{-14}$	$<10^{-7}$

Table 2.3.1. Discounted costs

In the average cost case we again compared the optimal policy and the SQP. The relative difference between their average costs can be found in table 2.3.2. Once again the SQP is nearly optimal. When comparing both tables, we see that the average cost case does not behave as the limiting case for the discounted cost case. A possible explanation is the following. The difference in the assignments in state $\binom{0 \ k}{1 \ k}$, with $k \approx 10$, manifests itself if the second center becomes empty. As there are many customers in the second center, this requires that β is close to 1. However, in the limiting average case the dependence on the starting state has disappeared, and the effect of the few states where the optimal action is not according to the SQP is small. It is interesting to note that the states where the SQP is not optimal all have the same form, with few customers in the first center, and heavily loaded, balanced queues in the second center, as in the example at the beginning of this section.

$\lambda = .1$.25	.5	1	1.5	1.9
$<10^{-15}$	$1.3 \cdot 10^{-12}$	$2.1 \cdot 10^{-10}$	$<10^{-9}$	$<10^{-15}$	$<10^{-4}$

Table 2.3.2. Average costs

Having seen that the SQP is not optimal in the case where the policies depend on the whole state of the system, we continue with studying the case where the policies only depend on local information, i.e. where the policy at a certain center depends only on the state of that center. We call this the *partial information* case, as contrasted to the *full information* case.

At the end of this section we study the general partial information case. We will see there that also in this case a counterexample to the optimality of the SQP can be constructed, for discounted costs. If we restrict the class of admissible policies even more, namely to *static policies* at the second center, we can prove the optimality of the SQP. A policy R is called a static policy if it is defined by a sequence of random variables $\{\Pi_n, n \in \mathbb{N}\}$, where $\Pi_n = j$ corresponds to routing the n th arriving customer to queue j . The routing probabilities are stochastically independent of the queue lengths and the arrival times. Both the Equal Splitting Policy and the Cyclic Assignment Policy of section 1.9 are static, as can be shown by taking all Π_n independent and $\mathbb{P}(\Pi_n = 1) = \frac{1}{2}$ for all n for the ESP and $\mathbb{P}(\Pi_{n+1} = j + 1(\bmod 2) | \Pi_n = j) = 1$ for all n for the CAP. The SQP is not static. We prove that, for partial information, the SQP in both centers gives an earlier departure process than the two center policy which uses a static policy in the second center. Because we use coupling this means also that the number of customers is minimized by the SQP in both centers.

To show our result we need two theorems. The first states that the SQP gives a pathwise earlier departure process. The second theorem says that for a static policy an earlier arrival process gives an earlier departure process. Combining these theorems gives indeed the result on static policies. We see an arrival process as a sequence of arrival times. That is, the arrival process $V = \{V_n, n \in \mathbb{N}\}$ has V_n as the time of the n th arrival. For arrival processes

$V = \{V_n, n \in \mathbb{N}\}$ and $W = \{W_n, n \in \mathbb{N}\}$ we say that V is pathwise earlier than W and we write $V \leq_p W$ if there are arrival processes V^* and W^* with $V^* \stackrel{d}{=} V$ and $W^* \stackrel{d}{=} W$ such that $V_n^*(\omega) \leq W_n^*(\omega)$ for all ω in some probability space and $n \in \mathbb{N}$. We use a similar definition and notation for departure processes. With $\stackrel{d}{=}$ we mean that the processes on either side have the same distribution. By theorem 1.5.2 we have:

2.3.1. Corollary. *Consider a center with two parallel queues, arrival process U and policy SQP and a similar center with an arbitrary policy R . If V and \tilde{V} are the respective departure processes, then $V \leq_p \tilde{V}$.*

2.3.2. Theorem. *Consider one center with two parallel queues and a static policy R . For arrival processes T and \tilde{T} the departure processes are denoted by V and \tilde{V} , respectively. If $T \leq_p \tilde{T}$, then $V \leq_p \tilde{V}$.*

Proof. Because $T \leq_p \tilde{T}$ there are arrival processes T^* and \tilde{T}^* with $T \stackrel{d}{=} T^*$ and $\tilde{T} \stackrel{d}{=} \tilde{T}^*$ such that $T_n^*(\omega) \leq \tilde{T}_n^*(\omega)$ for all n and ω . Fix $\omega \in \Omega$. We use the following notation: $T_n^*(\omega) = t_n$, $\tilde{T}_n^*(\omega) = \tilde{t}_n$. Let S_n (\tilde{S}_n) be the service time of the n th customer and U_n (\tilde{U}_n) the queue to which the n th customer is routed. Of course $S_n \stackrel{d}{=} \tilde{S}_n$. Because R is static we also have $U_n \stackrel{d}{=} \tilde{U}_n$. Hence by coupling arguments we may assume that $S_n = \tilde{S}_n$ and $U_n = \tilde{U}_n$ for all n . Denote an arbitrary realization of $S_n, U_n, n > 0$ with $s_n, u_n, n > 0$. We omit the superscript $*$. Let $\xi^{T,V}(t)$ ($\tilde{\xi}^{T,V}(t)$) be the number of arrived (served) customers at time t . A subscript j denotes a specific queue. Then

$$\begin{aligned} \xi_j^V(t) &= \sum_{n=1}^{\xi^T(t)} \mathbf{I}\{u_n = j\} \mathbf{I}\{t_k + \sum_{l=k}^n \mathbf{I}\{u_l = j\} s_l \leq t, k = 1, \dots, n\} \\ &\geq \sum_{n=1}^{\tilde{\xi}^T(t)} \mathbf{I}\{u_n = j\} \mathbf{I}\{t_k + \sum_{l=k}^n \mathbf{I}\{u_l = j\} s_l \leq t, k = 1, \dots, n\} \\ &\geq \sum_{n=1}^{\tilde{\xi}^T(t)} \mathbf{I}\{u_n = j\} \mathbf{I}\{\tilde{t}_k + \sum_{l=k}^n \mathbf{I}\{u_l = j\} s_l \leq t, k = 1, \dots, n\} \\ &= \tilde{\xi}_j^V(t), \quad j = 1, 2. \end{aligned}$$

Thus $\xi^V(t) = \xi_1^V(t) + \xi_2^V(t) \geq \tilde{\xi}^V(t)$ for all t and $t_n^V \leq \tilde{t}_n^V$ for all n . \square

This theorem is also true for general service times. Unfortunately, it does not hold for the SQP as the following counterexample shows.

Take $T_1 = T_2 = T_3 = \tilde{T}_1 = \tilde{T}_2 = 0$; $\tilde{T}_3 = h$; $T_n = \tilde{T}_n > 1 + h$ for all $n \geq 4$. Thus $T \leq_p \tilde{T}$. Compare the probabilities that 2 customers have left at $t = 1 + h$. Condition on the number of departures in $[0, h]$. If no departures occur in $[0, h]$, the two systems are the same.

On the other hand, if exactly one departure occurs in $[0, h]$, the time until the next departure in the T -model has with probability $\frac{1}{2}$ an exponential distribution with parameter μ and with probability $\frac{1}{2}$ an exponential distribution with parameter 2μ . Indeed, the customer departing in $[0, h]$ leaves the queue with one customer with probability $\frac{1}{2}$ and the queue with two customers with probability $\frac{1}{2}$ as well. In the \tilde{T} -model the at h arriving customer chooses the empty queue, therefore the time until the next departure is $\text{Erlang}(2\mu)$ distributed. The difference between these two probabilities, say c , does not depend on h , but only on μ . The probability that one customer leaves in $[0, h]$ is equal to $2\mu h + o(h)$.

The probability that two customers leave in $[0, h]$ is $o(h)$. Now we have:

$$\begin{aligned} \mathbb{P}_T(2 \text{ customers leave in } [0, 1+h]) - \\ \mathbb{P}_{\tilde{T}}(2 \text{ customers leave in } [0, 1+h]) = 2\mu hc + o(h) > 0, \end{aligned}$$

if h is small enough. Note that the idea behind this counterexample is similar to the counterexample in the full information case; there, by sending to the longer queue, the arrivals of the customers at the second center were delayed.

Combining corollary 2.3.1 and theorem 2.3.2 gives the following result for the two centers in tandem.

2.3.3. Theorem. *Let $R = (R_1, R_2)$ be the two center policy with static policy R_2 in center 2. Let $R^* = (\text{SQP}, \text{SQP})$ be the two center policy which uses the SQP in both centers. For a general arrival process T let W (\hat{W}) be the departure processes of the second center under R (R^*). Then $\hat{W} \leq_p W$.*

Proof. The proof follows easily from corollary 2.3.1 and theorem 2.3.2. As depicted in figure 2.3.1 let V (\tilde{V}) denote the departure processes of the first center under policy R (R^*). The departure process of the second center for the policy (SQP, R_2) is denoted by \tilde{W} .

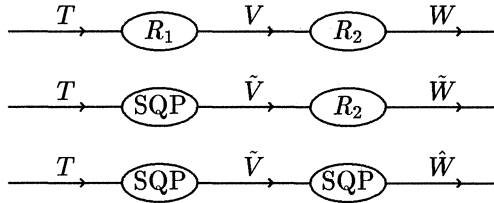


Figure 2.3.1.

From corollary 2.3.1 we have $\tilde{V} \leq_p V$. Hence by theorem 2.3.2 $\tilde{W} \leq_p W$. Corollary 2.3.1 also gives $\hat{W} \leq_p \tilde{W}$. Combining the last two inequalities yields $\hat{W} \leq_p W$. \square

This result can also be found in [22].

It is straightforward to generalize theorem 2.3.3 to a network of centers in tandem. The proof goes by induction on the number of centers. Suppose it is true for k centers. Assume that V, \tilde{V} is the departure process of the k th center when using (R_1, \dots, R_k) with R_i static for $2 \leq i \leq k$, respectively the SQP in each center. Then by the induction hypothesis $\tilde{V} \leq_p V$ and we can use again the same arguments as in the proof of theorem 2.3.3.

In the partial information case the policy in each center is not allowed to depend on the state of the other center. But, the fact that the state of the other center is unknown does not mean that there is no information on the other center. For example, in the discounted cost case, decisions taken early in time weight more heavily than decisions taken later. This means that there is a dependence on the starting state. The same phenomenon occurs for average costs, for multichain models. Of course the model studied here is unichain, but for discounted costs we were able to construct a policy R^* , which uses partial information, that is better for certain starting states than the SQP. In center 2 R^* uses the SQP. Also in center 1 the SQP is used, except in state $(0, 1)$ and $(1, 0)$. With the counterexample from the beginning of this section in mind, we might expect that R^* performs better than the SQP for starting states of the form $(0, 1, k, k)$ and suitable choices of parameters. Indeed, take $\mu = 1$, $\lambda = 0.01$ and discount factor 0.9 in the discrete-time normalized model. Then the infinite horizon expected discounted number of customers is smaller under the SQP for starting states like $(0, 1, 0, 0)$, but R^* is better for starting state $(0, 1, 10, 10)$. The relative differences are respectively $1.9 \cdot 10^{-3}$ and $-1.3 \cdot 10^{-7}$. Whether the SQP is optimal for average costs remains an open question. On the one hand, the model is unichain, and therefore the optimal policy is independent of the starting state; on the other hand, the numbers of customers in the centers are not independent, meaning that some information on the state of the other center can be obtained from the state of the present center.

In Hordijk & Koole [22] we conjectured that the SQP is optimal in the partial information case. This is clearly falsified by the present results. Our conjecture was based on numerical results obtained by Loeve & Pols [44], who used an algorithm derived by Kulkarni & Serin [41] to find local optima or saddle points in the class of policies that use partial information. In all problem instances they considered, the SQP is optimal.

2.4. Networks of customer assignment models with workloads

The model studied in section 1.8 is concerned with the workloads and not with the actual departures. This means that we cannot distinguish the departure epochs of the customers. Therefore it is not of interest to the network models studied in this chapter to generalize the dynamic programming result of theorem 1.8.2 on the optimality of the SWP to arrivals according to an MDAP. For this reason we leave this generalization to chapter 3. Of interest here is to try to obtain similar results as in section 2.3 for tandem systems and other networks.

In section 1.5 we saw that the SQP is pathwise optimal in the queue length model. The same holds for the workload model, as stated in section 1.8. However, we showed in the previous section that the SQP is not monotone in the sense that earlier arrivals give earlier departures. This is not the case for the SWP. Because the SWP is equivalent to a single multi-server queue with FCFS discipline, the monotonicity is easily shown by a coupling argument. This gives the following result.

2.4.1. Theorem. *Let $R = (R_1, R_2)$ be a two center policy with R_1 and R_2 not depending on the workload of the other center (thus R uses partial information). Denote with $R^* = (\text{SWP}, \text{SWP})$ the two center policy which use the SWP in both centers. For a general arrival process T let W (\tilde{W}) be the departure processes of the second center under R (R^*). Then $\tilde{W} \leq_p W$.*

Proof. Again, the result follows easily by considering a picture.

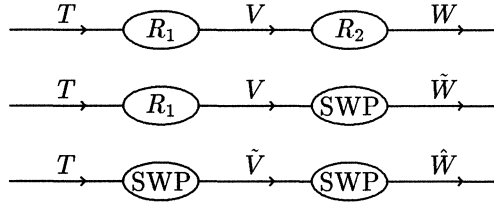


Figure 2.4.1.

By the pathwise optimality of the SWP we have $\tilde{W} \leq_p W$ and $\tilde{V} \leq_p V$. By the monotonicity of the SWP we have $\hat{W} \leq_p \tilde{W}$. Combining the inequalities yields $\hat{W} \leq_p W$. \square

Due to the fact that the SWP is monotone we can generalize the results to networks of centers with feedback to the network, and to policies using full information. Related to this are the results of Righter & Shanthikumar [59]. They also consider networks of centers, each with one server, and show that, in the case of a service time distribution with an increasing likelihood ratio, the departures are earlier if the customers are served non-preemptively. Monotonicity plays an important role there too.

Consider c centers, where routing between the centers is according to static rules. Remember that static policies were introduced in the previous section for the assignment of customers to parallel queues within a center. Here they are used for routing between different centers. The model is either open or closed. Let R be an arbitrary policy, that possibly uses information on all centers. In the case of random service times, the assignment decisions are not allowed to depend on the workloads. Now, let $V(i, j)$ ($\tilde{V}(i, j)$) be the stream of customers going from center i to center j , using the SWP (R). Outside arrivals are assumed to be coming from center 0.

2.4.2. Theorem. $V(i, j) \leq_p \tilde{V}(i, j)$ for all i and j .

Proof. Due to the (possible) feedback in the network arrival times depend on prior departure times. Therefore we cannot use arguments similar to those in the proof of theorem 2.4.1. We couple the networks, one using the SWP and one using R , by constructing $V^*(i, j)$ and $\tilde{V}^*(i, j)$ with $V^*(i, j) \stackrel{d}{=} V(i, j)$ and $\tilde{V}^*(i, j) \stackrel{d}{=} \tilde{V}(i, j)$ for all i and j . The routing is coupled by letting the n th customer that leaves center i go to the same center in both networks. Note that, by taking $i = 0$, we have $V^*(0, j) = \tilde{V}^*(0, j)$. In the case of deterministic service times the models are completely coupled now. The service times are coupled for each queue separately, such that the departures are earlier under the SWP. Now consider a realization.

Events in the networks with streams V^* and \tilde{V}^* occur at points $v_1 < v_2 < \dots$ and $\tilde{v}_1 < \tilde{v}_2 < \dots$. Each event consists of a transition of a customer from one center to another. Transitions from center i to center j occur at $v_1(i, j) < v_2(i, j) < \dots$ and $\tilde{v}_1(i, j) < \tilde{v}_2(i, j) < \dots$. (If 2 or more events occur at the same time, we assume that they are logically ordered. For example, if a customer arrives at a center, receives 0 processing time and leaves again, we assume that the arrival occurs before the departure.) We use the fact that if the arrivals up to T at a certain center are earlier in the SWP model, then the departures up to T are also earlier. The proof uses induction on the number of events in the network operated by R . Choose n^* . Define n_{ij}^* as follows: $\tilde{v}_{n_{ij}^*}(i, j) \leq \tilde{v}_{n^*} < \tilde{v}_{n_{ij}^*+1}(i, j)$. Suppose

$$v_l(i, j) \leq \tilde{v}_l(i, j) \text{ for all } l = 1, \dots, n_{ij}^*, i \text{ and } j.$$

Consider transition $n^* + 1$ in the network operated by R . Suppose that a customer moves from center i^* to center j^* at this transition. Consider center i^* . By the induction hypothesis for $j = i^*$, the arrivals at i^* before \tilde{v}_{n^*} are earlier under the SWP. Because there are no arrivals at center i^* between \tilde{v}_{n^*} and \tilde{v}_{n^*+1} in the network operated by R , also the arrivals before \tilde{v}_{n^*+1} are earlier under the SWP. By the optimality and monotonicity of the SWP, the departures are also earlier, and thus $v_{n_{i^*j^*}^*+1} \leq \tilde{v}_{n_{i^*j^*}^*+1}$, completing the induction step. \square

Note that we can also have non-controllable centers in the network, as long as they are monotone. Even more general, we can also insert controllable centers of the type considered in Righter & Shanthikumar [59]. Another possibility is the inclusion of centers with Bernoulli routing, as long as the assignment in the center is more balanced for R^* than for R . This follows from the monotonicity of this type of center (theorem 2.3.2), and from the pathwise optimality (as shown in section 1.9). The next corollary follows easily.

2.4.3. Corollary. *In a closed network, the SWP maximizes the throughput. In an open network, the SWP minimizes the number of customers in the system.*

For their model Righter & Shanthikumar [59] formulate a similar corollary.

2.5. Server assignment model with multiple servers

In section 2.2 the dynamic programming results of section 1.2 were easily generalized to arrivals according to an MDAP. The generalization is possible in most customer assignment models. For the server assignment models it is more complicated. In this section we show that lemma 1.12.1, which shows the optimality of LEPT, can be generalized to arrivals according to an MDAP. This means that, as in the customer assignment models, LEPT is optimal in the last center of a tandem system, where each center has its own servers and customers keep their class. Lemma 1.11.1 however, which deals with single server models, cannot be generalized in its full generality as two counterexamples show.

We follow the analysis of section 1.12. Again assume $\mu_1 \leq \dots \leq \mu_m$. The other remarks made there are also valid here. The analogue of (1.12.1) is:

$$v_{(x,i)}^{n+1} = c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{ay} \left(\sum_{j=1}^m q_{xay}^j v_{(y,i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xay}^j \right) v_{(y,i)}^n \right) \right\} +$$

$$\min_{l_1, \dots, l_{s(x)}} \left\{ \sum_{k=1}^{s(x)} \left(\mu_{l_k} v_{(x,i-e_{l_k})}^n + (\mu - \mu_{l_k}) v_{(x,i)}^n \right) \right\} + (1 - \gamma - s(x)\mu) v_{(x,i)}^n.$$

The lemma which gives the optimal policy is also the same:

2.5.1. Lemma. *If*

$$\mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i)} \leq \mu_{j_2} w_{(x,i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x,i)} \quad (2.5.1)$$

for $i_{j_1}, i_{j_2} > 0$ and $j_1 < j_2$

and

$$w_{(x,i-e_{j_1})} \leq w_{(x,i)} \quad \text{for } i_{j_1} > 0 \quad (2.5.2)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The model studied in section 3.6 is a generalization of the present model, for example with partial availability of the servers. For a proof of the lemma we refer to the proof of lemma 3.6.1.

2.5.2. Theorem. A SIP minimizes the costs at T (and from 0 to T) for all cost functions satisfying (2.5.1) and (2.5.2).

The same cost functions as in section 1.12 are allowable here. Of course, this includes the optimality of the SIP in the single server case if $\mu_1 \leq \dots \leq \mu_m$. If the queues are not ordered this way, we have seen in section 1.12 that the SIP is in general not optimal in the multiple server case. Because the MDAP is a generalization of the MAP, this also holds for the present model. However, in the single server case the SIP was optimal, independent of the ordering. This does not hold in the case of MDAP's, as the following counterexamples show. Summarizing, in the case of dependent arrivals we need $\mu_1 \leq \dots \leq \mu_m$ both in the multiple and in the single server case. This result can also be found in [25].

We consider a system of two centers in tandem, each with two queues, where each center has one server. There are no arrivals, and when a customer leaves queue j at the first center, it enters at the second center again queue j . We show, for certain choices of the service parameters, holding costs and starting states, that the μc -rule in the second center is not optimal. This contradicts the results in section 4 of Nain [49] and in section 2 of Nain et al. [50]. We show that the expected total costs over the infinite horizon are not minimized by a policy that uses the μc -rule in the second center. This means that there are T 's for which the μc -rule does not minimize the expected costs at T . The first example (which can be found in [23]) is the simplest, although we must assume that the policies allow idling in the first center. In the second example this is not the case.

We use a similar notation for the tandem system as in section 2.2, i.e. we add a tilde to denote the first center. The parameters of the first example are given in figure 2.5.1.

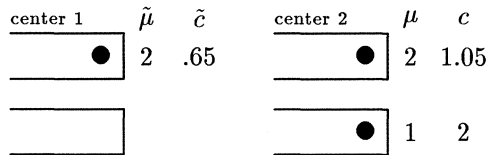


Figure 2.5.1

Denote by K_{ijk} the total expected holding cost when at time 0 there are i customers in the first queue of center 1, j customers in the first queue of center 2 and k customers in the second queue of center 2. It follows from the optimality of the μc -rule for a single center that the optimal policy in center 2 is the μc -rule when center 1 is empty. Because the holding costs are positive, idleness in center 2 is not optimal. Hence the total expected holding cost for the optimal policy in starting states with the first center empty are:

$$\begin{aligned}
K_{001} &= \frac{2}{1} = 2; \\
K_{010} &= \frac{1.05}{2} = 0.525; \\
K_{011} &= \frac{3.05}{2} + K_{001} = 3.525; \\
K_{020} &= \frac{2.1}{2} + K_{010} = 1.575; \\
K_{021} &= \frac{4.1}{2} + K_{011} = 5.575.
\end{aligned}$$

In starting states with customers in both centers the total expected holding cost is the minimum of terms corresponding to different actions. Denote by (i, j) the possible actions: i (j) is the queue served in center 1 (2). The successive terms in the computation below correspond to the action pairs $(1, 1)$, $(1, 2)$, $(2, 1)$ and $(2, 2)$ respectively, where terms belonging to actions corresponding to idleness in center 2 are deleted. The optimal action pair in starting state ijk is denoted by a_{ijk} .

$$\begin{aligned}
K_{100} &= \frac{0.65}{2} + K_{010} = 0.85; \\
K_{101} &= \min\left\{\frac{2.65}{3} + \frac{1}{3}K_{100} + \frac{2}{3}K_{011}; \frac{2.65}{1} + K_{100}\right\} \\
&= \min\{3.51667; 3.5\} = 3.5; \\
a_{101} &= (2, 2); \\
K_{110} &= \min\left\{\frac{1.7}{4} + \frac{1}{2}K_{020} + \frac{1}{2}K_{100}; \frac{1.7}{2} + K_{100}\right\} \\
&= \min\{1.6375; 1.7\} = 1.6375; \\
a_{110} &= (1, 1); \\
K_{111} &= \min\left\{\frac{3.7}{4} + \frac{1}{2}K_{021} + \frac{1}{2}K_{101}; \frac{3.7}{3} + \frac{1}{3}K_{110} + \frac{2}{3}K_{021}; \right. \\
&\quad \left. \frac{3.7}{2} + K_{101}; \frac{3.7}{1} + K_{110}\right\} \\
&= \min\{5.4625; 5.49583; 5.35; 5.3375\} = 5.3375; \\
a_{111} &= (2, 2).
\end{aligned}$$

From $a_{110} = (1, 1)$, $a_{101} = (2, 2)$ and $a_{111} = (2, 2)$ we conclude that the server in center 1 starts serving the job in queue 1 after the job in queue 2 of center 2 has finished its service. Hence the optimal action in center 1 depends on the state in center 2. Since $a_{111} = (2, 2)$ the server in center 2 serves the job in queue 2 before the job in queue 1, thus the μc -rule is not optimal at center 2. Note that the optimal action in center 2 depends on the state in center 1.

The error in Nain [49] and Nain et al. [50] can best be explained with the help of the example. Basically, in both articles, the authors try to improve

an arbitrary policy by keeping the behavior of the first center the same and changing the policy in the second center to the μc -rule. In the example, the customer in center 1 (customer 1) is not served until the customer in center 2, queue 2 (customer 3) has departed. We change the policy by serving the customer in center 2, queue 1 (customer 2) first, but now we cannot let the server in center 1 be idle for the service time of customer 3, because we do not know its service time yet.

Another possibility, by which we keep the stochastic behavior of center 1 the same, is taking the idle time at center 1 independent of the service time of customer 3, but with the same distribution. In the example, the server at center 1 idles during an exponentially distributed time with parameter 1, while customer 2 is served at center 2. This trivially does not improve the optimal policy, but we also calculate it.

Let K_{1jk}^* denote the total expected holding cost when the customer, initially in center 1, is still there and when the server starts idling. With K_{ijk} we denote the same, if the customer in center 1 has already departed or if the server at center 1 is servicing the customer. Since the policy is fixed there is no minimization in the computation. The total expected holding cost for states with $i = 0$ and state 100 are equal to those of the optimal policy. The computation of the other values is as follows,

$$\begin{aligned} K_{100}^* &= \frac{0.65}{1} + K_{100} = 1.5; \\ K_{101} &= \frac{2.65}{3} + \frac{2}{3}K_{011} + \frac{1}{3}K_{100} = 3.51667; \\ K_{101}^* &= \frac{2.65}{2} + \frac{1}{2}K_{101} + \frac{1}{2}K_{100}^* = 3.83333; \\ K_{111} &= \frac{3.7}{4} + \frac{1}{2}K_{021} + \frac{1}{2}K_{101} = 5.47083; \\ K_{111}^* &= \frac{3.7}{3} + \frac{1}{3}K_{111} + \frac{2}{3}K_{101}^* = 5.6125. \end{aligned}$$

Indeed we see, when comparing K_{111}^* with K_{111} previously obtained, that K_{111}^* is larger.

The second example has 4 customers present, one in each of the 4 queues. The parameters of the exponential distributions and the holding costs are given in figure 2.5.2.

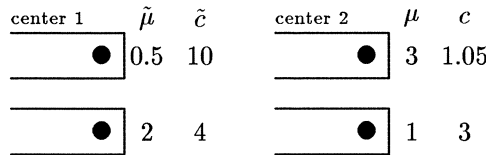


Figure 2.5.2

Straightforward calculation gives in this model the following values and optimal decisions:

$$\begin{aligned} K_{1100} &= 30.683; & a_{1100} &= (2, 1); \\ K_{1101} &= 35.839; & a_{1101} &= (1, 2); \\ K_{1110} &= 31.491; & a_{1110} &= (2, 1); \\ K_{1111} &= 37.938; & a_{1111} &= (1, 2). \end{aligned}$$

The μc -rule in center 2 gives priority to queue 1. However, the optimal policy serves queue 2 first if center 1 is occupied. Hence, in this model also the optimal decision rule in center 2 depends on the state in center 1. Note that the optimal policy never idles. In the next section we will see that this is a consequence of the fact that $\tilde{c}_1 \geq c_1$ and $\tilde{c}_2 \geq c_2$.

2.6. Tandems of server assignment models with a single server

In this section we consider a tandem of two centers, each with m queues, a single server, and with arrivals according to an MAP at the first center. The service rates in queue j in the first center are $\tilde{\mu}_j$, in the second μ_j . Thus the counterexamples of the previous section are special cases of this model, with $m = 2$ and no arrivals. In the previous section it has been showed that (for suitable cost functions) the SIP is optimal in the second center if $\mu_1 \leq \dots \leq \mu_m$. No results were obtained on the optimal policy at the first center. In general the optimal policy in the first center depends on the state of the second center, even if the SIP is optimal in the second center, and is therefore hard to characterize.

In this section we first show monotonicity in both centers. In the case of linear costs, the conditions for the first center are that the costs must be higher in each class than the costs, for the same class, in the second center; for the second center the costs must be positive. This leads in the linear case to an adaptation of the μc -rule, for which we show the optimality in the heavy traffic case. With the help of calculations we investigate how this policy behaves for other values of the parameters.

We assume $\sum_y \lambda_{xy} = \gamma$ for all x and that $\gamma + \tilde{\mu} + \mu \leq 1$, where again $\mu = \max_j \mu_j$ and $\tilde{\mu} = \max_j \tilde{\mu}_j$. The dynamic programming equation is:

$$\begin{aligned} v_{(x, \tilde{i}, i)}^{n+1} = \min_{l, l} \Big\{ & \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y, \tilde{i}+e_j, i)}^n + (1 - \sum_{j=1}^m q_{xy}^j) v_{(y, \tilde{i}, i)}^n \right) + \\ & \tilde{\mu}_l v_{(x, \tilde{i}-e_l, i+e_l)}^n + (\tilde{\mu} - \tilde{\mu}_l) v_{(x, \tilde{i}, i)}^n + \\ & \mu_l v_{(x, \tilde{i}, i-e_l)}^n + (\mu - \mu_l) v_{(x, \tilde{i}, i)}^n + (1 - \gamma - \tilde{\mu} - \mu) v_{(x, \tilde{i}, i)}^n \Big\} = \\ & \sum_y \lambda_{xy} \left(\sum_{j=1}^m q_{xy}^j v_{(y, \tilde{i}+e_j, i)}^n + (1 - \sum_{j=1}^m q_{xy}^j) v_{(y, \tilde{i}, i)}^n \right) + \end{aligned}$$

$$\min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i}-e_j, i+e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\} + \quad (2.6.1)$$

$$\min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\} + (1 - \gamma - \tilde{\mu} - \mu) v_{(x, \tilde{i}, i)}^n.$$

The minimization ranges over all non-empty queues. Idleness corresponds to action 0 with $\tilde{\mu}_0 = \mu_0 = 0$. Now we prove monotonicity in both centers. It is easily seen that the monotonicity in the second center can also be proven in the more general case of an MDAP.

2.6.1. Lemma. *If*

$$w_{(x, \tilde{i}-e_{j_1}, i+e_{j_1})} \leq w_{(x, \tilde{i}, i)} \text{ for } \tilde{i}_{j_1} > 0 \quad (2.6.2)$$

and

$$w_{(x, \tilde{i}, i-e_{j_1})} \leq w_{(x, \tilde{i}, i)} \text{ for } i_{j_1} > 0 \quad (2.6.3)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof of this lemma can be found in chapter 4. We have the following:

2.6.2. Theorem. *The optimal policy at T is non-idling in both centers for all cost functions satisfying (2.6.2) and (2.6.3).*

Let us see what the inequalities mean for linear costs. Equation (2.6.3) requires that, as in the analysis in section 1.11, $c_j \geq 0$ for all j . It is easily seen that (2.6.2) requires $\tilde{c}_j - c_j \geq 0$ for all j . This is not surprising, as this number is the cost reduction when a class j customer moves from center 1 to 2. This gives us a conjecture on how an optimal policy might be: in center 1 serving the queue with highest $\tilde{\mu}_j(\tilde{c}_j - c_j)$ and in center 2 the μc -rule, i.e. serving the queue with highest $\mu_j c_j$. We call this policy the tandem μc -rule. However, this policy is not optimal due to problems when the second center is almost empty, meaning that not only cost reduction is important, but so is keeping the second server busy. Therefore we have the following lemma, in which it is assumed that there are enough customers in the second center.

2.6.3. Lemma. *Assume idleness is not allowed. If*

$$\begin{aligned} \tilde{\mu}_{j_1} w_{(x, \tilde{i}-e_{j_1}, i+e_{j_1})} + (\tilde{\mu} - \tilde{\mu}_{j_1}) w_{(x, \tilde{i}, i)} &\leq \tilde{\mu}_{j_2} w_{(x, \tilde{i}-e_{j_2}, i+e_{j_2})} + (\tilde{\mu} - \tilde{\mu}_{j_2}) w_{(x, \tilde{i}, i)} \\ \text{for } j_1 < j_2 \text{ and } \tilde{i}_{j_1}, \tilde{i}_{j_2} > 0 \text{ and } n \leq i_j \end{aligned} \quad (2.6.4)$$

and, for some \tilde{j} ,

$$\begin{aligned} \mu_{\tilde{j}} w_{(x, \tilde{i}, i-e_{\tilde{j}})} + (\mu - \mu_{\tilde{j}}) w_{(x, \tilde{i}, i)} &\leq \mu_{j_1} w_{(x, \tilde{i}, i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x, \tilde{i}, i)} \\ \text{for } i_{j_1} > 0 \text{ and } n \leq i_j \end{aligned} \quad (2.6.5)$$

hold for the cost function v^0 , then they hold for all v^n .

The proof can be found in chapter 4. Note that because queue \tilde{j} in center 2 is never empty, (2.6.5) is weaker than usual. Thus the SIP is optimal in the first center, if $i_{\tilde{j}}$ is large enough. In the second center, queue \tilde{j} has highest priority, and is always served because of the number of customers in the queue. The lemma is the basis of our heavy traffic theorem. The proof is included as the theorem does not follow from uniformization. We assume that no idleness is allowed.

2.6.4. Theorem. *For all T , cost functions satisfying (2.6.4) and (2.6.5) and $\varepsilon > 0$ there is a number N such that the tandem μc -rule in both centers is ε -optimal at T , if there are more than N customers in queue \tilde{j} at time 0.*

Proof. Let N_1 denote the fixed number of customers in the first center, at time 0. We compare the costs of two policies: the tandem μc -rule and the optimal policy R^* . Let the r.v. $\Phi_x^T(\mu c)$ and $\Phi_x^T(R^*)$ denote their costs, where x is the starting state of the whole system. We can use uniformization which gives us the possibility of conditioning on the number of jumps. If this number is smaller than N , then the expected costs under R^* are larger, by lemma 2.6.3. Let A_N denote the event that there are more than N jumps in $[0, T]$. Thus $\mathbb{E}(\Phi_x^T(\mu c)|A_N^c) - \mathbb{E}(\Phi_x^T(R^*)|A_N^c) \leq 0$. Then

$$\begin{aligned} & \mathbb{E}\Phi_x^T(\mu c) - \mathbb{E}\Phi_x^T(R^*) = \\ & \left(\mathbb{E}(\Phi_x^T(\mu c)|A_N) - \mathbb{E}(\Phi_x^T(R^*)|A_N) \right) \mathbb{P}(A_N) + \\ & \left(\mathbb{E}(\Phi_x^T(\mu c)|A_N^c) - \mathbb{E}(\Phi_x^T(R^*)|A_N^c) \right) \mathbb{P}(A_N^c) \leq \\ & \left(\mathbb{E}(\Phi_x^T(\mu c)|A_N) - \mathbb{E}(\Phi_x^T(R^*)|A_N) \right) \mathbb{P}(A_N). \end{aligned}$$

The expected number of arrivals, conditioned on A_N , is smaller than $N + T/\gamma$. Thus the expected number of customers available at T , conditioned on A_N , for both the tandem μc -rule and R^* , is smaller than $N_1 + 2N + T/\gamma$. The expected costs are bounded by $(N_1 + 2N + T/\gamma)c$ for some c . It remains to show that there is a N such that $\mathbb{P}(A_N)(N_1 + 2N + T/\gamma)c \leq \frac{\varepsilon}{2}$. This follows easily as $\mathbb{P}(A_N)$ and $N\mathbb{P}(A_N) \downarrow 0$ as $N \rightarrow \infty$. \square

Indeed, it is easily checked in the case of linear costs that the tandem μc -rule is optimal if $\tilde{\mu}_1(\tilde{c}_1 - c_1) \geq \dots \geq \tilde{\mu}_m(\tilde{c}_m - c_m)$ and $\mu_j c_j \geq \mu_{\tilde{j}} c_{\tilde{j}}$ for all j . If idleness is allowed, we can, as usual, combine lemma 2.6.3 and 2.6.1:

2.6.5. Theorem. *For all T , cost functions satisfying (2.6.4), (2.6.5), (2.6.2) and (2.6.3) and $\varepsilon > 0$ there is a number N such that the tandem μc -rule in*

both centers is ε -optimal at T , if there are more than N customers in queue \tilde{j} at time 0.

Now we restrict ourselves to $m = 2$ queues, and we assume that the tandem μc -rule has the same priority in both centers, i.e. serving queue 1 is optimal in both centers if there are enough customers. Then we do not need to assume that there are more than n customers in the first queue of the second center, instead it suffices to assume that there are, in total, more than n customers in the second center.

2.6.6. Lemma. *Assume idleness is not allowed. If*

$$\begin{aligned} \tilde{\mu}_1 w_{(x, \tilde{i}-e_1, i+e_1)} + (\tilde{\mu} - \tilde{\mu}_1) w_{(x, \tilde{i}, i)} &\leq \tilde{\mu}_2 w_{(x, \tilde{i}-e_2, i+e_2)} + (\tilde{\mu} - \tilde{\mu}_2) w_{(x, \tilde{i}, i)} \\ &\text{for } \tilde{i}_1, \tilde{i}_2 > 0 \text{ and } n \leq i_1 + i_2 \end{aligned} \quad (2.6.6)$$

and

$$\begin{aligned} \mu_1 w_{(x, \tilde{i}, i-e_1)} + (\mu - \mu_1) w_{(x, \tilde{i}, i)} &\leq \mu_2 w_{(x, \tilde{i}, i-e_2)} + (\mu - \mu_2) w_{(x, \tilde{i}, i)} \\ &\text{for } i_1, i_2 > 0 \text{ and } n \leq i_1 + i_2 \end{aligned} \quad (2.6.7)$$

hold for the cost function $w = v^0$, then they hold for all v^n .

The proof can be found in chapter 4. As in the previous case, we can show the following.

2.6.7. Theorem. *For all T , cost functions satisfying (2.6.6) and (2.6.7) (and (2.6.2) and (2.6.3) if idleness is allowed) and $\varepsilon > 0$ there is a number N such that the tandem μc -rule in both centers is ε -optimal at T , if there are more than N customers in the second center at time 0.*

An interesting question is how well the tandem μc -rule performs for other traffic than heavy traffic. We did some computations on the model of figure 2.6.1. The arrivals at both queues are Poisson with the same rate.

λ	center 1	$\tilde{\mu}$	\tilde{c}	center 2	μ	c
λ^*	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	1	4	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	2	1.1
λ^*	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	2	2	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	1	2

Figure 2.6.1

In table 2.6.1 the results for the discounted cost case are summarized. For all combinations we computed the relative difference between the costs under the optimal policy and under the μc -rule, for the starting states with each

queue empty (shown above) and 5 customers in each queue of both centers. Of course we had to make the state space finite. We did this by giving an upper bound on the total number of customers in the system. By doing it this way, buffer influences are relatively small. Note that the average load is equal to $\frac{3}{2}\lambda^*$, and thus $\lambda^* = 0.6$ gives an average load of 0.9.

	$\beta = 0.01$	0.1	0.25	0.5	0.75
$\lambda^* = 0.1$	0 0	0 0	0 0	0 0	$2.6 \cdot 10^{-8}$ $4.3 \cdot 10^{-6}$
0.2	0 0	0 0	0 0	0 0	$4.0 \cdot 10^{-7}$ $4.2 \cdot 10^{-6}$
0.3	0 0	0 0	0 0	0 0	$4.5 \cdot 10^{-7}$ $2.1 \cdot 10^{-6}$
0.4	0 0	0 0	0 0	0 0	0 $< 10^{-15}$
0.5	0 0	0 0	0 0	0 0	0 $< 10^{-15}$
0.6	0 0	0 0	0 0	0 $< 10^{-14}$	$< 10^{-14}$ $< 10^{-13}$

Table 2.6.1. Discounted costs

The results for the average cost case in table 2.6.2 indicate that theorem 2.6.7 does not hold for average costs. The results for high traffic intensities are less accurate (indicated with \approx) due to the finite state space, although we had a model with a maximum of 60 customers, giving more than $6 \cdot 10^5$ states. Note that not only the buffer influence, but also the relative differences in these models are larger than in the customer assignment models.

λ^*	R^*	μc	rel. diff.
0.1	0.886	0.889	$3.4 \cdot 10^{-3}$
0.2	2.134	2.171	$1.7 \cdot 10^{-2}$
0.3	4.024	4.202	$4.4 \cdot 10^{-2}$
0.4	7.248	≈ 7.939	$9.5 \cdot 10^{-2}$
0.5	≈ 14.092	≈ 16.862	$2.0 \cdot 10^{-1}$
0.6	≈ 36.6	≈ 48.5	$3.2 \cdot 10^{-1}$

Table 2.6.2. Average costs

The results of this section are also published in [26].

2.7. Tandems of server assignment models with a single server and identical centers

Here we consider the special case $\tilde{\mu} = \mu$. Consider recurrence relation (2.6.1). Instead of inequalities we consider equalities here.

2.7.1. Lemma. *Assume idleness is not allowed. If*

$$\mu_{j_1} w_{(x, \tilde{i}-e_{j_1}, i+e_{j_1})} + (\mu - \mu_{j_1}) w_{(x, \tilde{i}, i)} = \mu_{j_2} w_{(x, \tilde{i}-e_{j_2}, i+e_{j_2})} + (\mu - \mu_{j_2}) w_{(x, \tilde{i}, i)} \quad \text{for } j_1 < j_2 \text{ and } \tilde{i}_{j_1}, \tilde{i}_{j_2} > 0, \quad (2.7.1)$$

$$\mu_{j_1} w_{(x, \tilde{i}, i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x, \tilde{i}, i)} = \mu_{j_2} w_{(x, \tilde{i}, i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x, \tilde{i}, i)} \quad \text{for } j_1 < j_2 \text{ and } i_{j_1}, i_{j_2} > 0, \quad (2.7.2)$$

and

$$\begin{aligned} \mu_{j_1}^2 w_{(x, \tilde{i}-e_{j_1}, 0)} + \mu_{j_1} (\mu - \mu_{j_1}) w_{(x, \tilde{i}-e_{j_1}, e_{j_1})} + (\mu - \mu_{j_1}) \mu w_{(x, \tilde{i}, 0)} = \\ \mu_{j_2}^2 w_{(x, \tilde{i}-e_{j_2}, 0)} + \mu_{j_2} (\mu - \mu_{j_2}) w_{(x, \tilde{i}-e_{j_2}, e_{j_2})} + (\mu - \mu_{j_2}) \mu w_{(x, \tilde{i}, 0)} \end{aligned} \quad \text{for } j_1 < j_2 \text{ and } \tilde{i}_{j_1}, \tilde{i}_{j_2} > 0 \quad (2.7.3)$$

hold for the cost function v^0 , then they hold for all v^n .

Equation (2.7.1) and (2.7.2) give, for allowable cost functions, the optimality of all possible policies: (2.7.1) shows that serving queue j_1 or queue j_2 in center 1 makes no difference. Similarly, (2.7.2) shows that serving any queue in center 2 is optimal. Equation (2.7.3) is needed in the proof of (2.7.1). The proof of the lemma can be found in chapter 4.

Now consider allowable cost functions. The only interesting ones we could find are both $\mathbf{I}\{|\tilde{i}| + |i| = 0\}$ and $\mathbf{I}\{|\tilde{i}| + |i| > 0\}$. This means the following.

2.7.2. Theorem. *For every non-idling policy the probability that there are customers present at T is the same.*

Changing the system as in section 1.12 gives:

2.7.3. Corollary. *The distribution of the length of the busy period is equal for all non-idling policies.*

Heuristically, we can say the following. If the policy in center 1 does not depend on center 2, the arrivals at center 2 are independent (according to an MAP) and it is clear that every policy in center 2 minimizes the makespan. Thus, a possible explanation of theorem 2.7.3 centers around the first center. If there is enough work at center 2, again the policy does not matter. However, in case the server at center 2 has little work, there are 2 possibilities. The first is to serve a fast customer in center 1, giving the server at center 2 work as soon

as possible. However, the amount of work is small. When a slow customer is served in center 1 the situation is reversed: it takes long for the work to arrive, but the amount of work is large. Apparently, the two phenomena balance each other.

As usual in single server models, we can combine equation (2.7.1) with (2.7.3), (2.6.2) and (2.6.3) when idleness is allowed. Note that, by (2.6.3), $\mathbf{I}\{|z| + |z| = 0\}$ is not a valid cost function anymore. Therefore we have:

2.7.4. Corollary. *The length of the busy period is stochastically identical under all work-conserving policies, if both centers have equal service rates and idleness is allowed.*

Remark. When each queue in center one has one customer initially present, and no arrivals occur, and if the SIP is employed in both centers, then we can think of the servers as going from queue to queue instead of the customers going from center to center. Using this equivalence (which was pointed out to me by Rhonda Righter, and which can be found in Pinedo & Schrage [55, p. 190]), and using corollary 2.7.4, we see that reordering of the queues in a system where the service rate depends only on the server has no effect on the makespan. This interchangeability of $\cdot|M|1$ queues is well known, see Weber [77] for references. Note that the equivalence is only valid under certain restrictions on the model. Similarly, the interchangeability is proven for a more general model. Therefore the results on both models are of independent interest.

It appears that lemma 2.7.1 cannot be generalized easily to inequalities, although we conjecture that a similar lemma with equalities replaced by inequalities holds. We give some numerical results supporting this conjecture, with $m = 2$, Poisson arrivals and $\tilde{c} = c$. By scaling we can fix $\mu_2 = 1$ and $c_1 = 1$, giving the parameters as in figure 2.7.1.

λ	center 1	μ	c	center 2	μ	c
λ^*	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	μ^*	1	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	μ^*	1
λ^*	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	1	c^*	<div style="border: 1px solid black; width: 80px; height: 20px;"></div>	1	c^*

Figure 2.7.1

With value iteration we computed the average costs for the optimal policy, the policy that gives priority to queue 1 (R_1), and to the policy that gives priority to queue 2 in both centers (R_2). It appeared that for low values of λ^* the differences are most significant. Because of the computational method we had to introduce a number B equal to the maximum number of customers in the system. For $B = 25$ there was no influence from the buffer (when we took λ^* small), meaning that $B = 30, 35$ and 40 gave the same results. Taking $\mu^* = 2$ appeared to be satisfactory. We took $\lambda^* = 0.25$, giving an average

workload of 0.375. In figure 2.7.2 the values of the different policies can be seen for various values of c^* .

For $c^* \leq 2$, R_1 appeared to be optimal. If $c^* \geq 2.66$, then R_2 is optimal. For $2 < c^* \leq 2.66$ the optimal policy is neither R_1 nor R_2 . The number 2 can easily be explained: below 2 R_1 is both faster and costs less. The value 2.66 is explained as follows. When there are no arrivals, the total costs can be computed. It appears that, for general μ^* , the optimal action in $(1, 1, 0, 0)$ is queue 1 if $c^* \leq 2\mu^{*2}/(1 + \mu^*)$ and queue 2 if $c^* \geq 2\mu^{*2}/(1 + \mu^*)$. For $\mu^* = 2$ this number is indeed equal to $8/3$. Computations show that $2\mu^{*2}/(1 + \mu^*)$ is the turn-over point for various μ^* . This indicates that

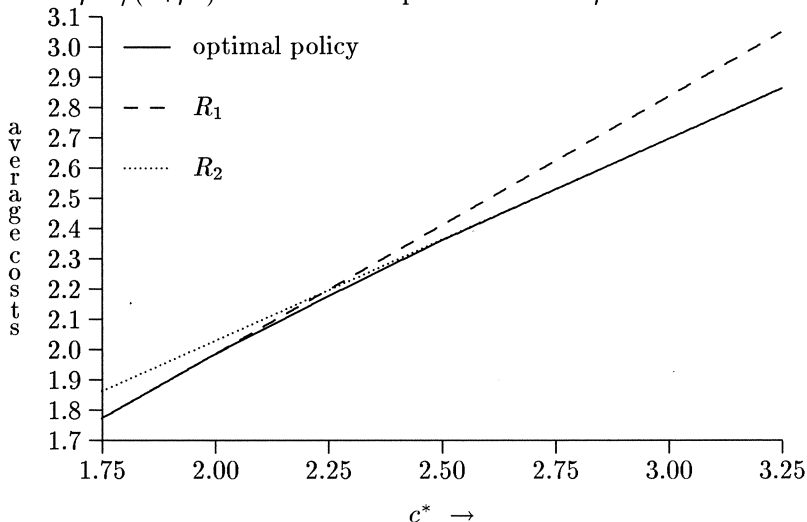


Figure 2.7.2

$(1, 1, 0, 0)$, the only state with 2 customers in which the action is non-trivial, plays an important role in this model.

Chapter 3

Models with Dependent Markov Decision Arrival Processes

3.1. Dependent Markov Decision Arrival Processes

In some models we can generalize the arrival process even more, by letting the arrival probabilities depend on the state of the queues.

3.1.1. Definition. (Dependent Markov Decision Arrival Process) Let Λ be the countable state space of a Markov decision process with transition intensities λ_{xay} with $x, y \in \Lambda$ and $a \in A(x)$, the set of actions in x . When this process moves from x to y , while action a was chosen and the state of the queues is i , then with probability $q_{xay,i}^k$ an arrival in class $1 \leq k \leq m$ occurs, and with probability q_{xay}^{n+k} an event with server $1 \leq k \leq s$ occurs. There are sets $\Lambda_1^s, \dots, \Lambda_s^s$ such that server k is available if and only if $x \in \Lambda_k^s$, and sets $\Lambda_{1n}^a, \dots, \Lambda_{mn}^a$, $n \in \mathbb{N}$, such that if $x \in \Lambda_{kn}^a$ then there have been n or more arrivals of class k . We call the quadruple (Λ, A, λ, q) a Dependent Markov Decision Arrival Process (DMDAP).

Naturally, we can also let the transition rates and the probabilities of server events depend on the state of the queues i . Because we do not study models where this is the case we did not allow this type of dependency.

How the arrival probabilities are allowed to depend on i will be specified for each model. Note that if there is no dependency, we have an MDAP. It is clear that conditions on the DMDAP must be given; to give an example where the optimality result does not hold, assume that in the customer assignment model of section 1.2 the arrival probabilities are higher in more balanced states than in unbalanced states. Then assigning to the longer queue might be more favorable.

3.2. Asymmetric customer assignment model

In this section we deal with a customer assignment model with asymmetric service rates. In the sections 3.3 and 3.4 we will see that the results of section 1.2 and 1.3 are special cases of the result proved here. The present result gives only a partial characterization of the optimal policy in the general model, even if the arrivals are non-controlled. We discuss some computational results at the end of the section.

The model is as follows. Customers arrive according to a DMDAP, all in one class (write q instead of q^1). There are no server vacations. Arriving customers have to be assigned to one of the non-full queues, where B are the buffer sizes. In state (x, i) , a customer in queue j is served with rate μ_{ji} . We end the description by giving conditions on the arrival probabilities and the service rates.

To make the notation shorter (although it abuses notational conventions a bit), let i^* be the permutation of i with i_{j_1} and i_{j_2} switched, that is, $i_j^* = i_j$ if $j \neq j_1, j_2$, $i_{j_1}^* = i_{j_2}$ and $i_{j_2}^* = i_{j_1}$. Assume all vectors considered are componentwise smaller than B . Now we formulate the conditions on the arrival probabilities and the departure rates.

The q must satisfy the following conditions:

$$q_{xay; i+e_{j_1}} \leq q_{xay; i+e_{j_2}} \text{ if } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2 \quad (3.2.1)$$

$$q_{xay; i} \leq q_{xay; i^*} \text{ if } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2 \quad (3.2.2)$$

An interesting example which satisfies the conditions is a DMDAP with $\Lambda = \{1\}$, $A(1) = \{1\}$, $\lambda_{111} = \tilde{\lambda}$ and $q_{111; i} = (N - |i|)/N$, the well known finite source model. In fact, if $q_{xay; i}$ only depends on $|i|$ (and x, a and y), every other dependency is allowed.

The μ satisfy the following:

$$\mu_{ji} = 0 \text{ if } i_j = 0$$

$$\mu_{ji+e_{j_1}} \geq \mu_{ji+e_{j_2}} \text{ if } j \neq j_1, j_2, i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} \geq \mu_{j_1 i+e_{j_2}} + \mu_{j_2 i+e_{j_2}} \text{ if } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{ji} \geq \mu_{ji+e_{j_1}} \text{ if } j \neq j_1$$

$$\mu_{ji} \geq \mu_{ji^*} \text{ if } j \neq j_1, j_2, i_{j_1} > i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{j_1 i} \geq \mu_{j_2 i^*} \text{ if } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{j_1 i} + \mu_{j_2 i} \geq \mu_{j_1 i^*} + \mu_{j_2 i^*} \text{ if } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2$$

We also assume that $\mu_{ji} \leq \mu$ for some constant μ . An interesting example is $\mu_{ji} = \min\{i_j, s_j\}\tilde{\mu}_j$, with $s_1 \geq \dots \geq s_m$ and $\tilde{\mu}_1 \geq \dots \geq \tilde{\mu}_m$. Thus lower numbered queues have more and faster working servers.

Another example is the following. Assume that customers which are not served require a certain amount of attention which decreases the service rate

of the customer being served. The amount of attention needed depends on the queue. This results in $\mu_{ji} = (1 - p_j i_j) \tilde{\mu}$, $p_1 \leq \dots \leq p_m$. We assume that B is such that $p_j B_j \leq 1$ for all j . Note that the service rate at queue j is decreasing in i_j , an at first sight counterintuitive fact. This μ_{ji} satisfies the conditions.

Again we assume that $\gamma + m\mu \leq 1$, where $\sum_y \lambda_{xay} = \gamma$ for all x and a . The dynamic programming equation is:

$$v_{(x,i)}^{n+1} = c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay,i} \min_j \{ v_{(y,i+e_j \wedge B)}^n \} + (1 - q_{xay,i}) v_{(y,i)}^n \right) \right\} + \sum_{j=1}^m \mu_{ji} v_{(x,i-e_j)}^n + (1 - \sum_{j=1}^m \mu_{ji} - \gamma) v_{(x,i)}^n. \quad (3.2.3)$$

3.2.1. Lemma. *If*

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \quad \text{for } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2, \quad (3.2.4)$$

$$w_{(x,i)} \leq w_{(x,i+e_{j_1})} \quad (3.2.5)$$

and

$$w_{(x,i)} \leq w_{(x,i^*)} \quad \text{for } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2 \quad (3.2.6)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The proof of lemma 3.2.1 can be found in chapter 4. Recall that we assumed that all vectors considered are smaller than B . Equation (3.2.4) gives a partial characterization of the optimal policy. It says that an arriving customer should be assigned to queue j_1 instead of queue j_2 if there are less customers in queue j_1 and if $j_1 < j_2$. Usually, this does not specify the optimal policy completely. Therefore we called the characterization partial. Note that sending the customer to queue j_1 gives a higher total service rate, and in the case $\mu_{ji} = \tilde{\mu}_j$ with $\tilde{\mu}_1 \geq \dots \geq \tilde{\mu}_m$, the customer is sent to the faster queue. Therefore we call such a policy a *Shorter Faster Queue Policy* (SFQP). Equation (3.2.6) is needed to prove equation (3.2.4). Equation (3.2.5) is the well known monotonicity. Using corollary 5.3.4, we have the following.

3.2.2. Theorem. *For all T , an SFQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.2.4) to (3.2.6).*

A special case of this result is proven in [24].

The conditions (3.2.4) to (3.2.6) are weaker than (1.2.2) to (1.2.4), meaning that all Schur convex cost functions are allowable. It is easy to give non-Schur convex functions that are allowable (for example, $v_{(x,i)}^0 = \sum_{j=1}^m c_j i_j$ with $0 \leq c_1 < \dots < c_m$), meaning that the class of allowable functions is strictly bigger. In the present case however, we were not able to give a complete characterization of all allowable cost functions, although we have a conjecture, stated in appendix C. Note that, for reasons explained in section 2.2, there are no stochastic results.

For the class of non-symmetric additive cost functions we have a sufficient condition. We consider cost functions c which only depend on i , because the dependence on x can be arbitrary. We consider $c_{(x,i)} = f_1(i_1) + \dots + f_m(i_m)$. Define $\Delta f_j(i) = f_j(i+1) - f_j(i)$. Then the following conditions are sufficient: f_j increasing, $\Delta f_1(i) \leq \dots \leq \Delta f_m(i)$ for all i and $m-1$ of the m functions are convex. Since of any two functions one is convex, either $\Delta f_{j_1}(i_{j_1}) \leq \Delta f_{j_1}(i_{j_2}) \leq \Delta f_{j_2}(i_{j_2})$ or $\Delta f_{j_1}(i_{j_1}) \leq \Delta f_{j_2}(i_{j_1}) \leq \Delta f_{j_2}(i_{j_2})$ holds if $j_1 < j_2$ and $i_{j_1} \leq i_{j_2}$, and (3.2.4) follows. Equation (3.2.5) is immediate, and (3.2.6) follows because $\Delta f_{j_1}(i) \leq \Delta f_{j_2}(i)$ for all i , and thus $f_{j_1}(i_{j_1}) - f_{j_1}(i_{j_2}) \leq f_{j_2}(i_{j_1}) - f_{j_2}(i_{j_2})$.

Even in the case of an MAP, the optimal policy is not myopic. Consider the following simple model with Poisson arrivals, $m = 2$, $\mu_2 \ll \mu_1$, $B = (\infty, \infty)$ and $v_{(i_1, i_2)}^0 = i_1 + i_2$. Now consider $v_{(1,0)}^2$ and $v_{(1,0)}^3$. No matter how small μ_2 is, if $n = 2$ action 2 is optimal because, if there is an arrival, there is at most 1 service completion before the planning horizon. If $n = 3$ however, it is possible that queue 1 is served twice before $n = 0$, and we can choose the parameters such that action 1 is optimal.

Also in the continuous-time case (and again independent arrivals), there is no unique optimal policy. However, for the model with Poisson arrivals and a single server at each queue (i.e., $\mu_{ji} = \min\{i_j, 1\}\tilde{\mu}_j$) attempts have been made to describe the optimal policy in more detail. Theoretically it has been shown by Hajek [19] that the optimal policy is monotone, meaning that there is an increasing switching curve, and it has been shown by Katehakis & Levine [32] that for an arrival rate which is sufficiently small the policy that assigns to the queue with smallest expected workload is optimal.

In the papers Van Moorsel & De Vries [47], Nobel & Tijms [54], Houck [30] and Shenker & Weinreb [65] computational results are obtained, mostly for $m = 2$ and $B_1 = B_2$. Van Moorsel & De Vries [47] and Nobel & Tijms [54] use successive approximation, in the other two papers simulation is used. Nearly optimal policies are proposed, for example the policy that assigns each arriving customer to the queue where its expected delay is minimal. It is clear that successive approximation is a better method than simulation, because with simulation a policy cannot be compared with the optimal one, and because simulation is computationally less attractive. (Note that this contradicts a remark by Shenker & Weinreb [65], where it is stated that, using methods from Markov decision theory, it is difficult to find the optimal policy “even in the smallest non-trivial case of just two non-identical servers”. In the previous chapter we had no problems finding optimal policies in models with 4 queues, with an accuracy which is hard to obtain with simulation.) All policies studied in the cited papers are SFQP’s. Nobel & Tijms [54] also consider the case where there is more than one server in each queue, i.e. the case $\mu_{ji} = \min\{i_j, s_j\}\tilde{\mu}_j$.

3.3. Symmetric customer assignment models

In this section we first analyze the symmetric model using lemma 3.2.1. Then we generalize this model by introducing batch arrivals. By a limiting argument we obtain results for models with workloads. Unfortunately we cannot allow finite buffers in this model. In the third model of this section we allow non-routable arrivals at the queues, and we introduce an extra movable server, as Menich & Serfozo [46] did. All parameters are allowed to depend on the whole state of the system, with conditions as general as possible. If we take an arrival process without actions and with one state, we have the model of Menich & Serfozo [46].

To start with the first model, we modify the conditions of the previous section as follows. Assume again that all vectors considered are componentwise smaller than B .

The q satisfy the following:

$$\begin{aligned} q_{xay;i+e_{j_1}} &\leq q_{xay;i+e_{j_2}} \text{ if } i_{j_1} \leq i_{j_2} \\ q_{xay;i} &= q_{xay;i^*} \text{ for all } j_1, j_2 \end{aligned} \quad (3.3.1)$$

Recall that i^* agrees with i except for j_1 and j_2 being interchanged. The last condition is called symmetry. Note that the finite source model satisfies these conditions.

Also μ is made symmetric:

$$\begin{aligned} \mu_{ji} &= 0 \text{ if } i_j = 0 \\ \mu_{ji+e_{j_1}} &\geq \mu_{ji+e_{j_2}} \text{ if } j \neq j_1, j_2 \text{ and } i_{j_1} \leq i_{j_2} \\ \mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} &\geq \mu_{j_1 i+e_{j_2}} + \mu_{j_2 i+e_{j_2}} \text{ if } i_{j_1} \leq i_{j_2} \\ \mu_{ji} &\geq \mu_{ji+e_{j_1}} \text{ if } j \neq j_1 \\ \mu_{ji} &= \mu_{ji^*} \text{ if } j \neq j_1, j_2 \\ \mu_{j_1 i} &= \mu_{j_2 i^*} \text{ and } \mu_{j_2 i} = \mu_{j_1 i^*} \end{aligned}$$

The symmetric versions of the examples of the previous section, $\mu_{ji} = \min\{i_j, s\}\tilde{\mu}$ and $\mu_{ji} = (1 - pi_j)\tilde{\mu}$ for suitable B , are allowed here.

The present model is general enough to capture that of Johri [31]. There Poisson arrivals are taken, together with the following assumptions on the service rates: $\mu_{ji} = \mu_{\tilde{j}i}$ if $i_j = \tilde{i}_j$, $\mu_{ji} \leq \mu_{ji+e_j}$ and $\mu_{ji+2e_j} - \mu_{ji+e_j} \leq \mu_{ji+e_j} - \mu_{ji}$, i.e. the service rate in a queue depends only on the number of customers in that queue, and is both increasing and concave. For example, the model with multiple servers at each queue conforms to this description.

The dynamic programming equation remains the same as in the previous section. The conditions are stronger than those in section 3.2, giving the validity of lemma 3.2.1 for the model studied here.

We can obtain the optimality result for the symmetric case from lemma 3.2.1. Let Π be a permutation matrix. Assume that v^0 and c are symmetric in i , i.e. $v_{(x,i)}^0 = v_{(x,i\Pi)}^0$ and $c_{(x,i)} = c_{(x,i\Pi)}$.

3.3.1. Lemma. Assume we have vectors B and $\tilde{B} = B\Pi$, a permutation of B . Let v^n and \tilde{v}^n be value function for identical models, except for the buffer sizes, being B and \tilde{B} . Then $v^n_{(x,i)} = \tilde{v}^n_{(x,\tilde{i})}$ with $\tilde{i} = i\Pi$ for all n .

As the arrival and departure rates are symmetric the inductive proof is trivial.

Now consider equation (3.2.4). By exchanging queue j_1 and queue j_2 in the ordering we have the reversed inequality. By doing the same with (3.2.6) we have rewritten the set of inequalities, giving the following.

3.3.2. Corollary. If

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \text{ for } i_{j_1} \leq i_{j_2}, \quad (3.3.2)$$

$$w_{(x,i)} \leq w_{(x,i+e_j)} \quad (3.3.3)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \quad (3.3.4)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The equations (3.3.2) to (3.3.4) are the same as (1.2.2) to (1.2.4) and (2.2.2) to (2.2.4). Because the MAP and the MDAP are both special cases of the DMDAP, and because $\mu_{ji} = \min\{i_j, 1\}\tilde{\mu}$ satisfies the conditions, lemma 1.2.1 and 2.2.1 follow.

3.3.3. Theorem. For all T , an SQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.3.2) to (3.3.4).

Again, all Schur convex function are allowable cost functions.

In the second model of this section we want to generalize the results of section 1.8 to arrivals according to a DMDAP. If we want to do this straightforwardly, then we would have to generalize the uniformization results of chapter 5 to include the model here, for example generalizing the countable state space to \mathbb{R}^m . Instead of this, we show that the workload model is the limiting case of a queue length model with batch arrivals, for which the SQP is optimal. Assume that each batch consists with probability β_k of k customers. It is essential that the whole batch is assigned to the same queue. If we want to model batch arrivals where each member of a batch can be assigned to another queue, we can simply use the model without batch arrivals and regard the model with batch arrivals as a limiting case. We consider the simple case of each queue having a single server. Because the size of the batch can be arbitrarily large, blocking can always occur in the case of finite buffers, therefore we do not model them. The DMDAP has the same conditions as in the previous model.

The dynamic programming equation is:

$$v_{(x,i)}^{n+1} = c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay;i} \min_j \left\{ \sum_k \beta_k v_{(y,i+ke_j)}^n \right\} + (1 - q_{xay;i}) v_{(y,i)}^n \right) \right\} + \sum_{j=1}^m \mu v_{(x,(i-e_j)+)}^n + (1 - m\mu - \gamma) v_{(x,i)}^n.$$

3.3.4. Lemma. *If*

$$\sum_k \beta_k w_{(x,i+ke_{j_1})} \leq \sum_k \beta_k w_{(x,i+ke_{j_2})} \text{ for } i_{j_1} \leq i_{j_2} \quad (3.3.5)$$

$$w_{(x,i)} \leq w_{(x,i+e_j)} \quad (3.3.6)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \quad (3.3.7)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The proof can be found in chapter 4. Note that equation (3.3.5) is not valid without the summation, for the same reason as that (1.8.2) was not valid without the integration. Using corollary 5.3.4, we get the following.

3.3.5. Theorem. *For all T , an SQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.3.5) to (3.3.7).*

Again, all Schur convex cost functions satisfy the conditions.

By lemma A.2 we can approximate any service time distribution arbitrarily closely by phase-type distributions. Assigning to the shortest queue is, in the limit, equivalent to assigning to the queue with the shortest workload. This gives the following result.

3.3.6. Theorem. *For all T , an SWP minimizes the costs at T (from 0 to T) for all Schur convex cost functions.*

Although the arrival process is a DMDAP with which we can model a finite source, we cannot model a finite source in the workload model, because we do not know the actual number of customers in the system. Note however that we already proved in theorem 2.4.2 that the SWP is optimal in a finite source model.

Now we look at the model that has additional non-routable arrival streams, and an extra movable processor. The combination of finite buffers and additional arrivals is not allowed as the SQP might not be optimal anymore. This can be seen from the following example: take $m = 2$, $B = (3, \infty)$. In state

(2, 1) it may be optimal to assign an arriving customer in the assignable stream to queue 1, because then future non-routable arrivals at queue 1 are blocked.

The extra arrival streams can easily be modeled with the DMDAP: arrivals in class 0 are routable, customers arriving in class k , $1 \leq k \leq m$, join queue k . The arrival probabilities of class 0 are allowed to depend on the assignment action, i.e. we have arrival probabilities $q_{xay;ij}^0$, where j is the assignment. Assume that there are numbers \tilde{q}_{xay} such that $q_{xay;ij}^0 \leq \tilde{q}_{xay}$ for all i and j . We let the service probabilities depend also on x . Denote the service rate of the movable processor with $\tilde{\mu}_{ji;x}$, if it serves queue j in state i . Assume also that $\tilde{\mu}_{ji;x} \leq \tilde{\mu}$ for all i, j and x . The dynamic programming equation is:

$$\begin{aligned}
 v_{(x,i)}^{n+1} = & c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{xay} \left(\min_j \left\{ q_{xay;ij}^0 v_{(y,i+e_j)}^n + (\tilde{q}_{xay} - q_{xay;ij}^0) v_{(y,i)}^n \right\} + \right. \right. \\
 & \left. \left. \sum_{j=1}^m q_{xay;i}^j v_{(y,i+e_j)}^n + (1 - \sum_{j=1}^m q_{xay;i}^j - \tilde{q}_{xay}) v_{(y,i)}^n \right) \right\} + \\
 & \min_j \{ \tilde{\mu}_{ji;x} v_{(x,i-e_j)}^n + (\tilde{\mu} - \tilde{\mu}_{ji;x}) v_{(x,i)}^n \} + \\
 & \sum_{j=1}^m \mu_{ji;x} v_{(x,i-e_j)}^n + (1 - \gamma - \tilde{\mu} - \sum_{j=1}^m \mu_{ji;x}) v_{(x,i)}^n.
 \end{aligned} \tag{3.3.8}$$

Now we give the conditions. Recall that i^* is the vector equal to i , but with queue j_1 and j_2 interchanged. First we have symmetry of all parameters involved (called interchangeability in Menich & Serfozo [46]):

$$\begin{aligned}
 q_{xay;ij}^0 &= q_{xay;i^*j}^0 \text{ if } j \neq j_1, j_2 \text{ and } q_{xay;ij_1}^0 = q_{xay;i^*j_2}^0 \\
 q_{xay;i}^j &= q_{xay;i^*}^j \text{ if } j \neq j_1, j_2 \text{ and } q_{xay;i}^{j_1} = q_{xay;i^*}^{j_2} \\
 \tilde{\mu}_{ji;x} &= \tilde{\mu}_{ji^*;x} \text{ if } j \neq j_1, j_2 \text{ and } \tilde{\mu}_{j_1 i;x} = \tilde{\mu}_{j_2 i^*;x} \\
 \mu_{ji;x} &= \mu_{ji^*;x} \text{ if } j \neq j_1, j_2 \text{ and } \mu_{j_1 i;x} = \mu_{j_2 i^*;x}
 \end{aligned}$$

We also assume the following on q^0 (with, as in section 1.5, (j) the index of the j th smallest component of i):

$$q_{xay;i(1)}^0 \leq q_{xay;ij}^0 \tag{3.3.9}$$

$$q_{xay;i+e_{j_1}(1)}^0 \leq q_{xay;i+e_{j_2}(1)}^0 \text{ if } i_{j_1} \leq i_{j_2} \tag{3.3.10}$$

Assume $l_1 < l_2$. The conditions on the non-routable arrival probabilities are:

$$\sum_{j=k}^m q_{xay;i+e_{(l_1)}}^{(j)} \leq \sum_{j=k}^m q_{xay;i+e_{(l_2)}}^{(j)} \text{ for } k = 1, \dots, l_1, l_2 + 1, \dots, m \tag{3.3.11}$$

$$\sum_{j=k}^m q_{xay;i}^{(j)} \leq \sum_{j=k}^m q_{xay;i+e_{(l_1)}}^{(j)} \text{ for } k = l_1 + 1, \dots, m \quad (3.3.12)$$

Concerning servers, we assume, besides the interchangeability:

$$\tilde{\mu}_{ji;x} = \mu_{ji;x} = 0 \text{ if } i_j = 0$$

The assumptions on the service rate of the routable server are the reverse of those on q^0 :

$$\begin{aligned} \tilde{\mu}_{(m)i;x} &\geq \tilde{\mu}_{ji;x} \\ \tilde{\mu}_{(m)i+e_{j_1};x} &\geq \tilde{\mu}_{(m)i+e_{j_2};x} \text{ if } i_{j_1} \leq i_{j_2} \end{aligned} \quad (3.3.13)$$

The assumptions on the fixed servers are much like those on the non-routable arrivals:

$$\begin{aligned} \sum_{j=k}^m \mu_{(j)i+e_{(l_1)};x} &\geq \sum_{j=k}^m \mu_{(j)i+e_{(l_2)};x} \text{ for } k = 1, \dots, l_1, l_2 + 1, \dots, m \\ \sum_{j=k}^m \mu_{(j)i;x} &\geq \sum_{j=k}^m \mu_{(j)i+e_{(l_1)};x} \text{ for } k = l_1 + 1, \dots, m \end{aligned}$$

Note that the conditions are the same as in Menich & Serfozo [46]. Now we can formulate our inductive result.

3.3.7. Lemma. *If*

$$w_{(x,i+e_{j_1})} \leq w_{(x,i+e_{j_2})} \text{ for } i_{j_1} \leq i_{j_2}, \quad (3.3.14)$$

$$w_{(x,i)} \leq w_{(x,i+e_{j_1})}, \quad (3.3.15)$$

and

$$w_{(x,i)} = w_{(x,i^*)} \quad (3.3.16)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

3.3.8. Theorem. *For all T , an SQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.3.14) to (3.3.16).*

By appendix C the class of allowable cost functions is the class of weak Schur convex functions.

Remark. As we saw, the arrival probabilities and service rates are allowed to depend both on the state of the arrival process and the state of the queues. Therefore the term *environment* instead of arrival process would be more appropriate. Typically, in an environment the arrivals are according to a Markov Modulated Poisson Process. Here however, we kept the arrivals occurring at the transitions of the environment, in order to maintain the generality of the arrivals. In most other models studied in this thesis we can allow the service rates to depend on the state of the arrival process. Because the generalization is only minor and because of notational simplicity we refrained from doing so.

3.4. Customer assignment models without waiting room

The results of section 3.2 can also be used to obtain results in the model without waiting room, much like the results on the symmetric model were obtained in the previous section. We have the same condition on the arrival probabilities and the service rates, but because $B = (1, \dots, 1)$ the inequalities of lemma 3.2.1 simplify to

$$w_{(x, i+e_{j_1})} \leq w_{(x, i+e_{j_2})} \text{ for } j_1 < j_2 \text{ and } i_{j_1} = i_{j_2} = 0 \quad (3.4.1)$$

and

$$w_{(x, i)} \leq w_{(x, i+e_{j_1})} \text{ for } i_{j_1} = 0. \quad (3.4.2)$$

These are the same inequalities as (1.3.2) and (1.3.3). Lemma 1.3.1 follows because $\mu_{ji} = \min\{i_j, 1\}\bar{\mu}_j$ satisfies the conditions of lemma 3.2.1.

3.4.1. Theorem. *For all T , an FQP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.4.1) to (3.4.2).*

The class of allowable cost functions are the functions that respect the partial sum ordering, as introduced and discussed in appendix C.

In Sobel [66] a model is studied in which customers of \tilde{m} classes arrive according to independent Poisson processes. Besides that, the model is similar to the model studied here. The analysis of that paper appears to be erroneous. (The basic theorem 1 does not hold as the derivation of $B \geq 1$ is incorrect.) Sobel & Srivastava [67] wrote a revision. The model is essentially a single class model. The optimal policy does not depend on the class, and the only place the class of a customer plays a role is in the cost function. However, no example is given of a cost function that indeed depends on the customer classes. Here we prefer to study a more complex model in which rejection is allowed.

Consider m exponential servers with decreasing service rates $\mu_1 \geq \dots \geq \mu_m$ and arrivals according to an MAP. (At the end of this section we show that the results cannot be generalized to (D)MDAPs.) Arrivals occur in \tilde{m} classes. When a class k customer arrives, it can either be rejected or sent to one of the free servers. The service times depend only on the server, not on the customer class. When a class k customer is rejected blocking costs b_k are incurred, $b_1 \geq \dots \geq b_{\tilde{m}} \geq 0$. (It can be shown that if a class has negative blocking costs, it will always be blocked.) At the servers, an action has to be chosen for each class of customers. We denote with a_k the free server to which an arrival in class k is assigned, with action 0 corresponding to blocking. The dynamic programming equation becomes (assume $e_0 = 0$):

$$v_{(x, i)}^{n+1} = \sum_y \lambda_{xy} \left(\sum_{k=1}^{\tilde{m}} q_{xy}^k \min_{a_k} \left\{ \mathbf{I}\{a_k = 0\} b_k + v_{(y, i+e_{a_k})}^n \right\} + \right. \\ \left. (1 - \sum_{k=1}^{\tilde{m}} q_{xy}^k) v_{(y, i)}^n \right) +$$

$$\sum_{j=1}^m \mu_j v_{(x, (i-e_j)^+)}^n + (1 - \gamma - \sum_{j=1}^m \mu_j) v_{(x, i)}^n.$$

3.4.2. Lemma. *If $v_{(x, i)}^0 = 0$ for all x and i , then the following equations hold for all n :*

$$v_{(x, i+e_{j_1})}^n \leq v_{(x, i+e_{j_2})}^n \text{ for } j_1 < j_2 \text{ and } i_{j_1} = i_{j_2} = 0 \quad (3.4.3)$$

$$v_{(x, i)}^n \leq v_{(x, i+e_{j_1})}^n \text{ for } i_{j_1} = 0 \quad (3.4.4)$$

$$v_{(x, i+e_{j_1})}^n \leq b_1 + v_{(x, i)}^n \text{ for } i_{j_1} = 0 \quad (3.4.5)$$

$$v_{(x, i+e_{j_1})}^n - v_{(x, i)}^n \leq v_{(x, i+e_{j_2}+e_{j_1})}^n - v_{(x, i+e_{j_2})}^n \quad (3.4.6)$$

for $j_1 = \min\{j | (i + e_{j_2})_j = 0\}$ and $i_{j_2} = 0$

The proof can be found in chapter 4. Let us consider the consequences of the lemma. As can be deduced from the dynamic programming equation, when considering assigning an arbitrary customer to one of the free servers, we have to compare $v_{(x, i+e_j)}^n$ for various j . By (3.4.3) $v_{(x, i+e_j)}^n$ is minimal for the j corresponding to the fastest free server. Equation (3.4.4) is the well known monotonicity. (Because we did not use $b_k \geq 0$ in its proof, it follows from the monotonicity that blocking is always optimal if $b_k < 0$.) Equation (3.4.5) is concerned with the assignment of customers with the highest blocking costs. It says that assigning such a customer to an arbitrary server is better than blocking, i.e. a class 1 customer should never be blocked unless the system is full. Equation (3.4.6) says that when a class k customer is blocked in state (x, i) , i.e. $v_{(x, i+e_{j_1})}^n - b_k - v_{(x, i)}^n \geq 0$, it is also blocked when there are more customers present (and the state of the MAP is the same). On the other hand, when a customer is admitted, it is admitted as well in states with less customers. Another monotonicity property is the following. If $v_{(x, i+e_{j_1})}^n - b_{k_1} - v_{(x, i)}^n \geq 0$, then also $v_{(x, i+e_{j_1})}^n - b_{k_2} - v_{(x, i)}^n \geq 0$, if $k_1 < k_2$. Thus, when blocking is favorable for class k_1 , then blocking is also favorable for class k_2 . Similarly, when customers of a certain class are admitted, then all customer classes with higher blocking costs are admitted as well. This gives the following.

Theorem. *For all T , an optimal policy minimizing the blocking costs from 0 to T exists and has the following properties:*

If a customer is admitted it should be sent to the fastest free server;

Class 1 customers are never blocked, unless the system is full;

If a class k customer is blocked in (x, i_1) , it is blocked in $(x, i_1 + i_2)$;

If a class k customer is admitted in $(x, i_1 + i_2)$, it is admitted in (x, i_1) ;

If a class k customer is blocked in (x, i) , all classes with indices higher than k are blocked as well in (x, i) ;

If a class k customer is admitted in (x, i) , all classes with indices lower than k are admitted as well in (x, i) .

This result comes from [35].

If the arrival stream is an MDAP, then (3.4.3), (3.4.4) and (3.4.5) still hold, but (3.4.6) fails. We demonstrate this with the following example. Take $m = 2$ and $\mu_1 = \mu_2 = 1$. The arrival process is as follows. There is an arrival in class 2 at $t = 0$, after which the arrival process moves to one of 2 states. If action 1 is chosen, the arrival process moves to state 1, where customers of class 1 arrive according to a Poisson process with rate λ_1 . There are no class 2 arrivals. If action 2 is chosen, the arrival process moves to state 2, in which there are no class 1 arrivals, but where there are Poisson arrivals in class 2 with rate λ_2 . This arrival process can easily be approximated by MDAP's. It appears that, for suitable values of $b_1, b_2, \lambda_1, \lambda_2$ and T , it is optimal to block the class 2 customer and choose action 1 if the system is empty, but to admit the class 2 customer and choose action 2 if there is one customer available. This means that (3.4.6) does not hold. Using the uniformization method the different strategies can easily be compared. Equation (3.4.6) fails for example for $b_1 = 10, b_2 = 1, \lambda_1 = 1, \lambda_2 = 3.5$ and $T = 5$. It is straightforward to give an intuitive explanation.

3.5. Customer assignment models with rejections

Here we want to generalize the model of section 1.6 to asymmetric servers. Section 1.6 deals with a symmetric customer assignment model, for which it is shown that the SQP maximizes the number of departures from the system. Thus, we analyze the model of section 3.2, but with a different objective function. We will see however, that the conditions on the arrival probabilities and the service rates need to be different. The model we study has as dynamic programming equation:

$$\begin{aligned}
 v_{(x,i,k)}^{n+1} = & c_{(x,i,k)} + \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay;i} \min_j \{ v_{(y,i+e_j \wedge B,k)}^n \} + \right. \right. \\
 & \left. \left. (1 - q_{xay;i}) v_{(y,i,k)}^n \right) \right\} + \\
 & \sum_{j=1}^m \mu_{ji} \left(\delta_{ij} v_{(x,i-e_j,k+1)}^n + (1 - \delta_{ij}) v_{(x,i,k)}^n \right) + (1 - \gamma - \sum_{j=1}^m \mu_{ji}) v_{(x,i,k)}^n.
 \end{aligned} \tag{3.5.1}$$

The extra component of the state space k counts the number of departures. As in the server assignment models with a single server we study both the case in which rejection is allowed and the case in which it is not allowed.

We start with the model in which rejection is not allowed, meaning that the minimization ranges over all j for which $i_j < B_j$. To make the notation shorter, let i^* again be the permutation of i with i_{j_1} and i_{j_2} interchanged. Assume all vectors considered are componentwise smaller than B . The conditions for q are:

$$q_{xay;i} = q_{xay;i^*} \text{ if } |i| = |i^*|$$

$$q_{xay;i} \geq q_{xay;i+e_{j_1}}$$

Although the conditions are more restrictive than in section 3.2, the finite source model still satisfies them.

For μ , we take the same conditions as in section 3.2. Now we have:

3.5.1. Lemma. *If*

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i+e_{j_2},k)} \quad \text{for } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2, \quad (3.5.2)$$

$$w_{(x,i,k+1)} \leq w_{(x,i+e_{j_1},k)}, \quad (3.5.3)$$

$$w_{(x,i,k+1)} \leq w_{(x,i,k)} \quad (3.5.4)$$

and

$$w_{(x,i,k)} \leq w_{(x,i^*,k)} \quad \text{for } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2 \quad (3.5.5)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The proof can be found in chapter 4. The only meaningful cost function is again $v^0_{(x,i,k)} = -k$ (and $c_{(x,i,k)} = 0$).

3.5.2. Theorem. *In the case of a DMDAP, an SFQP maximizes the expected number of departed customers between 0 and T , if rejection is not allowed.*

If we want to allow rejections, we have to assume that $q_{xay;i}$ is independent of i , i.e. the arrival process is an MDAP, and we cannot model a finite source. This is intuitively clear: if the system is relatively full it might be better to reject a customer in order to make a better choice when the customer comes again.

Concerning the service rates, we need the extra condition $\mu_{ji+e_{j_1}} \geq \mu_{ji}$ for all j . As we already assumed the reverse for $j \neq j_1$, it amounts to:

$$\mu_{ji} = \mu_{j\tilde{i}} \text{ if } i_j = \tilde{i}_j$$

and

$$\mu_{j_1 i+e_{j_1}} \geq \mu_{j_1 i}.$$

This is not surprising. On one hand, if we assign to the shortest queue, customers leave the system fast. To agree with this, service rates must be high in states with few customers. On the other hand, states with few customers can be reached by rejecting customers, and to agree with this, service rates should be high in states with many customers. This reasoning intuitively explains the fact that the service rates must be constant.

For completeness, we give the other conditions as well.

$$\mu_{ji} = 0 \text{ if } i_j = 0$$

$$\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} \geq \mu_{j_1 i+e_{j_2}} + \mu_{j_2 i+e_{j_2}} \text{ if } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{j_1 i} \geq \mu_{j_2 i^*} \text{ if } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2$$

$$\mu_{j_1 i} + \mu_{j_2 i} \geq \mu_{j_1 i^*} + \mu_{j_2 i^*} \text{ if } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2$$

3.5.3. Lemma. *If*

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i+e_{j_2},k)} \quad \text{for } i_{j_1} \leq i_{j_2} \text{ and } j_1 < j_2, \quad (3.5.6)$$

$$w_{(x,i,k+1)} \leq w_{(x,i+e_{j_1},k)}, \quad (3.5.7)$$

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i,k)} \quad (3.5.8)$$

and

$$w_{(x,i,k)} \leq w_{(x,i^*,k)} \quad \text{for } i_{j_1} > i_{j_2} \text{ and } j_1 < j_2 \quad (3.5.9)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The proof can be found in chapter 4. Equation (3.5.8) shows that there exists an optimal policy that does not reject customers. Because of the arbitrary buffers we need (3.5.8) in the proof of (3.5.7). Note that lemma 1.6.1 is a special case of lemma 3.5.3, using the equivalent of lemma 3.3.1.

3.5.4. Theorem. *In the case of an MDAP, an SFQP maximizes the expected number of departed customers between 0 and T , if rejection is allowed.*

Also the model with $B = e$ gives a myopic optimal policy, as the first model studied in the previous section. Because we did not handle this model in chapter 1, we do it here. As is easily seen equation (3.5.6) to (3.5.9) simplify to

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i+e_{j_2},k)} \quad \text{if } j_1 < j_2,$$

$$w_{(x,i,k+1)} \leq w_{(x,i+e_{j_1},k)}$$

and

$$w_{(x,i+e_{j_1},k)} \leq w_{(x,i,k)}.$$

Then we have, in case of an MAP:

3.5.5. Theorem. *For all T , the FQP maximizes the number of departed customers between 0 and T stochastically.*

Remark. We end this section by considering the differences between the models of this section. In the second model, in which rejection is allowed, we need $q_{xay;i+e_{j_1}} \geq q_{xay;i}$ to prove (3.5.8), and $q_{xay;i} \geq q_{xay;i+e_{j_1}}$ to prove (3.5.7). Thus $q_{xay;i}$ must be independent of i . In the first model, in which rejection is not allowed, we do not have (3.5.8), and thus we only assume $q_{xay;i} \geq q_{xay;i+e_{j_1}}$.

3.6. Server assignment model with multiple servers

In this section we generalize the result for the server assignment model with multiple servers of section 2.5 to servers which are partially available, and to arrival processes that stop producing customers if $i = 0$. Let us first describe the model of section 2.5 again.

Customers arrive in m different classes. The service times of customers in class j are exponentially distributed with rate μ_j , $\mu_1 \leq \dots \leq \mu_m$. In the case of arrivals according to an MDAP and multiple servers it is shown in section 2.5 that the SIP is optimal.

Here we have arrivals according to a DMDAP, with the following condition. There are numbers q_{xay}^k such that $q_{xay;0}^k \leq q_{xay}^k$ and $q_{xay;i}^k = q_{xay}^k$ if $i \neq 0$. If we take $q_{xay;0}^k = 0$ the system stays empty once it becomes empty. This way we can study the length of the busy period for the model with an MDAP, even in the case that there are arrivals (in the system with $q_{xay;0}^k = q_{xay}^k$) after the first emptiness.

Perhaps more interesting is the following generalization. In the model of section 1.12 we modeled server vacations, i.e. a server is either working at full speed or not working at all. Here we introduce more possibilities, by assuming that server k is working at speed $p_k(x)$, $0 \leq p_k(x) \leq 1$. Note that this can also be modeled with the arrival process. The dynamic programming equation is:

$$\begin{aligned} v_{(x,i)}^{n+1} = & c_{(x,i)} + \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay;i}^j v_{(y,i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xay;i}^j \right) v_{(y,i)}^n \right) \right\} + \\ & \min_{l_1, \dots, l_s} \left\{ \sum_{k=1}^s p_k(x) \left(\mu_{l_k} v_{(x,i-e_{l_k})}^n + (\mu - \mu_{l_k}) v_{(x,i)}^n \right) \right\} + \\ & (1 - \gamma - \sum_{k=1}^s p_k(x) \mu) v_{(x,i)}^n, \end{aligned}$$

with l_k the queue to which server k is assigned, and the second minimization taken over all allowable actions (with possibly $l_k = 0$, meaning that server k idles). We have again:

3.6.1. Lemma. *If*

$$\begin{aligned} \mu_{j_1} w_{(x,i-e_{j_1})} + (\mu - \mu_{j_1}) w_{(x,i)} & \leq \mu_{j_2} w_{(x,i-e_{j_2})} + (\mu - \mu_{j_2}) w_{(x,i)} \quad (3.6.1) \\ & \text{for } i_{j_1}, i_{j_2} > 0 \text{ and } j_1 < j_2 \end{aligned}$$

and

$$w_{(x,i-e_{j_1})} \leq w_{(x,i)} \text{ for } i_{j_1} > 0 \quad (3.6.2)$$

hold for the cost functions c and v^0 , then they hold for all v^n .

The proof can be found in chapter 4. There it is also shown that the policy that assigns the servers with the highest speed to the customers in low

numbered classes is optimal. We call such a policy a *Fastest Server Smallest Index Policy* (FSSIP). The first to obtain this type of optimality result (without arrivals) were Weiss & Pinedo [80]. As a special case they showed that the FSSIP minimizes the expected makespan.

3.6.2. Theorem. *A FSSIP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.6.1) and (3.6.2).*

See section 1.12 for a discussion of the allowable cost functions.

Remark. In the proof of (3.6.2) we used neither (3.6.1) nor $\mu_1 \leq \dots \leq \mu_m$, meaning that lemma 3.6.1 not only gives the optimality of LEPT, but also the monotonicity in the server assignment models of the sections 1.12, 2.5 and 2.6.

3.7. Server assignment model with a single server and a finite source

In this chapter we introduced the DMDAP. The main motivation to do so was to model a finite source in the customer assignment models. In the server assignment models this cannot be done in general due to the multiple customer classes. In this section we handle a special model with m customer classes, all of finite source type, and with a single server. The service parameters are as usual, λ_j is the rate at which each customer of class j enters queue j , and N_j is the total number of customers of class j . We show that, for certain cost functions, the SIP is optimal if $\lambda_1 \leq \dots \leq \lambda_m$. The case $\lambda_1 \leq \dots \leq \lambda_m$ and $\mu_1 \geq \dots \geq \mu_m$ is studied in Righter [56]. We formulate the dynamic programming equation. The direct costs are not modeled because the optimal policy appears to be myopic, and thus we can use corollary 5.2.2 or 5.2.3.

$$v_i^{n+1} = \sum_{j=1}^m (N_j - i_j) \lambda_j v_{i+e_j}^n + \min_l \left\{ \mu_l v_{i-e_l}^n + (\mu - \mu_l) v_i^n \right\} +$$

$$(1 - \sum_{j=1}^m (N_j - i_j) \lambda_j - \mu) v_i^n.$$

As contrasted with the other single server models, we need monotonicity here to prove the structure of the optimal policy, giving the following lemma.

3.7.1. Lemma. *If*

$$\mu_{j_1} w_{i-e_{j_1}} + (\mu - \mu_{j_1}) w_i \leq \mu_{j_2} w_{i-e_{j_2}} + (\mu - \mu_{j_2}) w_i \quad (3.7.1)$$

for $i_{j_1}, i_{j_2} > 0$ and $j_1 < j_2$

and

$$w_{i-e_{j_1}} \leq w_i \text{ for } i_{j_1} > 0 \quad (3.7.2)$$

hold for the cost function v^0 , then they hold for all v^n .

3.7.2. Theorem. *If $\lambda_1 \leq \dots \leq \lambda_m$, then the SIP minimizes the costs at T (from 0 to T) for all cost functions satisfying (3.7.1) and (3.7.2).*

Equation (3.7.1) and (3.7.2) are the same as equation (1.11.1) and (1.11.2), thus the same cost functions are allowable.

An interesting interpretation for linear costs is given in Chakka & Mitrani [9]. In this model the customer are the servers of a multi-server queue. They are subject to failure (with rates λ_j), and are repaired by a single repairman (with rates μ_j), which is the server in our model. If we assume that a class j server has a service rate c_j , then minimizing $\sum_j i_j c_j$ corresponds to maximizing the total service capacity.

The condition $\lambda_1 \leq \dots \leq \lambda_m$ is essential; to illustrate this, we give an example with linear costs where no list policy is optimal. (A list policy is a policy if all customers are ordered (the list) and served according to this order.) For our example we choose a model with three customers and the following parameters: $\lambda_1 = 2.00$, $\lambda_2 = 1.00$, $\lambda_3 = 0.10$, $\mu_1 = 3.15$, $\mu_2 = 2.00$, $\mu_3 = 1.00$, $c_1 = 1.00$, $c_2 = 1.00$ and $c_3 = 0.05$. We see that $\mu_1 c_1 \geq \mu_2 c_2 \geq \mu_3 c_3$ and $\lambda_1 \geq \lambda_2 \geq \lambda_3$, making this model fall outside the scope of theorem 3.7.2. For each of the 24 different policies we computed the average holding costs. For the six list policies the values are given below. Each list policy is characterized by its list, thus policy $\{a, b, c\}$ indicates the policy which gives highest priority to customer a , and lowest priority to c , and its value is denoted by $v(a, b, c)$. The values are as follows: $v(1, 2, 3) = 0.8803$, $v(1, 3, 2) = 0.9338$, $v(2, 1, 3) = 0.8806$, $v(2, 3, 1) = 0.9285$, $v(3, 1, 2) = 0.9569$, and $v(3, 2, 1) = 0.9559$. Thus $(1, 2, 3)$ is the best list policy. However, let us consider the policy that gives lowest priority to the third customer, that serves customer 1 in state $(1, 1, 0)$, but serves customer 2 in state $(1, 1, 1)$. Computations show that this policy is optimal, with value 0.8800. This shows that there need not be an optimal list policy.

We could leave it at that, but let us try to gain some more insight in the model by giving a heuristic explanation for this phenomenon. Customer three plays a role of little importance. It fails seldomly (as $\lambda_3 = 0.10$), and if it has failed, it has the lowest repair priority (as $c_3 = 0.05$). The parameters are chosen such that if only the customers 1 and 2 are available for repair, then customer 1 gets served first. However, if customer 3 is also at the queue, the time it takes to repair customers 1 and 2 plays a more important role, as this determines the instant at which the repair of customer 3 begins. To start repair early on customer 3, service should start with customer 2 (cf. theorem 3.7.2, as $\lambda_2 < \lambda_1$). The parameters for customer 3 are chosen such that the availability of customers changes the order in which customers 1 and 2 should be served.

In Koole & Vrijenhoek [40] these results are also derived, and additional references are given. Furthermore, we derive policies which are asymptotically optimal. For the case that the server idles most of the time, the μc -rule is optimal; for the heavy traffic case the SIP is optimal if $\mu_1 c_1 / \lambda_1 \geq \dots \geq \mu_m c_m / \lambda_m$.

Chapter 4

Proofs of dynamic programming results

4.1. Proofs of chapter 1

Proof of lemma 1.4.1. By induction on n . The case $n = 0$ is the condition on the cost function. Now assume that (1.4.2) to (1.4.4) hold up to n . First we determine the optimal action at v^{n+1} in (i, j) . Consider the dynamic programming equation (1.4.1). If $i = j$ then both terms in the minimization are equal by (1.4.4), symmetry. Thus, again by symmetry, it is enough to consider $i < j$.

It is easily seen that $(i - 1)^+ \leq (j - 1)^+$, $(i - 1)^+ \leq j$, $i \leq (j - 1)^+$ and $i \leq j$, from which follows, by (1.4.2), that the 4 terms in $v_{(i,j)}^{n+1}$ corresponding to assigning the arriving customer to queue 1 are one by one smaller than the terms corresponding to assigning to queue 2. This gives us that sending an arriving customer to the first queue is better, even if we knew where departures would take place.

Note that combining (1.4.3), monotonicity in the first queue, and symmetry gives $v_{(i,j)}^n \leq v_{(i,j+1)}^n$, monotonicity in the second queue, and that (1.4.2) and symmetry gives $v_{(i,j+1)}^n \leq v_{(i+1,j)}^n$ if $i \geq j$.

Now we prove (1.4.2) for v^{n+1} . The case $i = j$ follows from symmetry. Thus assume $i < j$. Because of this, assignment to the first queue is not only optimal in (i, j) , but also in states like $(i, j - 1)$. We have

$$\lambda\mu^2 v_{(i+1,j-1)}^n \leq \lambda\mu^2 v_{((i-1)^++1,j)}^n$$

by (1.4.2) if $i > 0$ and by monotonicity in the second queue if $i = 0$;

$$\lambda\mu(1 - \mu)v_{(i+1,j)}^n \leq \lambda\mu(1 - \mu)v_{((i-1)^++1,j+1)}^n$$

and

$$\lambda(1 - \mu)\mu v_{(i+2,j-1)}^n \leq \lambda(1 - \mu)\mu v_{(i+1,j)}^n$$

by (1.4.2) if $i < j - 1$; in case $i = j - 1$ we have

$$\begin{aligned} & \lambda\mu(1 - \mu)v_{(i+1,i+1)}^n + \lambda(1 - \mu)\mu v_{(i+2,i)}^n \leq \\ & \lambda\mu(1 - \mu)v_{((i-1)^++1,i+2)}^n + \lambda(1 - \mu)\mu v_{(i+1,i+1)}^n \end{aligned}$$

by symmetry if $i > 0$ and monotonicity if $i = 0$; we have

$$\lambda(1 - \mu)^2 v_{(i+2,j)}^n \leq \lambda(1 - \mu)^2 v_{(i+1,j+1)}^n$$

by (1.4.2);

$$(1 - \lambda)\mu^2 v_{(i,j-1)}^n \leq (1 - \lambda)\mu^2 v_{((i-1)^+,j)}^n$$

and

$$(1 - \lambda)\mu(1 - \mu)v_{(i,j)}^n \leq (1 - \lambda)\mu(1 - \mu)v_{((i-1)^+,j+1)}^n$$

by (1.4.2) if $i > 0$ and monotonicity if $i = 0$;

$$(1 - \lambda)(1 - \mu)\mu v_{(i+1,j-1)}^n \leq (1 - \lambda)(1 - \mu)\mu v_{(i,j)}^n$$

and

$$(1 - \lambda)(1 - \mu)^2 v_{(i+1,j)}^n \leq (1 - \lambda)(1 - \mu)^2 v_{(i,j+1)}^n$$

by (1.4.2). Summing all terms gives $v_{(i+1,j)}^{n+1} \leq v_{(i,j+1)}^{n+1}$.

We continue with (1.4.3). If $i + 1 < j$, then also $i < j$ and assignment to the first queue is optimal in both (i, j) and $(i + 1, j)$; if $i + 1 > j$, then assignment to the second queue is optimal in both (i, j) and $(i + 1, j)$. Choose action 1 in $(i + 1, j)$ if $i + 1 = j$. Then the optimal action in (i, j) is the same as in $(i + 1, j)$. Showing $v_{(i,j)}^{n+1} \leq v_{(i+1,j)}^{n+1}$ can now be done by using (1.4.3) on all corresponding terms, unless $i = 0$, then we have equality in all terms corresponding to departures in queue 1.

The last equation, $v_{(i,j)}^{n+1} = v_{(j,i)}^{n+1}$, follows easily. \square

Proof of lemma 1.5.1. By induction. We will check (1.5.1) to (1.5.3) for all possible realizations of U_{n+1} . From the induction hypothesis we have that all relations given below hold for each realization of U_1, \dots, U_n . We start with (1.5.1). Assume $i_{j_1} < i_{j_2}$. The case $i_{j_1} = i_{j_2}$ is a special case of (1.5.3). If $U_{n+1} \in [\sum_{z < y} \lambda_{xz}, \sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy})$ an arrival occurs. Let j^* be the shortest queue in $i + e_{j_2}$, i.e. the optimal action in $(y, i + e_{j_2})$. Then, if $j^* \neq j_1$,

$$V_{(x,i+e_{j_1})}^{n+1} = \min_j \{V_{(y,i+e_{j_1}+e_j)}^n\} \leq V_{(y,i+e_{j_1}+e_{j^*})}^n \stackrel{(1.5.1)}{\leq}$$

$$V_{(y,i+e_{j_2}+e_{j^*})}^n = \min_j \{V_{(y,i+e_{j_2}+e_j)}^n\} = V_{(x,i+e_{j_2})}^{n+1}.$$

If $j^* = j_1$ then (we omit the terms with V^{n+1})

$$\min_j \{V_{(y,i+e_{j_1}+e_j)}^n\} \leq V_{(y,i+e_{j_1}+e_{j_2})}^n = \min_j \{V_{(y,i+e_{j_2}+e_j)}^n\}.$$

If $U_{n+1} \in [\sum_{z < y} \lambda_{xz} + \lambda_{xy} q_{xy}, \sum_{z \leq y} \lambda_{xz})$, then trivially

$$V_{(y,i+e_{j_1})}^n \leq V_{(y,i+e_{j_2})}^n.$$

We took $(j) < (j+1)$ if $i_{(j)} = i_{(j+1)}$. However, by (1.5.3), the ordering in case of ties can be taken arbitrary. Now we can make sure that queue j_1 is served in $i + e_{j_1}$ and $i + e_{j_2}$ for the same value of U_{n+1} , by taking queue j_1 first in $i + e_{j_1}$ amongst the queues with $i_{j_1} + 1$ customers and by taking queue j_1 last in $i + e_{j_2}$ amongst the queues with i_{j_1} customers. Similarly, we can assure that queue j_2 is served for the same values of U_{n+1} in $i + e_{j_1}$ and $i + e_{j_2}$. Now, if $U_{n+1} \in [\gamma + (j-1)\mu, \gamma + j\mu)$ with $(j) \neq j_1$ or j_2 ,

$$V_{(x, (i+e_{j_1}-e_{(j)})^+)}^n \stackrel{(1.5.1)}{\leq} V_{(x, (i+e_{j_2}-e_{(j)})^+)}^n.$$

If $(j) = j_1$ then, if $i_{j_1} > 0$,

$$V_{(x, i)}^n \stackrel{(1.5.1)}{\leq} V_{(x, i+e_{j_2}-e_{j_1})}^n$$

and, if $i_{j_1} = 0$,

$$V_{(x, i)}^n \stackrel{(1.5.2)}{\leq} V_{(x, i+e_{j_2})}^n.$$

If $(j) = j_2$ then

$$V_{(x, i+e_{j_1}-e_{j_2})}^n \stackrel{(1.5.1)}{\leq} V_{(x, i)}^n.$$

If $U_{n+1} \geq \gamma + m\mu$, then

$$V_{(x, i+e_{j_1})}^n \stackrel{(1.5.1)}{\leq} V_{(x, i+e_{j_2})}^n.$$

We continue with (1.5.2). If $U_{n+1} \in [\sum_{z < y} \lambda_{xz}, \sum_{z < y} \lambda_{xz} + \lambda_{xy}q_{xy})$ and $i + e_{j_1} = B$, we have

$$\min_j \{V_{(y, i+e_j)}^n\} = V_{(y+e_{j_1})}^n,$$

if $i + e_{j_1} \neq B$ then

$$\min_j \{V_{(y, i+e_j)}^n\} \leq V_{(y, i+e_{j_1})}^n \stackrel{(1.5.2)}{\leq} \min_j \{V_{(y, i+e_{j_1}+e_j)}^n\}.$$

The cases $U_{n+1} \in [\sum_{z < y} \lambda_{xz} + \lambda_{xy}q_{xy}, \sum_{z \leq y} \lambda_{xz})$ and $U_{n+1} \in [\gamma + m\mu, 1]$ follow easily. With respect to the departures we can again reorder the ties such that all queues in i and $i + e_{j_1}$ are served for the same U_{n+1} . Now look at the departures at queue j . If $j \neq j_1$,

$$V_{(x, (i-e_j)^+)}^n \stackrel{(1.5.2)}{\leq} V_{(x, (i+e_{j_1}-e_j)^+)}^n.$$

If $j = j_1$ and $i_{j_1} > 0$, then

$$V_{(x, i-e_{j_1})}^n \stackrel{(1.5.2)}{\leq} V_{(x, i)}^n$$

and if $j = j_1$ and $i_{j_1} = 0$ then

$$V_{(x, i)}^n \leq V_{(x, i)}^n.$$

As for (1.5.3), the only non-trivial eventuality is when a customer arrives, because the buffers might give problems. However, it is easily checked that the smallest non-full queue in i and i^* have the same number of customers. \square

Proof of lemma 1.7.1. By induction. Assume the lemma holds up to n . We start with (1.7.2). Assume $i_{j_1} < i_{j_2}$. The case $i_{j_1} = i_{j_2}$ can be done with (1.7.4). Let j^* be the optimal action in $(y, i + e_{j_2})$ at stage $n+1$. Then $j^* \neq j_2$. If $j^* = j_1$, we have

$$\begin{aligned} & q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \leq \\ & \quad q_{xy}v_{(y, i+e_{j_1}+e_{j_2})}^n + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \stackrel{(1.7.2)}{\leq} \\ & \quad q_{xy}v_{(y, i+e_{j_1}+e_{j_2})}^n + (1 - q_{xy})v_{(y, i+e_{j_2})}^n = \\ & \quad q_{xy} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_2})}^n. \end{aligned}$$

If $j^* \neq j_1$ we have

$$\begin{aligned} & q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \leq \\ & \quad q_{xy}v_{(y, i+e_{j_1}+e_{j^*})}^n + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \stackrel{(1.7.2)}{\leq} \\ & \quad q_{xy}v_{(y, i+e_{j_2}+e_{j^*})}^n + (1 - q_{xy})v_{(y, i+e_{j_2})}^n = \\ & \quad q_{xy} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_2})}^n. \end{aligned}$$

Now it follows that

$$\begin{aligned} & \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \right) \leq \\ & \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_2})}^n \right). \end{aligned}$$

Concerning the departures, note that each customer in $(x, i+e_{j_1})$ and $(x, i+e_{j_2})$ is served. We have

$$\mu v_{(x, i+e_{j_1}-e_j)}^n \stackrel{(1.7.2)}{\leq} \mu v_{(x, i+e_{j_2}-e_j)}^n,$$

if $i_j > 0$. Summing this for all customers in state i gives all terms, except those corresponding to the extra customers in queue j_1 and j_2 . However, their term is easy:

$$\mu v_{(x, i+e_{j_1}-e_{j_1})}^n = \mu v_{(x, i+e_{j_2}-e_{j_2})}^n.$$

The dummy term follows easily from (1.7.2). Summing the terms gives

$$v_{(x, i+e_{j_1})}^{n+1} \leq v_{(x, i+e_{j_2})}^{n+1}.$$

We continue with (1.7.3). Let j^* be the optimal action in (y, i) . If $j^* \neq j_1$, then j^* is also optimal in $(y, i + e_{j_1})$, and

$$\begin{aligned} & q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n = \\ & q_{xy} v_{(y, i+e_{j_1}+e_{j^*})}^n + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \stackrel{(1.7.3)}{\leq} \\ & q_{xy} v_{(y, i+e_{j^*})}^n + (1 - q_{xy})v_{(y, i)}^n = \\ & q_{xy} \min_j \{v_{(y, i+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i)}^n. \end{aligned}$$

If $j^* = j_1$ we have

$$\begin{aligned} & q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \leq v_{(y, i+e_{j_1})}^n \stackrel{(1.7.3)}{\leq} \\ & q_{xy} v_{(y, i+e_{j_1})}^n + (1 - q_{xy})v_{(y, i)}^n = \\ & q_{xy} \min_j \{v_{(y, i+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i)}^n. \end{aligned}$$

Note that this derivation also holds in case $i + e_{j_1} = B$.

Now we have

$$\begin{aligned} & \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i+e_{j_1})}^n \right) \leq \\ & \sum_y \lambda_{xy} \left(q_{xy} \min_j \{v_{(y, i+e_j \wedge B)}^n\} + (1 - q_{xy})v_{(y, i)}^n \right). \end{aligned}$$

For all customers except for the extra customer in class j_1 we have

$$\mu v_{(x, i+e_{j_1}-e_j)}^n \stackrel{(1.7.3)}{\leq} \mu v_{(x, i-e_j)}^n.$$

The extra customer is considered together with a dummy term with coefficient μ :

$$\mu v_{(x, i+e_{j_1}-e_{j_1})}^n = \mu v_{(x, i)}^n.$$

The coefficients of the remaining dummy terms are equal and the inequalities follow easily.

Equation (1.7.4) follows easily. □

Proof of lemma 1.7.3. By induction. We start with (1.7.5). The arrival terms go the same as in lemma 1.7.1. Now consider the departures. If $i + e_{j_1}$ and $i + e_{j_2}$ both have an empty group, there is only the dummy term. If $i + e_{j_2}$, but not $i + e_{j_1}$ has an empty group, (1.7.6) can be used, and what remains are dummy terms with equal coefficients. If the system is in both $i + e_{j_1}$ and $i + e_{j_2}$ up, we have

$$v_{(x, i+e_{j_1}-e_j, k+1)}^n \leq v_{(x, i+e_{j_2}-e_j, k+1)}^n$$

for each j , by (1.7.5).

We continue with (1.7.6). Let j^* be the optimal assignment in (y, i, k) at step $n + 1$. Then, if $i \neq B$,

$$\begin{aligned} \sum_{j_1=1}^m \left(q_{xy} \min_j \{ v_{(y, i-e_{j_1}+e_j \wedge B, k)}^n \} + (1 - q_{xy}) v_{(y, i-e_{j_1}, k)}^n \right) &\leq \\ \sum_{j_1=1}^m \left(q_{xy} v_{(y, i-e_{j_1}+e_{j^*}, k)}^n + (1 - q_{xy}) v_{(y, i-e_{j_1}, k)}^n \right) &\stackrel{(1.7.6)}{\leq} \\ m \left(q_{xy} v_{(y, i+e_{j^*}, k)}^n + (1 - q_{xy}) v_{(y, i, k)}^n \right) &= \\ m \left(q_{xy} \min_j \{ v_{(y, i+e_j \wedge B, k)}^n \} + (1 - q_{xy}) v_{(y, i, k)}^n \right). \end{aligned}$$

If $i = B$ then in each state $(y, i - e_{j_1}, k)$ we can send an arrival to a full group:

$$\begin{aligned} \sum_{j_1=1}^m \left(q_{xy} \min_j \{ v_{(y, i-e_{j_1}+e_j \wedge B, k)}^n \} + (1 - q_{xy}) v_{(y, i-e_{j_1}, k)}^n \right) &\leq \\ \sum_{j_1=1}^m \left(q_{xy} v_{(y, i-e_{j_1}, k)}^n + (1 - q_{xy}) v_{(y, i-e_{j_1}, k)}^n \right) &\stackrel{(1.7.6)}{\leq} \\ m \left(q_{xy} v_{(y, i, k)}^n + (1 - q_{xy}) v_{(y, i, k)}^n \right) &= \\ m \left(q_{xy} \min_j \{ v_{(y, i+e_j \wedge B, k)}^n \} + (1 - q_{xy}) v_{(y, i, k)}^n \right). \end{aligned}$$

This gives the inequalities for the arrival terms.

Concerning the departures, if $i_{j_1} > 1$ we have

$$\mu \sum_{j_2=1}^m v_{(x, i-e_{j_1}-e_{j_2}, k+2)}^n \stackrel{(1.7.6)}{\leq} \mu m v_{(x, i-e_{j_1}, k+1)}^n;$$

if $i_{j_1} = 1$ we have

$$\mu m v_{(x, i-e_{j_1}, k+1)}^n = \mu m v_{(x, i-e_{j_1}, k+1)}^n.$$

Summation of these terms for $j_1 = 1, \dots, m$ gives the terms concerning departures, leaving dummy terms with the same coefficients.

We continue with (1.7.7). The arrival term can be shown similar to the arrival term of (1.7.3). When the system is up or down in both $i + e_{j_1}$ and i the departure terms follow easily by induction. If $i + e_{j_1} \geq e$ and $i_{j_1} = 1$, then first (1.7.6) should be used.

Equation (1.7.8) follows easily by induction. Also (1.7.9) can be proven easily. \square

Proof of lemma 1.8.1. By induction. First we will show that

$$\int v_{(i+te_{j_1}-se)^+}^n dP(t) \leq \int v_{(i+te_{j_2}-se)^+}^n dP(t) \text{ if } i_{j_1} \leq i_{j_2}$$

holds for all s , i.e. that it is optimal to assign to the queue with the smallest workload. First assume that $i_{j_1} - s \geq 0$. This means that $(i + te_j - se)^+ = (i - se)^+ + te_j$ for $j = j_1$ and $j = j_2$. Then we have

$$\begin{aligned} \int v_{(i+te_{j_1}-se)^+}^n dP(t) &= \int v_{(i-se)^++te_{j_1}}^n dP(t) \stackrel{(1.8.2)}{\leq} \\ &= \int v_{(i-se)^++te_{j_2}}^n dP(t) = \int v_{(i+te_{j_2}-se)^+}^n dP(t). \end{aligned}$$

Now assume that $i_{j_1} - s < 0$, but $i_{j_2} - s \geq 0$. By (1.8.3), monotonicity, we have $v_{(i+te_{j_1}-se)^+}^n \leq v_{(i-se)^++te_{j_1}}^n$. This gives

$$\begin{aligned} \int v_{(i+te_{j_1}-se)^+}^n dP(t) &\leq \int v_{(i-se)^++te_{j_1}}^n dP(t) \stackrel{(1.8.2)}{\leq} \\ &= \int v_{(i-se)^++te_{j_2}}^n dP(t) = \int v_{(i+te_{j_2}-se)^+}^n dP(t). \end{aligned}$$

Finally assume that $i_{j_2} - s < 0$. We can rewrite $(i + te_{j_2} - se)^+$ as $(i - se)^+ + t^* e_{j_2}$ with $t^* = (t - s + i_{j_2})^+$. Note that $t^* < t$. Because $(i + te_{j_1} - se)^+ \leq (i - se)^+ + t^* e_{j_1}$ we have, by (1.8.3), $v_{(i+te_{j_1}-se)^+}^n \leq v_{(i-se)^++t^*e_{j_1}}^n$. Thus

$$\begin{aligned} \int v_{(i+te_{j_1}-se)^+}^n dP(t) &\leq \int v_{(i-se)^++t^*e_{j_1}}^n dP(t) = \\ &= \int v_{(i-se)^++t^*e_{j_2}}^n dP(t) = \int v_{(i+te_{j_2}-se)^+}^n dP(t). \end{aligned}$$

Having shown that assigning to the smallest queue is optimal, the inequalities will follow quite easily.

Consider (1.8.2). Let j^* be the optimal assignment in $i + te_{j_2}$. If $j^* = j_1$, then

$$\int \min_j \left\{ \int v_{(i+te_{j_1}+se_j-u_n e)^+}^n dP(s) \right\} dP(t) \leq$$

$$\int \int v_{(i+te_{j_1}+se_{j_2}-u_n e)+}^n dP(s)dP(t) =$$

$$\int \min_j \left\{ \int v_{(i+te_{j_2}+se_j-u_n e)+}^n dP(s) \right\} dP(t).$$

If $j^* \neq j_1$, then

$$\int \min_j \left\{ \int v_{(i+te_{j_1}+se_j-u_n e)+}^n dP(s) \right\} dP(t) \leq$$

$$\int \int v_{(i+te_{j_1}+se_{j^*}-u_n e)+}^n dP(s)dP(t) \leq$$

$$\int \int v_{(i+te_{j_2}+se_{j^*}-u_n e)+}^n dP(s)dP(t) =$$

$$\int \min_j \left\{ \int v_{(i+te_{j_2}+se_j-u_n e)+}^n dP(s) \right\} dP(t),$$

the second inequality by the optimality of the SWP as shown above.

Concerning (1.8.3), if j^* is the optimal action in $i + te_{j_1}$, we have

$$\min_j \left\{ \int v_{(i+se_j-u_n e)+}^n dP(s) \right\} \leq \int v_{(i+se_{j^*}-u_n e)+}^n dP(s) \stackrel{(1.8.3)}{\leq}$$

$$\int v_{(i+te_{j_1}+se_{j^*}-u_n e)+}^n dP(s) = \min_j \left\{ \int v_{(i+te_{j_1}+se_j-u_n e)+}^n dP(s) \right\}.$$

Equation (1.8.4), symmetry, is as usual trivial to prove. \square

Proof of lemma 1.11.5. By induction. Assume the lemma holds up to n . We start with the arrivals. Because

$$\mu_{j_1} v_{(y,i-e_{j_1}+e_j)}^n + (\mu - \mu_{j_1}) v_{(y,i-e_{j_1}+e_{f(j_1)}+e_j)}^n \stackrel{(1.11.6)}{\leq}$$

$$\mu_{j_2} v_{(y,i-e_{j_2}+e_j)}^n + (\mu - \mu_{j_2}) v_{(y,i-e_{j_2}+e_{f(j_2)}+e_j)}^n$$

the arrival term follows easily.

Consider the terms concerning departures. Let j^* be the optimal action in $(x, i - e_{j_2})$. Because $(i - e_{j_2})_{j_1} > 0$, $j^* \leq j_1$. Because $f(j_2) \geq j_2 - 1$ we see that j^* is also optimal in $(x, i - e_{j_2} + e_{f(j_2)})$. We distinguish two cases, $j^* < j_1$ and $j^* = j_1$. Assume $j^* < j_1$. Then j^* is also optimal in $(x, i - e_{j_1})$ and $(x, i - e_{j_1} + e_{f(j_1)})$. We have

$$\mu_{j_1} \min_l \left\{ \mu_l v_{(x,i-e_{j_1}-e_l)}^n + (\mu - \mu_l) v_{(x,i-e_{j_1}-e_l+e_{f(l)})}^n \right\} +$$

$$(\mu - \mu_{j_1}) \min_l \left\{ \mu_l v_{(x,i-e_{j_1}+e_{f(j_1)}-e_l)}^n + (\mu - \mu_l) v_{(x,i-e_{j_1}+e_{f(j_1)}-e_l+e_{f(l)})}^n \right\} =$$

$$\begin{aligned}
 & \mu_{j_1}(\mu_{j^*} v_{(x, i-e_{j_1}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_1}-e_{j^*}+e_{f(j^*)})}^n) + \\
 & (\mu - \mu_{j_1})(\mu_{j^*} v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j^*}+e_{f(j^*)})}^n) \stackrel{(1.11.6)}{\leq} \\
 & \mu_{j_2}(\mu_{j^*} v_{(x, i-e_{j_2}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_2}-e_{j^*}+e_{f(j^*)})}^n) + \\
 & (\mu - \mu_{j_2})(\mu_{j^*} v_{(x, i-e_{j_2}+e_{f(j_2)}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_2}+e_{f(j_2)}-e_{j^*}+e_{f(j^*)})}^n) = \\
 & \mu_{j_2} \min_l \{ \mu_l v_{(x, i-e_{j_2}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_2}-e_l+e_{f(l)})}^n \} + \\
 & (\mu - \mu_{j_2}) \min_l \{ \mu_l v_{(x, i-e_{j_2}+e_{f(j_2)}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_2}+e_{f(j_2)}-e_l+e_{f(l)})}^n \}.
 \end{aligned}$$

Now consider $j^* = j_1$. Then j_2 is an allowable action in $(x, i - e_{j_1})$ and $(x, i - e_{j_1} + e_{f(j_1)})$. Then we have

$$\begin{aligned}
 & \mu_{j_1} \min_l \{ \mu_l v_{(x, i-e_{j_1}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}-e_l+e_{f(l)})}^n \} + \\
 & (\mu - \mu_{j_1}) \min_l \{ \mu_l v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l+e_{f(l)})}^n \} \leq \\
 & \mu_{j_1}(\mu_{j_2} v_{(x, i-e_{j_1}-e_{j_2})}^n + (\mu - \mu_{j_2}) v_{(x, i-e_{j_1}-e_{j_2}+e_{f(j_2)})}^n) + \\
 & (\mu - \mu_{j_1})(\mu_{j_2} v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j_2})}^n + (\mu - \mu_{j_2}) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j_2}+e_{f(j_2)})}^n) = \\
 & \mu_{j_2} \min_l \{ \mu_l v_{(x, i-e_{j_2}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_2}-e_l+e_{f(l)})}^n \} + \\
 & (\mu - \mu_{j_2}) \min_l \{ \mu_l v_{(x, i-e_{j_2}+e_{f(j_2)}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_2}+e_{f(j_2)}-e_l+e_{f(l)})}^n \}.
 \end{aligned}$$

The dummy transition follows easily by induction. \square

Proof of lemma 1.11.6. With induction. Assume the lemma holds up to n . The arrival term follows easily, like in the proof of lemma 1.11.5. Let j^* be the optimal action in state (x, i) . If $j^* \neq j_1$, then

$$\begin{aligned}
 & \mu_{j_1} \min_l \{ \mu_l v_{(x, i-e_{j_1}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}-e_l+e_{f(l)})}^n \} + \\
 & (\mu - \mu_{j_1}) \min_l \{ \mu_l v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l+e_{f(l)})}^n \} \leq \\
 & \mu_{j_1}(\mu_{j^*} v_{(x, i-e_{j_1}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_1}-e_{j^*}+e_{f(j^*)})}^n) + \\
 & (\mu - \mu_{j_1})(\mu_{j^*} v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j^*})}^n + (\mu - \mu_{j^*}) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_{j^*}+e_{f(j^*)})}^n) \stackrel{(1.11.7)}{\leq} \\
 & \mu \mu_{j^*} v_{(x, i-e_{j^*})}^n + \mu(\mu - \mu_{j^*}) v_{(x, i-e_{j^*}+e_{f(j^*)})}^n = \\
 & \mu \min_l \{ \mu_l v_{(x, i-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_l+e_{f(l)})}^n \}.
 \end{aligned}$$

If $j^* = j_1$, then, because idling is allowed now,

$$\begin{aligned}
 & \mu_{j_1} \min_l \{ \mu_l v_{(x, i-e_{j_1}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}-e_l+e_{f(l)})}^n \} + \\
 & (\mu - \mu_{j_1}) \min_l \{ \mu_l v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_{j_1}+e_{f(j_1)}-e_l+e_{f(l)})}^n \} \leq \\
 & \mu_{j_1} \mu v_{(x, i-e_{j_1})}^n + (\mu - \mu_{j_1}) \mu v_{(x, i-e_{j_1}+e_{f(j_1)})}^n = \\
 & \mu \min_l \{ \mu_l v_{(x, i-e_l)}^n + (\mu - \mu_l) v_{(x, i-e_l+e_{f(l)})}^n \}.
 \end{aligned}$$

The dummy transition follows easily by induction. \square

4.2. Proofs of chapter 2

Proof of lemma 2.6.1. By induction. Assume the lemma holds up to n . We start with (2.6.2). The terms regarding arrivals at the first center follow easily.

Consider the departures from the first center. Let j^* be the optimal action in the first center in state (x, \tilde{i}, i) . If $j^* \neq j_1$, j^* is allowable in state $(x, \tilde{i} - e_{j_1}, i + e_{j_1})$ and the term follows by induction. If $j^* = j_1$, and $\tilde{i} = e_{j_1}$, then idling is the only action in state in state $(x, \tilde{i} - e_{j_1}, i + e_{j_1})$ and the term follows by induction. If there is at least one more customer available, say in queue j_2 , and $j^* = j_1$, then

$$\begin{aligned} & \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_{j_1} - e_j, i + e_{j_1} + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n \right\} \leq \\ & \tilde{\mu}_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, i + e_{j_1} + e_{j_2})}^n + (\tilde{\mu} - \tilde{\mu}_{j_2}) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n \leq \\ & \tilde{\mu}_{j_2} v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n + (\tilde{\mu} - \tilde{\mu}_{j_2}) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n \leq \\ & \tilde{\mu}_{j_1} v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n + (\tilde{\mu} - \tilde{\mu}_{j_1}) v_{(x, \tilde{i}, i)}^n = \\ & \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_j, i + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned} \tag{2.6.2}$$

Consider the departures from the second center. The optimal action in (x, \tilde{i}, i) is allowable in $(x, \tilde{i} - e_{j_1}, i + e_{j_1})$. Therefore the term follows easily by induction.

Equation (2.6.3) follows from a result in section 3.6. \square

Proof of lemma 2.6.3. By induction. Assume the lemma holds up to n . We start with (2.6.4). Assume $n + 1 \leq i_j$. The terms concerning arrivals follow immediately, using induction, because $n < i_j$. Consider the terms corresponding to departures from center 1. Let j^* be the optimal action in state \tilde{i} . Because $\tilde{i}_{j_1} > 0$, j^* is also optimal in $\tilde{i} - e_{j_2}$. If $j^* \neq j_1$, then the terms follow easily by induction. If $j^* = j_1$, then

$$\begin{aligned} & \tilde{\mu}_{j_1} \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_{j_1} - e_j, i + e_{j_1} + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_{j_1}) \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_j, i + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\} \leq \\ & \tilde{\mu}_{j_1} \tilde{\mu}_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, i + e_{j_1} + e_{j_2})}^n + \tilde{\mu}_{j_1} (\tilde{\mu} - \tilde{\mu}_{j_2}) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n + \\ & (\tilde{\mu} - \tilde{\mu}_{j_1}) \tilde{\mu}_{j_2} v_{(x, \tilde{i} - e_{j_2}, i + e_{j_2})}^n + (\tilde{\mu} - \tilde{\mu}_{j_1}) (\tilde{\mu} - \tilde{\mu}_{j_2}) v_{(x, \tilde{i}, i)}^n = \\ & \tilde{\mu}_{j_2} \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_{j_2} - e_j, i + e_{j_2} + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i} - e_{j_2}, i + e_{j_2})}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_{j_2}) \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_j, i + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned}$$

Consider the second center. By (2.6.5), serving queue \tilde{j} is always optimal. The terms follow by induction. Note that we used (2.6.5) at step n with at least $n + 1$ customers in queue \tilde{j} . Also the dummy term follows easily.

Consider (2.6.5). Again the terms concerning arrivals and the dummy transition follow easily. The optimal action in the first center of (x, \tilde{i}, i) depends only on \tilde{i} . Because the number of customers in queue \tilde{j} in state $(x, \tilde{i}, i - e_{j_1})$, $(x, \tilde{i}, i - e_{\tilde{j}})$ and (x, \tilde{i}, i) is $i_{\tilde{j}} - 1$ or more, there are at least n customers available, meaning that, by (2.6.4), the same action is optimal in each state. Therefore also the terms concerning departures from the first center follow easily. Concerning the second center, serving queue \tilde{j} is optimal in each state. Also these terms follow easily by induction. \square

Proof of lemma 2.6.6. By induction. Assume the lemma holds up to n . We start with (2.6.6). Assume $n + 1 \leq i_1 + i_2$. The terms concerning arrivals follow immediately, using induction, because $n < i_1 + i_2$. Consider the terms corresponding to departures from center 1. In \tilde{i} and $\tilde{i} - e_2$ it is optimal to serve queue 1. Thus

$$\begin{aligned} & \tilde{\mu}_1 \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_1 - e_j, i + e_1 + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i} - e_1, i + e_1)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_1) \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_j, i + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\} \leq \\ & \tilde{\mu}_1 \tilde{\mu}_2 v_{(x, \tilde{i} - e_1 - e_2, i + e_1 + e_2)}^n + \tilde{\mu}_1 (\tilde{\mu} - \tilde{\mu}_2) v_{(x, \tilde{i} - e_1, i + e_1)}^n + \\ & (\tilde{\mu} - \tilde{\mu}_1) \tilde{\mu}_2 v_{(x, \tilde{i} - e_2, i + e_2)}^n + (\tilde{\mu} - \tilde{\mu}_1) (\tilde{\mu} - \tilde{\mu}_2) v_{(x, \tilde{i}, i)}^n = \\ & \tilde{\mu}_2 \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_2 - e_j, i + e_2 + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i} - e_2, i + e_2)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_2) \min_j \left\{ \tilde{\mu}_j v_{(x, \tilde{i} - e_j, i + e_j)}^n + (\tilde{\mu} - \tilde{\mu}_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned}$$

Consider the second center. If $i_1 > 0$, serving queue 1 is optimal in $i + e_1$, i and $i + e_2$, using that (2.6.7) holds for $i_1 + i_2 \geq n + 1$ at stage n . Then

$$\begin{aligned} & \tilde{\mu}_1 \min_j \left\{ \mu_j v_{(x, \tilde{i} - e_1, i + e_1 - e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i} - e_1, i + e_1)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_1) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i - e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\} = \\ & \tilde{\mu}_1 \mu_1 v_{(x, \tilde{i} - e_1, i + e_1 - e_1)}^n + \tilde{\mu}_1 (\mu - \mu_1) v_{(x, \tilde{i} - e_1, i + e_1)}^n + \\ & (\tilde{\mu} - \tilde{\mu}_1) \mu_1 v_{(x, \tilde{i}, i - e_1)}^n + (\tilde{\mu} - \tilde{\mu}_1) (\mu - \mu_1) v_{(x, \tilde{i}, i)}^n \stackrel{(2.6.6)}{\leq} \\ & \tilde{\mu}_2 \mu_1 v_{(x, \tilde{i} - e_2, i + e_2 - e_1)}^n + \tilde{\mu}_2 (\mu - \mu_1) v_{(x, \tilde{i} - e_2, i + e_2)}^n + \\ & (\tilde{\mu} - \tilde{\mu}_2) \mu_1 v_{(x, \tilde{i}, i - e_1)}^n + (\tilde{\mu} - \tilde{\mu}_2) (\mu - \mu_1) v_{(x, \tilde{i}, i)}^n = \end{aligned}$$

$$\begin{aligned} & \tilde{\mu}_2 \min_j \left\{ \mu_j v_{(x, \tilde{i}-e_2, i+e_2-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}-e_2, i+e_2)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_2) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned}$$

We wanted to prove the inequality for all i with $n+1 \leq i_1 + i_2$. We used at stage n (2.6.6) with $i_1 + i_2 + 1 > n$ customers in the second center.

If $i_1 = 0$, then $i_2 > 0$. Thus serving queue 2 is optimal in i and $i + e_2$. Then

$$\begin{aligned} & \tilde{\mu}_1 \min_j \left\{ \mu_j v_{(x, \tilde{i}-e_1, i+e_1-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}-e_1, i+e_1)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_1) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\} \leq \\ & \tilde{\mu}_1 \mu_2 v_{(x, \tilde{i}-e_1, i+e_1-e_2)}^n + \tilde{\mu}_1 (\mu - \mu_2) v_{(x, \tilde{i}-e_1, i+e_1)}^n + \\ & (\tilde{\mu} - \tilde{\mu}_1) \mu_2 v_{(x, \tilde{i}, i-e_2)}^n + (\tilde{\mu} - \tilde{\mu}_1) (\mu - \mu_2) v_{(x, \tilde{i}, i)}^n \stackrel{(2.6.6)}{\leq} \\ & \tilde{\mu}_2 \mu_2 v_{(x, \tilde{i}-e_2, i+e_2-e_2)}^n + \tilde{\mu}_2 (\mu - \mu_2) v_{(x, \tilde{i}-e_2, i+e_2)}^n + \\ & (\tilde{\mu} - \tilde{\mu}_2) \mu_2 v_{(x, \tilde{i}, i-e_2)}^n + (\tilde{\mu} - \tilde{\mu}_2) (\mu - \mu_2) v_{(x, \tilde{i}, i)}^n = \\ & \tilde{\mu}_2 \min_j \left\{ \mu_j v_{(x, \tilde{i}-e_2, i+e_2-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}-e_2, i+e_2)}^n \right\} + \\ & (\tilde{\mu} - \tilde{\mu}_2) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned}$$

Also the dummy term follows easily.

Consider (2.6.7). Again the terms concerning arrivals and the dummy transition follow easily. The optimal action in the first center of (x, \tilde{i}, i) depends only on \tilde{i} . Because the number of customers in center 2 in state $(x, \tilde{i}, i - e_1)$, $(x, \tilde{i}, i - e_2)$ and (x, \tilde{i}, i) is $i_1 + i_2 - 1$ or more, there are at least n customers available, meaning that, by (2.6.6), the same action is optimal in each state. Therefore also the terms concerning departures from the first center follow easily. Concerning the second center, we have

$$\begin{aligned} & \mu_1 \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_1-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i-e_1)}^n \right\} + \\ & (\mu - \mu_1) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\} \leq \\ & \mu_1 \mu_2 v_{(x, \tilde{i}, i-e_1-e_2)}^n + \mu_1 (\mu - \mu_2) v_{(x, \tilde{i}, i-e_1)}^n + \\ & (\mu - \mu_1) \mu_2 v_{(x, \tilde{i}, i-e_2)}^n + (\mu - \mu_1) (\mu - \mu_2) v_{(x, \tilde{i}, i)}^n = \\ & \mu_2 \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_2-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i-e_2)}^n \right\} + \\ & (\mu - \mu_2) \min_j \left\{ \mu_j v_{(x, \tilde{i}, i-e_j)}^n + (\mu - \mu_j) v_{(x, \tilde{i}, i)}^n \right\}. \end{aligned}$$

□

Proof of lemma 2.7.1. By induction. Assume the lemma holds up to n . In all 3 equations the term corresponding to arrivals and the dummy term go easily with induction, like in the proof of lemma 1.11.5. Therefore we only consider the terms regarding departures at the first and the last center. We start with equation (2.7.1). Serving queue j_2 in $(x, \tilde{i} - e_{j_1}, i + e_{j_1})$ is optimal, thus the terms corresponding to departures from the first center of the l.h.s. are:

$$\begin{aligned} & \mu_{j_1} \mu_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, i + e_{j_1} + e_{j_2})}^n + \mu_{j_1} (\mu - \mu_{j_2}) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n + \\ & (\mu - \mu_{j_1}) \mu_{j_2} v_{(x, \tilde{i} - e_{j_2}, i + e_{j_2})}^n + (\mu - \mu_{j_1}) (\mu - \mu_{j_2}) v_{(x, \tilde{i}, i)}^n. \end{aligned}$$

We have to show that this expression is equal to the one where all j_1 and j_2 are exchanged. Number the 4 terms consecutively. The first and 4th term are both symmetric in j_1 and j_2 . Term 2 with j_1 and j_2 exchanged is term 3.

We continue with the second center. First assume $i \neq 0$, thus there is a j_3 such that $i_{j_3} > 0$. Then we have:

$$\begin{aligned} & \mu_{j_1} \mu_{j_3} v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1} - e_{j_3})}^n + \mu_{j_1} (\mu - \mu_{j_3}) v_{(x, \tilde{i} - e_{j_1}, i + e_{j_1})}^n + \\ & (\mu - \mu_{j_1}) \mu_{j_3} v_{(x, \tilde{i}, i - e_{j_3})}^n + (\mu - \mu_{j_1}) (\mu - \mu_{j_3}) v_{(x, \tilde{i}, i)}^n. \end{aligned}$$

We use (2.7.1) twice, once with $\tilde{i}, i - e_{j_3}$ for terms 1 and 3 and once for terms 2 and 4.

When $i = 0$, we need (2.7.3) to prove (2.7.1). The terms corresponding to departures in the second center of the l.h.s. are:

$$\mu_{j_1} \mu_{j_1} v_{(x, \tilde{i} - e_{j_1}, 0)}^n + \mu_{j_1} (\mu - \mu_{j_1}) v_{(x, \tilde{i} - e_{j_1}, e_{j_1})}^n + (\mu - \mu_{j_1}) \mu v_{(x, \tilde{i}, 0)}^n.$$

Equation (2.7.3) immediately gives the expression wanted.

Now we prove (2.7.2). The terms corresponding to departures from the first center go directly with induction. Thus the following terms remain:

$$\begin{aligned} & \mu_{j_1} \mu_{j_2} v_{(x, \tilde{i}, i - e_{j_1} - e_{j_2})}^n + \mu_{j_1} (\mu - \mu_{j_2}) v_{(x, \tilde{i}, i - e_{j_1})}^n + \\ & (\mu - \mu_{j_1}) \mu_{j_2} v_{(x, \tilde{i}, i - e_{j_2})}^n + (\mu - \mu_{j_1}) (\mu - \mu_{j_2}) v_{(x, \tilde{i}, i)}^n. \end{aligned}$$

This expression is symmetric in j_1 and j_2 .

We continue with (2.7.3). The terms concerning departures at both centers are:

$$\begin{aligned} & \mu_{j_1}^2 \mu_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, e_{j_2})}^n + \mu_{j_1}^2 (\mu - \mu_{j_2}) v_{(x, \tilde{i} - e_{j_1}, 0)}^n + \mu_{j_1}^2 \mu v_{(x, \tilde{i} - e_{j_1}, 0)}^n + \\ & \mu_{j_1} (\mu - \mu_{j_1}) \mu_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, e_{j_1} + e_{j_2})}^n + \mu_{j_1} (\mu - \mu_{j_1}) (\mu - \mu_{j_2}) v_{(x, \tilde{i} - e_{j_1}, e_{j_1})}^n + \\ & \mu_{j_1} (\mu - \mu_{j_1}) \mu_{j_1} v_{(x, \tilde{i} - e_{j_1}, 0)}^n + \mu_{j_1} (\mu - \mu_{j_1}) (\mu - \mu_{j_1}) v_{(x, \tilde{i} - e_{j_1}, e_{j_1})}^n + \\ & (\mu - \mu_{j_1}) \mu \mu_{j_1} v_{(x, \tilde{i} - e_{j_1}, e_{j_1})}^n + (\mu - \mu_{j_1}) \mu (\mu - \mu_{j_1}) v_{(x, \tilde{i}, 0)}^n + (\mu - \mu_{j_1}) \mu^2 v_{(x, \tilde{i}, 0)}^n = \\ & \mu_{j_1}^2 \mu_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, e_{j_2})}^n + \mu_{j_1} (\mu - \mu_{j_1}) \mu_{j_2} v_{(x, \tilde{i} - e_{j_1} - e_{j_2}, e_{j_1} + e_{j_2})}^n + \\ & \mu_{j_1}^2 (3\mu - \mu_{j_1} - \mu_{j_2}) v_{(x, \tilde{i} - e_{j_1}, 0)}^n + \mu_{j_1} (\mu - \mu_{j_1}) (3\mu - \mu_{j_1} - \mu_{j_2}) v_{(x, \tilde{i} - e_{j_1}, e_{j_1})}^n + \\ & (\mu - \mu_{j_1}) \mu (3\mu - \mu_{j_1} - \mu_{j_2}) v_{(x, \tilde{i}, 0)}^n - (\mu - \mu_{j_1}) \mu (\mu - \mu_{j_2}) v_{(x, \tilde{i}, 0)}^n. \end{aligned}$$

Number the terms consecutively. For term 1 and 2 we use (2.7.2) for $\tilde{i} - e_{j_1} - e_{j_2}, e_{j_1} + e_{j_2}$, for term 3, 4 and 5 we use (2.7.3). Term 6 is symmetric. \square

4.3. Proofs of chapter 3

Proof of lemma 3.2.1. By induction. We start with (3.2.4). Assume $i_{j_1} < i_{j_2}$. The case $i_{j_1} = i_{j_2}$ is a special case of (3.2.6). We start with the term corresponding to arrivals. Let j^* be the optimal assignment in $(y, i + e_{j_2})$. Then we have

$$\begin{aligned}
 & q_{xay; i+e_{j_1}} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \leq \\
 & \qquad \qquad \qquad (3.2.4) \\
 & q_{xay; i+e_{j_1}} v_{(y, i+e_{j_1}+e_{j^*})}^n + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \leq \\
 & \qquad \qquad \qquad (3.2.1)+(3.2.5) \\
 & q_{xay; i+e_{j_1}} v_{(y, i+e_{j_2}+e_{j^*})}^n + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_2})}^n \leq \\
 & q_{xay; i+e_{j_2}} v_{(y, i+e_{j_2}+e_{j^*})}^n + (1 - q_{xay; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n = \\
 & q_{xay; i+e_{j_2}} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n
 \end{aligned}$$

if $j^* \neq j_1$ and

$$\begin{aligned}
 & q_{xay; i+e_{j_1}} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \leq \\
 & \qquad \qquad \qquad (3.2.4) \\
 & q_{xay; i+e_{j_1}} v_{(y, i+e_{j_1}+e_{j_2})}^n + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \leq \\
 & \qquad \qquad \qquad (3.2.1)+(3.2.5) \\
 & q_{xay; i+e_{j_1}} v_{(y, i+e_{j_1}+e_{j_2})}^n + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_2})}^n \leq \\
 & q_{xay; i+e_{j_2}} v_{(y, i+e_{j_2}+e_{j_1})}^n + (1 - q_{xay; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n = \\
 & q_{xay; i+e_{j_2}} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n
 \end{aligned}$$

if $j^* = j_1$. Note that j^* cannot be equal to j_2 . Now let a^* be the optimal action in $(x, i + e_{j_2})$. We have

$$\begin{aligned}
 & \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay; i+e_{j_1}} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \right) \right\} \leq \\
 & \sum_y \lambda_{xa^*y} \left(q_{xa^*y; i+e_{j_1}} \min_j \{v_{(y, i+e_{j_1}+e_j \wedge B)}^n\} + (1 - q_{xa^*y; i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \right) \leq \\
 & \sum_y \lambda_{xa^*y} \left(q_{xa^*y; i+e_{j_2}} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xa^*y; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n \right) = \\
 & \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay; i+e_{j_2}} \min_j \{v_{(y, i+e_{j_2}+e_j \wedge B)}^n\} + (1 - q_{xay; i+e_{j_2}}) v_{(y, i+e_{j_2})}^n \right) \right\}.
 \end{aligned}$$

Consider a departure at queue j , $j \neq j_1, j_2$:

$$\begin{aligned} \mu_{ji+e_{j_1}} v_{(x,i+e_{j_1}-e_j)}^n &\stackrel{(3.2.4)}{\leq} \mu_{ji+e_{j_1}} v_{(x,i+e_{j_2}-e_j)}^n \stackrel{(3.2.5)}{\leq} \\ &\mu_{ji+e_{j_2}} v_{(x,i+e_{j_2}-e_j)}^n + (\mu_{ji+e_{j_1}} - \mu_{ji+e_{j_2}}) v_{(x,i+e_{j_2})}^n. \end{aligned}$$

The terms corresponding to a departure from queue j_1 and j_2 will be considered together. We have $v_{(x,i+e_{j_1}-e_{j_2})}^n \leq v_{(x,i+e_{j_1}-e_{j_1})}^n = v_{(x,i+e_{j_2}-e_{j_2})}^n \leq v_{(x,i+e_{j_2}-e_{j_1})}^n$, by (3.2.4). As $\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} \geq \mu_{j_1 i+e_{j_2}} + \mu_{j_2 i+e_{j_2}}$, we have, together with (3.2.5),

$$\begin{aligned} &\mu_{j_1 i+e_{j_1}} v_{(x,i+e_{j_1}-e_{j_1})}^n + \mu_{j_2 i+e_{j_1}} v_{(x,i+e_{j_1}-e_{j_2})}^n \leq \\ &\mu_{j_1 i+e_{j_2}} v_{(x,i+e_{j_2}-e_{j_1})}^n + \mu_{j_2 i+e_{j_2}} v_{(x,i+e_{j_2}-e_{j_2})}^n + \\ &(\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} - \mu_{j_1 i+e_{j_2}} - \mu_{j_2 i+e_{j_2}}) v_{(x,i+e_{j_2})}^n. \end{aligned}$$

Note that we did not use (3.2.6) in the above proof. However, we used it for the case $i_{j_1} = i_{j_2}$.

Now we prove (3.2.5), monotonicity. The arrival term is easy. Let a^* be the optimal action in $(x, i + e_{j_1})$.

$$\begin{aligned} \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay;i} \min_j \{ v_{(y,i+e_j \wedge B)}^n \} + (1 - q_{xay;i}) v_{(y,i)}^n \right) \right\} &\leq \\ \sum_y \lambda_{xa^*y} \left(q_{xa^*y;i} v_{(y,i+e_{j_1})}^n + (1 - q_{xa^*y;i}) v_{(y,i)}^n \right) &\stackrel{(3.2.5)}{\leq} \\ \sum_y \lambda_{xa^*y} v_{(y,i+e_{j_1})}^n &\stackrel{(3.2.5)}{\leq} \\ \sum_y \lambda_{xa^*y} \left(q_{xa^*y;i+e_{j_1}} \min_j \{ v_{(y,i+e_{j_1}+e_j \wedge B)}^n \} + (1 - q_{xa^*y;i+e_{j_1}}) v_{(y,i+e_{j_1})}^n \right) &= \\ \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay;i+e_{j_1}} \min_j \{ v_{(y,i+e_{j_1}+e_j \wedge B)}^n \} + (1 - q_{xay;i+e_{j_1}}) v_{(y,i+e_{j_1})}^n \right) \right\}. \end{aligned}$$

Consider a departure at queue j , $j \neq j_1$. Then

$$\mu_{ji} v_{(x,i-e_j)}^n \stackrel{(3.2.5)}{\leq} \mu_{ji} v_{(x,i+e_{j_1}-e_j)}^n \stackrel{(3.2.5)}{\leq}$$

$$\mu_{ji+e_{j_1}} v_{(x,i+e_{j_1}-e_j)}^n + (\mu_{ji} - \mu_{ji+e_{j_1}}) v_{(x,i+e_{j_1})}^n,$$

because $\mu_{ji} \geq \mu_{ji+e_{j_1}}$. By $v_{(x,i-e_{j_1})}^n \leq v_{(x,i)}^n \leq v_{(x,i+e_{j_1})}^n$ we have for the term corresponding to a departure from queue j_1

$$\mu_{j_1 i} v_{(x,i-e_{j_1})}^n \leq \mu_{j_1 i+e_{j_1}} v_{(x,i)}^n + (\mu_{j_1 i} - \mu_{j_1 i+e_{j_1}}) v_{(x,i+e_{j_1})}^n$$

if $\mu_{j_1 i} - \mu_{j_1 i + e_{j_1}} \geq 0$ and

$$\mu_{j_1 i} v_{(x, i - e_{j_1})}^n + (\mu_{j_1 i + e_{j_1}} - \mu_{j_1 i}) v_{(x, i)}^n \leq \mu_{j_1 i + e_{j_1}} v_{(x, i)}^n$$

if $\mu_{j_1 i + e_{j_1}} - \mu_{j_1 i} \geq 0$.

We continue with (3.2.6). Let j^* be the optimal assignment in (y, i^*) . We know that $j^* \neq j_2$. We have

$$\begin{aligned} & q_{xay; i} \min_j \{v_{(y, i + e_j \wedge B)}^n\} + (1 - q_{xay; i}) v_{(y, i)}^n \leq \\ & \quad \quad \quad (3.2.6) \\ & q_{xay; i} v_{(y, i + e_{j^*})}^n + (1 - q_{xay; i}) v_{(y, i)}^n \leq \\ & \quad \quad \quad (3.2.2) + (3.2.5) \\ & q_{xay; i} v_{(y, i^* + e_{j^*})}^n + (1 - q_{xay; i}) v_{(y, i^*)}^n \leq \\ & q_{xay; i^*} v_{(y, i^* + e_{j^*})}^n + (1 - q_{xay; i^*}) v_{(y, i^*)}^n = \\ & q_{xay; i^*} \min_j \{v_{(y, i^* + e_j \wedge B)}^n\} + (1 - q_{xay; i^*}) v_{(y, i^*)}^n \end{aligned}$$

if $j^* \neq j_1$ and

$$\begin{aligned} & q_{xay; i} \min_j \{v_{(y, i + e_j \wedge B)}^n\} + (1 - q_{xay; i}) v_{(y, i)}^n \leq \\ & \quad \quad \quad (3.2.6) \\ & q_{xay; i} v_{(y, i + e_{j_2})}^n + (1 - q_{xay; i}) v_{(y, i)}^n \leq \\ & \quad \quad \quad (3.2.2) + (3.2.5) \\ & q_{xay; i} v_{(y, i^* + e_{j_1})}^n + (1 - q_{xay; i}) v_{(y, i^*)}^n \leq \\ & q_{xay; i^*} v_{(y, i^* + e_{j_1})}^n + (1 - q_{xay; i^*}) v_{(y, i^*)}^n = \\ & q_{xay; i^*} \min_j \{v_{(y, i^* + e_j \wedge B)}^n\} + (1 - q_{xay; i^*}) v_{(y, i^*)}^n \end{aligned}$$

if $j^* = j_1$. The term wanted is derived in the same way as for (3.2.4).

The departures at queue j , $j \neq j_1, j_2$, go similarly as those in (3.2.4). The terms corresponding to a departure from queue j_1 and j_2 will be considered together. Note that $v_{(x, i^* - e_{j_2})}^n \leq v_{(x, i^* - e_{j_1})}^n$, by (3.2.4). Thus, by $\mu_{j_1 i} \geq \mu_{j_2 i^*}$ and $\mu_{j_1 i} + \mu_{j_2 i} \geq \mu_{j_1 i^*} + \mu_{j_2 i^*}$, we have

$$\begin{aligned} & \mu_{j_1 i} v_{(x, i - e_{j_1})}^n + \mu_{j_2 i} v_{(x, i - e_{j_2})}^n \leq \quad (3.2.6) \\ & \mu_{j_1 i} v_{(x, i^* - e_{j_2})}^n + \mu_{j_2 i} v_{(x, i^* - e_{j_1})}^n \leq \\ & \quad \quad \quad (3.2.5) \\ & \mu_{j_2 i^*} v_{(x, i^* - e_{j_2})}^n + (\mu_{j_1 i} - \mu_{j_2 i^*} + \mu_{j_2 i}) v_{(x, i^* - e_{j_1})}^n \leq \\ & \mu_{j_2 i^*} v_{(x, i^* - e_{j_2})}^n + \mu_{j_1 i^*} v_{(x, i^* - e_{j_1})}^n + (\mu_{j_1 i} - \mu_{j_2 i^*} + \mu_{j_2 i} - \mu_{j_1 i^*}) v_{(x, i^*)}^n. \quad \square \end{aligned}$$

Proof of lemma 3.3.4. By induction. We start with (3.3.5). Let j^* be the shortest queue in state $i + ke_{j_2}$. Because $i_{j_1} \leq i_{j_2}$, j^* does not depend on k . It is easily seen that $q_{xay; i+ke_{j_1}} \leq q_{xay; i+ke_{j_2}}$. If $j^* \neq j_1$, we have

$$\begin{aligned}
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \min_j \left\{ \sum_l \beta_l v_{(y, i+ke_{j_1}+le_j)}^n \right\} + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) = \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \sum_l \beta_l v_{(y, i+ke_{j_1}+le_{j^*})}^n + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) \stackrel{(3.3.5)}{\leq} \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \sum_l \beta_l v_{(y, i+ke_{j_2}+le_{j^*})}^n + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) \stackrel{(3.3.1)}{\leq} \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_2}} \sum_l \beta_l v_{(y, i+ke_{j_2}+le_{j^*})}^n + (1 - q_{xay; i+ke_{j_2}}) v_{(y, i+ke_{j_2})}^n \right) = \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_2}} \min_j \left\{ \sum_l \beta_l v_{(y, i+ke_{j_2}+le_j)}^n \right\} + (1 - q_{xay; i+ke_{j_2}}) v_{(y, i+ke_{j_2})}^n \right).
 \end{aligned}$$

If $j^* = j_1$, then

$$\begin{aligned}
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \min_j \left\{ \sum_l \beta_l v_{(y, i+ke_{j_1}+le_j)}^n \right\} + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) \leq \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \sum_l \beta_l v_{(y, i+ke_{j_1}+le_{j_2})}^n + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) \stackrel{(3.3.5)}{\leq} \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_1}} \sum_l \beta_l v_{(y, i+ke_{j_1}+le_{j_2})}^n + (1 - q_{xay; i+ke_{j_1}}) v_{(y, i+ke_{j_1})}^n \right) \stackrel{(3.3.1)}{\leq} \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_2}} \sum_l \beta_l v_{(y, i+ke_{j_1}+le_{j_2})}^n + (1 - q_{xay; i+ke_{j_2}}) v_{(y, i+ke_{j_2})}^n \right) = \\
 & \sum_k \beta_k \left(q_{xay; i+ke_{j_2}} \min_j \left\{ \sum_l \beta_l v_{(y, i+ke_{j_2}+le_j)}^n \right\} + (1 - q_{xay; i+ke_{j_2}}) v_{(y, i+ke_{j_2})}^n \right).
 \end{aligned}$$

The departure term follows as in the proof of lemma 3.2.1.

Consider the departures. Note that, by (3.3.7), we can assume $i_{j_1} < i_{j_2}$. If $i_{j_1} > 0$, the term follows easily by induction. If $i_{j_1} = 0$, the term on all servers except j_1 also follow by induction. For server j_1 we have

$$\sum_k \beta_k w_{(x, (i+ke_{j_1}-e_{j_1})+)} \stackrel{(3.3.6)}{\leq} \sum_k \beta_k w_{(x, i+ke_{j_1})} \stackrel{(3.3.5)}{\leq} \sum_k \beta_k w_{(x, i+ke_{j_2})}.$$

The terms corresponding to the dummy transition and the immediate costs follow easily.

Also (3.3.6) and (3.3.7) follow with induction. \square

Proof of lemma 3.3.7. By induction. We start with (3.3.14). Assume $i_{j_1} < i_{j_2}$. The case $i_{j_1} = i_{j_2}$ is a special case of (3.3.16). We start with the term corresponding to the routable arrivals. Let j^* be the index of the shortest queue in $(y, i + e_{j_2})$. First note that assigning to the shortest queue is still optimal: if a customer arrives, it is favorable by (3.3.14), and the probability that a customer arrives is by (3.3.9) smallest, which is favorable by (3.3.15). Then, if $j^* \neq j_1$,

$$\begin{aligned}
 & \min_j \{q_{xay; i+e_{j_1} j}^0 v_{(y, i+e_{j_1}+e_j)}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} j}^0) v_{(y, i+e_{j_1})}^n\} \leq \\
 & \quad q_{xay; i+e_{j_1} j^*}^0 v_{(y, i+e_{j_1}+e_{j^*})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} j^*}^0) v_{(y, i+e_{j_1})}^n \leq \quad (3.3.14) \\
 & \quad q_{xay; i+e_{j_1} j^*}^0 v_{(y, i+e_{j_2}+e_{j^*})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} j^*}^0) v_{(y, i+e_{j_2})}^n \leq \quad (3.3.10)+(3.3.15) \\
 & \quad q_{xay; i+e_{j_2} j^*}^0 v_{(y, i+e_{j_2}+e_{j^*})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_2} j^*}^0) v_{(y, i+e_{j_2})}^n = \\
 & \quad \min_j \{q_{xay; i+e_{j_2} j}^0 v_{(y, i+e_{j_2}+e_j)}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_2} j}^0) v_{(y, i+e_{j_2})}^n\}.
 \end{aligned}$$

If $j^* = j_1$, assume that \tilde{j} is the shortest queue in $i + e_{j_1}$. We will use that $q_{xay; i+e_{j_1} \tilde{j}}^0 \leq q_{xay; i+e_{j_1} j_1}^0 \leq q_{xay; i+e_{j_2} j_1}^0$, the last inequality because j^* is also the smallest queue in i .

$$\begin{aligned}
 & \min_j \{q_{xay; i+e_{j_1} j}^0 v_{(y, i+e_{j_1}+e_j)}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} j}^0) v_{(y, i+e_{j_1})}^n\} = \\
 & \quad q_{xay; i+e_{j_1} \tilde{j}}^0 v_{(y, i+e_{j_1}+e_{\tilde{j}})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} \tilde{j}}^0) v_{(y, i+e_{j_1})}^n \leq \quad (3.3.14) \\
 & \quad q_{xay; i+e_{j_1} \tilde{j}}^0 v_{(y, i+e_{j_1}+e_{j_2})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_1} \tilde{j}}^0) v_{(y, i+e_{j_2})}^n \leq \quad (3.3.15) \\
 & \quad q_{xay; i+e_{j_2} \tilde{j}}^0 v_{(y, i+e_{j_2}+e_{\tilde{j}})}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_2} \tilde{j}}^0) v_{(y, i+e_{j_2})}^n = \\
 & \quad \min_j \{q_{xay; i+e_{j_2} j}^0 v_{(y, i+e_{j_2}+e_j)}^n + (\tilde{q}_{xay} - q_{xay; i+e_{j_2} j}^0) v_{(y, i+e_{j_2})}^n\}.
 \end{aligned}$$

Note that j^* cannot be equal to j_2 .

Concerning the non-routable arrivals, we have the following. We will show that if there are numbers $\tilde{q}_{l_1}^j, \tilde{q}_{l_2}^j$ such that $\sum_{j=k}^m \tilde{q}_{l_1}^j \leq \sum_{j=k}^m \tilde{q}_{l_2}^j$ for $1 \leq k \leq m$ and $l_1 < l_2$ then

$$\sum_{j=k}^m \tilde{q}_{l_1}^j v_{(x, i+e_{(l_1)}+e_{(j)})}^n + \sum_{j=k}^m (\tilde{q}_{l_2}^j - \tilde{q}_{l_1}^j) v_{(x, i+e_{(l_1)})}^n \leq \sum_{j=k}^m \tilde{q}_{l_2}^j v_{(x, i+e_{(l_2)}+e_{(j)})}^n. \quad (4.3.1)$$

Suppose the relation holds for fixed k . Consider $k-1$. If $\tilde{q}_{l_1}^{k-1} \leq \tilde{q}_{l_2}^{k-1}$, we have by (3.3.14) and (3.3.15),

$$\tilde{q}_{l_1}^{k-1} v_{(x, i+e_{(l_1)}+e_{(k-1)})}^n + (\tilde{q}_{l_2}^{k-1} - \tilde{q}_{l_1}^{k-1}) v_{(x, i+e_{(l_1)})}^n \leq \tilde{q}_{l_2}^{k-1} v_{(x, i+e_{(l_2)}+e_{(k-1)})}^n$$

and the result follows easily. If $\tilde{q}_{l_1}^{k-1} > \tilde{q}_{l_2}^{k-1}$, we have by (3.3.14)

$$\tilde{q}_{l_1}^{k-1} v_{(x, i+e_{(l_1)}+e_{(k-1)})}^n \leq (\tilde{q}_{l_1}^{k-1} - \tilde{q}_{l_2}^{k-1}) v_{(x, i+e_{(l_1)}+e_{(k)})}^n + \tilde{q}_{l_2}^{k-1} v_{(x, i+e_{(l_2)}+e_{(k-1)})}^n.$$

Thus it remains to show that (4.3.1) with \tilde{q} replaced by \hat{q} , with $\hat{q}_{l_1}^j = \tilde{q}_{l_1}^j$ for $j > k$, $\hat{q}_{l_2}^j = \tilde{q}_{l_2}^j$ for $j \geq k$, and $\hat{q}_{l_1}^k = \tilde{q}_{l_1}^k + \tilde{q}_{l_1}^{k-1} - \tilde{q}_{l_2}^{k-1}$ holds. It is easily seen that $\sum_{j=k_1}^m \hat{q}_{l_1}^j \leq \sum_{j=k_1}^m \hat{q}_{l_2}^j$ for $k_1 = k, \dots, m$, completing the induction step. By taking $\tilde{q}_l^j = q_{xay, i+e_{(l)}}^{(j)}$ and l_1 and l_2 such that $(l_1) = j_1$ and $(l_2) = j_2$ we are finished with the term concerning the non-routable arrivals, in case (3.3.11) holds for all k . If $\sum_{j=k}^m \tilde{q}_{l_1}^j \leq \sum_{j=k}^m \tilde{q}_{l_2}^j$ holds for $k = l_1$ and $k > l_2$ only, we can show (4.3.1) for $k = l_1$, using the fact that $v_{(x, i+e_{(l_1)}+e_{(j_1)})}^n \leq v_{(x, i+e_{(l_2)}+e_{(j_2)})}^n$ for all j_1 and j_2 with $l_1 \leq j_1 \leq l_2$ and $l_1 \leq j_2 \leq l_2$, in much the same way as the induction step above. Now, by adding a dummy term we get the arrival term in a similar way as in the proof of lemma 3.2.1.

Consider the assignable server. Omit in the notation the dependence of μ on x . Let j^* be the index of the longest queue in $(y, i + e_{j_2})$. Note that it cannot be j_1 . We can take j^* such that it is also the longest queue in i and $i + e_{j_1}$. By (3.3.13), (3.3.14) and (3.3.15), we see that assigning the server to the longest queue is optimal. We have

$$\min_j \{ \tilde{\mu}_{ji+e_{j_1}} v_{(y, i+e_{j_1}-e_j)}^n + (\tilde{\mu} - \tilde{\mu}_{ji+e_{j_1}}) v_{(y, i+e_{j_1})}^n \} \leq$$

$$\tilde{\mu}_{j^*i+e_{j_1}} v_{(y, i+e_{j_1}-e_{j^*})}^n + (\tilde{\mu} - \tilde{\mu}_{j^*i+e_{j_1}}) v_{(y, i+e_{j_1})}^n \leq \quad (3.3.14)$$

$$\tilde{\mu}_{j^*i+e_{j_1}} v_{(y, i+e_{j_2}-e_{j^*})}^n + (\tilde{\mu} - \tilde{\mu}_{j^*i+e_{j_1}}) v_{(y, i+e_{j_2})}^n \leq \quad (3.3.15)$$

$$\tilde{\mu}_{j^*i+e_{j_2}} v_{(y, i+e_{j_2}-e_{j^*})}^n + (\tilde{\mu} - \tilde{\mu}_{j^*i+e_{j_2}}) v_{(y, i+e_{j_2})}^n =$$

$$\min_j \{ \tilde{\mu}_{ji+e_{j_2}} v_{(y, i+e_{j_2}-e_j)}^n + (\tilde{\mu} - \tilde{\mu}_{ji+e_{j_2}}) v_{(y, i+e_{j_2})}^n \}.$$

Finally, consider the fixed server. Omit again the x in the notation of μ . (Note the similarity between what follows and the way non-routable arrivals were treated.) We will show that if there are numbers $\mu_{j_1}^*$, $\mu_{j_2}^*$ such that $\sum_{j=k}^m \mu_{j_1}^* \geq \sum_{j=k}^m \mu_{j_2}^*$ for $1 \leq k \leq m$ and $l_1 < l_2$ then

$$\sum_{j=k}^m \mu_{j_1}^* v_{(x, i+e_{(l_1)}-e_{(j)})}^n \leq \sum_{j=k}^m \mu_{j_2}^* v_{(x, i+e_{(l_2)}-e_{(j)})}^n + \sum_{j=k}^m (\mu_{j_1}^* - \mu_{j_2}^*) v_{(x, i+e_{(l_2)})}^n. \quad (4.3.2)$$

Suppose the relation holds for fixed k , consider $k-1$. If $\mu_{k-1l_1}^* \geq \mu_{k-1l_2}^*$, we have by (3.3.14) and (3.3.15),

$$\mu_{k-1l_1}^* v_{(x, i+e_{(l_1)}-e_{(k-1)})}^n \leq \mu_{k-1l_2}^* v_{(x, i+e_{(l_2)}-e_{(k-1)})}^n + (\mu_{k-1l_1}^* - \mu_{k-1l_2}^*) v_{(x, i+e_{(l_2)})}^n.$$

Equation (4.3.2) follows easily. If $\mu_{k-1l_1}^* < \mu_{k-1l_2}^*$, we have by (3.3.14)

$$\begin{aligned} & \mu_{k-1l_1}^* v_{(x,i+e_{(l_1)}-e_{(k-1)})}^n + (\mu_{k-1l_2}^* - \mu_{k-1l_1}^*) v_{(x,i+e_{(l_1)}-e_{(k)})}^n \leq \\ & \mu_{k-1l_2}^* v_{(x,i+e_{(l_2)}-e_{(k-1)})}^n. \end{aligned}$$

Now (4.3.2) follows by induction, with $\mu_{kl_2}^*$ replaced by $\mu_{kl_2}^* + \mu_{k-1l_2}^* - \mu_{k-1l_1}^*$. Using a similar reasoning as in the case of non-routable arrivals we find that $\sum_{j=k}^m \mu_{jl_1}^* \geq \sum_{j=k}^m \mu_{jl_2}^*$ only needs to hold for $k = 1, \dots, l_1, l_2 + 1, \dots, m$.

Now we prove (3.3.15), monotonicity. The term concerning the routable arrivals is easy:

$$\begin{aligned} & \min_j \left\{ q_{xay;ij}^0 v_{(y,i+e_j)}^n + (\tilde{q}_{xay} - q_{xay;ij}^0) v_{(y,i)}^n \right\} = \\ & q_{xay;i(1)}^0 v_{(y,i+e_{(1)})}^n + (\tilde{q}_{xay} - q_{xay;i(1)}^0) v_{(y,i)}^n \stackrel{(3.3.15)}{\leq} \\ & \stackrel{(3.3.14)}{\leq} \tilde{q}_{xay} v_{(y,i+e_{(1)})}^n \stackrel{(3.3.15)}{\leq} \tilde{q}_{xay} v_{(y,i+e_{j_1})}^n \leq \\ & \min_j \left\{ q_{xay;i+e_{j_1}j}^0 v_{(y,i+e_{j_1}+e_j)}^n + (\tilde{q}_{xay} - q_{xay;i+e_{j_1}j}^0) v_{(y,i+e_{j_1})}^n \right\}. \end{aligned}$$

Using (3.3.12) we can show

$$\begin{aligned} & \sum_{j=l_1+1}^m q_{xay;i}^{(j)} v_{(x,i+e_{(j)})}^n + \sum_{j=l_1+1}^m (q_{xay;i+e_{(l_1)}}^{(j)} - q_{xay;i}^{(j)}) v_{(x,i)}^n \leq \\ & \sum_{j=l_1+1}^m q_{xay;i+e_{(l_1)}}^{(j)} v_{(x,i+e_{(l_1)}+e_{(j)})}^n, \end{aligned}$$

similar to the analysis of (3.3.14). Because

$$v_{(x,i)}^n \leq v_{(x,i+e_{(j)})}^n \leq v_{(x,i+e_{(l_1)})}^n \leq v_{(x,i+e_{(l_1)}+e_{(j)})}^n$$

for $j \leq l_1$, we have the term wanted.

The terms corresponding to the assignable server are similar to the terms corresponding to the routable arrivals, the terms corresponding to the fixed servers are similar to the term corresponding to the non-routable arrivals.

Equation (3.3.16) is trivial to prove. \square

Proof of lemma 3.4.2. By induction. It is easily seen that $v^0 = 0$ satisfies the inequalities. Assume the lemma holds up to n . We prove the inequalities for the terms on the \bar{m} classes and the terms on departures separately. Multiplying with q_{xy}^k , summing etc. give the complete inequalities. The terms on arrivals are proven by considering the optimal action on the r.h.s., and then finding an action on the l.h.s. for which the inequality holds. We start with (3.4.3).

Consider an arbitrary customer class l . Assume the optimal action in $(x, i + e_{j_2})$ is blocking. Then we have (take blocking in $(x, i + e_{j_1})$):

$$b_l + v_{(x, i + e_{j_1})}^n \leq b_l + v_{(x, i + e_{j_2})}^n,$$

by induction. If the optimal action in $(x, i + e_{j_2})$ is sending to server j_1 , we take server j_2 in $(x, i + e_{j_1})$. Then we have:

$$v_{(x, i + e_{j_1} + e_{j_2})}^n \leq v_{(x, i + e_{j_2} + e_{j_1})}^n.$$

If the optimal action is server $j^* \neq j_1$ in state $(x, i + e_{j_2})$, we take the same action in $(x, i + e_{j_1})$. We have by induction

$$v_{(x, i + e_{j_1} + e_{j^*})}^n \leq v_{(x, i + e_{j_2} + e_{j^*})}^n.$$

Now we have

$$\begin{aligned} \min_{a_l} \left\{ \mathbf{I}\{a_l = 0\}(b_l + v_{(x, i + e_{j_1})}^n) + \sum_j \mathbf{I}\{a_l = j\} v_{(x, i + e_{j_1} + e_j)}^n \right\} \leq \\ \min_{a_l} \left\{ \mathbf{I}\{a_l = 0\}(b_l + v_{(x, i + e_{j_2})}^n) + \sum_j \mathbf{I}\{a_l = j\} v_{(x, i + e_{j_2} + e_j)}^n \right\} \end{aligned}$$

for all l . The term, concerning the arrival process, but without arrivals, goes by induction. Consider the terms corresponding to departures. Terms for $j \neq j_1, j_2$ are done with induction. With the help of (3.4.4) we have, using $\mu_{j_1} \geq \mu_{j_2}$,

$$\mu_{j_1} v_{(x, i)}^n + \mu_{j_2} v_{(x, i + e_{j_1})}^n \leq \mu_{j_2} v_{(x, i)}^n + \mu_{j_1} v_{(x, i + e_{j_1})}^n \leq \mu_{j_2} v_{(x, i)}^n + \mu_{j_1} v_{(x, i + e_{j_2})}^n.$$

Combining these results gives $v_{(x, i + e_{j_1})}^{n+1} \leq v_{(x, i + e_{j_2})}^{n+1}$.

We continue with (3.4.4). Take in (x, i) the optimal action of $(x, i + e_{j_1})$. Then (3.4.4) follows immediately.

Consider (3.4.5). Let j^* be the optimal action in (x, i) for some customer class l . If $j^* \neq j_1$, take action j^* on the l.h.s. and we have

$$v_{(x, i + e_{j^*} + e_{j_1})}^n \leq b_1 + v_{(x, i + e_{j^*})}^n$$

by induction. If the optimal action is j_1 , reject in $i + e_{j_1}$. Then

$$b_l + v_{(x, i + e_{j_1})}^n \leq b_1 + v_{(x, i + e_{j_1})}^n.$$

If the optimal action is blocking, take blocking as action on the l.h.s. For departures at servers $j \neq j_1$ we have

$$\mu_j v_{(x, i + e_{j_1} - e_j \vee 0)}^n \leq \mu_j b_1 + \mu_j v_{(x, i - e_j \vee 0)}^n$$

by induction. For server j_1 we have

$$\mu_{j_1} v_{(x,i)}^n \leq \mu_{j_1} b_1 + \mu_{j_1} v_{(x,i)}^n.$$

This completes the proof of (3.4.5).

Rewrite (3.4.6):

$$v_{(x,i+e_{j_1})}^n + v_{(x,i+e_{j_2})}^n \leq v_{(x,i+e_{j_2}+e_{j_1})}^n + v_{(x,i)}^n.$$

In the following table one can see the optimal actions of the r.h.s. in the left columns and the actions establishing the inequalities in the right columns. Let $j^* = \min\{j | (i + e_{j_1} + e_{j_2})_j = 0\}$. Note that $j^* \neq j_1, j_2$, and that j^* cannot be optimal in i , due to the choice of j_1 . The terms are identified by their states.

$i + e_{j_2} + e_{j_1}$	i	$i + e_{j_1}$	$i + e_{j_2}$	
0	0	0	0	induction
0	j_1	0	j_1	equality
0	j_2	j_2	0	equality
j^*	0	0	j^*	twice induction
j^*	j_1	j^*	j_1	induction
j^*	j_2	j_2	j^*	induction

For example, if, for a certain customer class l , rejection is optimal in i , and if sending a customer to queue j^* is optimal in $i + e_{j_1} + e_{j_2}$, the inequality is established by taking rejection in $i + e_{j_1}$ and action j^* in $i + e_{j_2}$, according to the fourth case in the table. Indeed,

$$v_{(x,i+e_{j_1})}^n - v_{(x,i)}^n \leq v_{(x,i+e_{j_1}+e_{j_2})}^n - v_{(x,i+e_{j_2})}^n \leq v_{(x,i+e_{j_1}+e_{j_2}+e_{j^*})}^n - v_{(x,i+e_{j_2}+e_{j^*})}^n$$

by using induction at both steps, giving

$$b_l + v_{(x,i+e_{j_1})}^n + v_{(x,i+e_{j_2}+e_{j^*})}^n \leq v_{(x,i+e_{j_1}+e_{j_2}+e_{j^*})}^n + b_l + v_{(x,i)}^n.$$

If $i + e_{j_1} + e_{j_2} = e$, only the first three cases have to be considered.

Regarding the departures we have, concerning server j_1 and j_2 ,

$$\mu_{j_1} v_{(x,i)}^n + \mu_{j_2} v_{(x,i+e_{j_1})}^n + \mu_{j_1} v_{(x,i+e_{j_2})}^n + \mu_{j_2} v_{(x,i)}^n$$

at both sides. The other terms follow by induction. □

Proof of lemma 3.5.1. The proof goes by induction. We start with (3.5.2). Let j^* be the optimal action in $i + e_{j_2}$. The analysis goes as usual by differentiating between $j^* \neq j_1$ and $j^* = j_1$. If $j^* \neq j_1$, then

$$v_{(y,i+e_{j_1}+e_{j^*},k)}^n \stackrel{(3.5.2)}{\leq} v_{(y,i+e_{j_2}+e_{j^*},k)}^n.$$

If $j^* = j_1$, then

$$v_{(y, i+e_{j_1}+e_{j_2}, k)}^n \leq v_{(y, i+e_{j_2}+e_{j_1}, k)}^n.$$

Because $q_{xay; i+e_{j_1}} = q_{xay; i+e_{j_2}}$, the term on arrival follows.

The terms concerning departures go the same as in the proof of lemma 3.2.1. Consider a departure at queue j , $j \neq j_1, j_2$ with $i_j > 0$:

$$\begin{aligned} \mu_{ji+e_{j_1}} v_{(x, i+e_{j_1}-e_j, k+1)}^n &\stackrel{(3.5.2)}{\leq} \mu_{ji+e_{j_1}} v_{(x, i+e_{j_2}-e_j, k+1)}^n \stackrel{(3.5.3)}{\leq} \\ &\mu_{ji+e_{j_2}} v_{(x, i+e_{j_2}-e_j, k+1)}^n + (\mu_{ji+e_{j_1}} - \mu_{ji+e_{j_2}}) v_{(x, i+e_{j_2}, k)}^n. \end{aligned}$$

The terms corresponding to a departure from queue j_1 and j_2 will be considered together. We have, by (3.5.2), that both $v_{(x, i+e_{j_1}-e_{j_1}, k+1)}^n$ and $v_{(x, i+e_{j_1}-e_{j_2}, k+1)}^n$ are smaller than both $v_{(x, i+e_{j_2}-e_{j_1}, k+1)}^n$ and $v_{(x, i+e_{j_2}-e_{j_2}, k+1)}^n$. As $\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} \geq \mu_{j_1 i+e_{j_2}} + \mu_{j_2 i+e_{j_2}}$, we have, together with (3.5.3),

$$\begin{aligned} \mu_{j_1 i+e_{j_1}} v_{(x, i+e_{j_1}-e_{j_1}, k+1)}^n + \mu_{j_2 i+e_{j_1}} v_{(x, i+e_{j_1}-e_{j_2}, k+1)}^n &\leq \\ \mu_{j_1 i+e_{j_2}} v_{(x, i+e_{j_2}-e_{j_1}, k+1)}^n + \mu_{j_2 i+e_{j_2}} v_{(x, i+e_{j_2}-e_{j_2}, k+1)}^n + \\ (\mu_{j_1 i+e_{j_1}} + \mu_{j_2 i+e_{j_1}} - \mu_{j_1 i+e_{j_2}} - \mu_{j_2 i+e_{j_2}}) v_{(x, i+e_{j_2}, k)}^n. \end{aligned}$$

The terms concerning costs and the dummy transition follow easily. We continue with (3.5.3). Let j^* be the optimal action in $i + e_{j_1}$. Then j^* is also allowed in i . Now we have, by (3.5.3) and (3.5.4),

$$\begin{aligned} q_{xay; i} \min_j \{v_{(y, i+e_j, k+1)}^n\} &\leq \\ q_{xay; i+e_{j_1}} v_{(y, i+e_{j^*}, k+1)}^n + (q_{xay; i} - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1}, k+1)}^n &\leq \\ q_{xay; i+e_{j_1}} v_{(y, i+e_{j_1}+e_{j^*}, k)}^n + (q_{xay; i} - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1}, k)}^n &= \\ q_{xay; i+e_{j_1}} \min_j \{v_{(y, i+e_{j_1}+e_j, k)}^n\} + (q_{xay; i} - q_{xay; i+e_{j_1}}) v_{(y, i+e_{j_1}, k)}^n. \end{aligned}$$

The arrival term follows as usual. Note that when $i + e_{j_1} = B$ and an arrival occurs, this customer is rejected. This is equivalent to taking $q_{xay; i} = 0$ if $|i| \geq |B|$.

By $\mu_{ji} \geq \mu_{ji+e_{j_1}}$, we have for $j \neq j_1$ and $i_j > 0$

$$\mu_{ji} v_{(x, i-e_j, k+2)}^n \leq \mu_{ji+e_{j_1}} v_{(x, i+e_{j_1}-e_j, k+1)}^n + (\mu_{ji} - \mu_{ji+e_{j_1}}) v_{(x, i+e_{j_1}, k)}^n,$$

using once or twice (3.5.3). Using (3.5.3) gives for queue j_1 :

$$\begin{aligned} \mu_{ji} v_{(x, i-e_{j_1}, k+2)}^n + (\mu - \mu_{j_1 i}) v_{(x, i, k+1)}^n &\leq \mu v_{(x, i, k+1)}^n \leq \\ \mu_{j_1 i+e_{j_1}} v_{(x, i, k+1)}^n + (\mu - \mu_{j_1 i+e_{j_1}}) v_{(x, i+e_{j_1}, k)}^n \end{aligned}$$

Again the terms concerning costs and the dummy transition follow easily. It is trivial to prove equation (3.5.4). The proofs of (3.5.5) and (3.5.2) are similar.

□

Proof of lemma 3.5.3. By induction. We follow the proof of lemma 3.5.1. First observe that

$$v_{(x,i,k+1)}^n \leq v_{(x,i+e_{j_1},k)}^n \leq v_{(x,i,k)}^n,$$

or $v_{(x,i,k+1)}^n \leq v_{(x,i-e_{j_1},k+1)}^n \leq v_{(x,i,k)}^n$ if $i = B$. Thus also (3.5.4) holds. The conditions on the arrival process and service rates are stronger than in the previous model, thus the proof of (3.5.6), (3.5.7) and (3.5.9) is equal to that of lemma 3.5.1.

Now consider equation (3.5.8). Let j^* be the optimal assignment in state (y, i, k) . If $j^* \neq j_1$ then

$$\min_j \{v_{(y,i+e_{j_1}+e_j,k)}^n\} \leq v_{(y,i+e_{j_1}+e_{j^*},k)}^n \stackrel{(3.5.8)}{\leq} v_{(y,i+e_{j^*},k)}^n = \min_j \{v_{(y,i+e_j,k)}^n\},$$

if $j^* = j_1$ then

$$\min_j \{v_{(y,i+e_{j_1}+e_j,k)}^n\} \leq v_{(y,i+e_{j_1},k)}^n = \min_j \{v_{(y,i+e_j,k)}^n\}.$$

Because $q_{xay;i} = q_{xay}$, the terms on arrivals follow.

The departure terms from each queue, except queue j_1 , follow easily with induction. Note that we use here that $\mu_{ji+e_{j_1}} = \mu_{ji}$ if $j \neq j_1$. Concerning queue j_1 , we have

$$\mu_{j_1 i+e_{j_1}} v_{(x,i,k+1)}^n \leq \mu_{j_1 i} v_{(x,i-e_{j_1},k+1)}^n + (\mu_{j_1 i+e_{j_1}} - \mu_{j_1 i}) v_{(x,i,k)}^n,$$

by (3.5.8) and (3.5.4). □

Proof of lemma 3.6.1. The proof goes by induction. Assume that the lemma holds up to n . First we show that the SIP is optimal for $n+1$. Consider two server assignments, which differ only in the assignment of 2 servers, say server k_1 and k_2 , which are assigned to queue j_1 and j_2 . In one assignment server k_1 is assigned to queue j_1 and server k_2 to j_2 , in the other assignment v.v. The difference between the departure terms is

$$(p_{k_1}(x) - p_{k_2}(x)) \left(\mu_{j_1} v_{(x,i-e_{j_1})}^n + (\mu - \mu_{j_1}) v_{(x,i)}^n - \mu_{j_2} v_{(x,i-e_{j_2})}^n - (\mu - \mu_{j_2}) v_{(x,i)}^n \right),$$

which is negative if $p_{k_1}(x) > p_{k_2}(x)$ and $j_1 < j_2$. Thus queue j_1 should be served by the faster server. By taking $p_{k_2}(x) = 0$ we have that serving queue j_1 is better than serving queue j_2 . Repeating this gives the optimality of the SIP.

We start with (3.6.1). Because $|i| \geq 2$ we only deal with states for which $q_{xay;i}^k = q_{xay}^k$, therefore we omit the i in the notation. The fact that $q_{xay;0}^k$ need not be equal to q_{xay}^k plays a role only in the proof of (3.6.2). Rewrite (3.6.1) as

$$\mu_{j_1} v_{(x,i-e_{j_1})}^n + (\mu_{j_2} - \mu_{j_1}) v_{(x,i)}^n \leq \mu_{j_2} v_{(x,i-e_{j_2})}^n, \quad \mu_{j_2} - \mu_{j_1} \geq 0$$

Consider the terms corresponding to arrivals. Assume a^* is the optimal action in $(x, i - e_{j_2})$. Then we have

$$\begin{aligned}
 & \mu_{j_1} \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay}^j v_{(y, i - e_{j_1} + e_j)}^n + (1 - \sum_{j=1}^m q_{xay}^j) v_{(y, i - e_{j_1})}^n \right) \right\} + \\
 & (\mu_{j_2} - \mu_{j_1}) \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay}^j v_{(y, i + e_j)}^n + (1 - \sum_{j=1}^m q_{xay}^j) v_{(y, i)}^n \right) \right\} \leq \\
 & \mu_{j_1} \sum_y \lambda_{xa^*y} \left(\sum_{j=1}^m q_{xa^*y}^j v_{(y, i - e_{j_1} + e_j)}^n + (1 - \sum_{j=1}^m q_{xa^*y}^j) v_{(y, i - e_{j_1})}^n \right) + \\
 & (\mu_{j_2} - \mu_{j_1}) \sum_y \lambda_{xa^*y} \left(\sum_{j=1}^m q_{xa^*y}^j v_{(y, i + e_j)}^n + (1 - \sum_{j=1}^m q_{xa^*y}^j) v_{(y, i)}^n \right) \stackrel{(3.6.1)}{\leq} \\
 & \mu_{j_2} \sum_y \lambda_{xa^*y} \left(\sum_{j=1}^m q_{xa^*y}^j v_{(y, i - e_{j_2} + e_j)}^n + (1 - \sum_{j=1}^m q_{xa^*y}^j) v_{(y, i - e_{j_2})}^n \right) = \\
 & \mu_{j_2} \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay}^j v_{(y, i - e_{j_2} + e_j)}^n + (1 - \sum_{j=1}^m q_{xay}^j) v_{(y, i - e_{j_2})}^n \right) \right\}.
 \end{aligned}$$

Note that we used $\mu_{j_1} \leq \mu_{j_2}$ explicitly here; if $\mu_{j_1} > \mu_{j_2}$ there would have been 2 expressions (with positive coefficients) on the r.h.s. and there would not have been 1 minimizing action. We would not have this problem if there were no actions to choose, i.e. if the arrivals are independent.

Consider the terms concerning departures. We write p_j instead of $p_j(x)$, and assume that $p_1 \geq \dots \geq p_s$. We also assume $|i| \geq s + 1$. We distinguish two cases. First assume that there are customers in queues $j_1^* \leq \dots \leq j_s^*$ present in state $i - e_{j_1}$ with $j_s^* \leq j_1$. Because $j_1 < j_2$, the same action, say j_1^*, \dots, j_s^* , is optimal in $i, i - e_{j_1}$ and $i - e_{j_2}$. We have for $1 \leq k \leq s$:

$$\begin{aligned}
 & \mu_{j_1} \mu_{j_k^*} v_{(x, i - e_{j_1} - e_{j_k^*})}^n + (\mu_{j_2} - \mu_{j_1}) \mu_{j_k^*} v_{(x, i - e_{j_k^*})}^n + \\
 & \mu_{j_1} (\mu - \mu_{j_k^*}) v_{(x, i - e_{j_1})}^n + (\mu_{j_2} - \mu_{j_1}) (\mu - \mu_{j_k^*}) v_{(x, i)}^n \stackrel{(3.6.1)}{\leq} \\
 & \mu_{j_2} \mu_{j_k^*} v_{(x, i - e_{j_2} - e_{j_k^*})}^n + \mu_{j_2} (\mu - \mu_{j_k^*}) v_{(x, i - e_{j_2})}^n.
 \end{aligned} \tag{4.3.3}$$

Now the departure terms of (3.6.1) follow easily:

$$\begin{aligned}
 & \mu_{j_1} \min_{l_1, \dots, l_s} \left\{ \sum_{k=1}^s \left(p_k \mu_{l_k} v_{(x, i - e_{j_1} - e_{l_k})}^n + p_k (\mu - \mu_{l_k}) v_{(x, i - e_{j_1})}^n \right) \right\} + \\
 & (\mu_{j_2} - \mu_{j_1}) \min_{l_1, \dots, l_s} \left\{ \sum_{k=1}^s \left(p_k \mu_{l_k} v_{(x, i - e_{l_k})}^n + p_k (\mu - \mu_{l_k}) v_{(x, i)}^n \right) \right\} =
 \end{aligned}$$

$$\begin{aligned}
& \sum_{k=1}^s \left\{ p_k \mu_{j_1} \mu_{j_k}^* v_{(x, i-e_{j_1}-e_{j_k}^*)}^n + p_k (\mu_{j_2} - \mu_{j_1}) \mu_{j_k}^* v_{(x, i-e_{j_k}^*)}^n \right\} + \\
& \sum_{k=1}^s \left\{ p_k \mu_{j_1} (\mu - \mu_{j_k}^*) v_{(x, i-e_{j_1})}^n + p_k (\mu_{j_2} - \mu_{j_1}) (\mu - \mu_{j_k}^*) v_{(x, i)}^n \right\} \stackrel{(4.3.3)}{\leq} \\
& \sum_{k=1}^s p_k \mu_{j_2} \mu_{j_k}^* v_{(x, i-e_{j_2}-e_{j_k}^*)}^n + \sum_{k=1}^s p_k \mu_{j_2} (\mu - \mu_{j_k}^*) v_{(x, i-e_{j_2})}^n = \\
& \mu_{j_2} \min_{l_1, \dots, l_s} \left\{ \sum_{k=1}^s \left(p_k \mu_{l_k} v_{(x, i-e_{j_2}-e_{l_k})}^n + p_k (\mu - \mu_{l_k}) v_{(x, i-e_{j_2})}^n \right) \right\}.
\end{aligned}$$

Concerning the second case, assume that all class j_1 customers are served in state i . Consider the optimal assignment in $i - e_{j_2}$, being $j_1^* \leq \dots \leq j_s^*$ with $j_{s_1}^* = j_1$. Assign server s_1 in both i and $i - e_{j_1}$ to queue j_2 and all other servers to the same queues as in $i - e_{j_2}$. Then (4.3.3) holds for server $1, \dots, s_1 - 1, s_1 + 1, \dots, s$. For server s_1 we have

$$\begin{aligned}
& \mu_{j_1} \mu_{j_2} v_{(x, i-e_{j_1}-e_{j_2})}^n + \mu_{j_1} (\mu - \mu_{j_2}) v_{(x, i-e_{j_1})}^n + \\
& (\mu_{j_2} - \mu_{j_1}) \mu_{j_2} v_{(x, i-e_{j_2})}^n + (\mu_{j_2} - \mu_{j_1}) (\mu - \mu_{j_2}) v_{(x, i)}^n \stackrel{(3.6.1)}{\leq} \\
& \mu_{j_2} \mu_{j_1} v_{(x, i-e_{j_2}-e_{j_1})}^n + \mu_{j_2} (\mu - \mu_{j_1}) v_{(x, i-e_{j_2})}^n.
\end{aligned}$$

The terms concerning departures follow in the same way as in the first case.

Now assume $2 \leq |i| \leq s$. Suppose that server s_1 is assigned to queue j_1 in state $i - e_{j_2}$. The term concerning server s_1 is similar to the corresponding term in the previous case, and in state i we keep one customer unserved. Again, the terms concerning departures follow easily.

It remains to study the dummy term, which goes by induction.

We continue with (3.6.2), which is much easier to prove. Let a^* be the optimal action for the MDAP in (x, i) . Note that $q_{xay; i-e_{j_1}}^j \leq q_{xay; i}^j$. Then we have

$$\begin{aligned}
& \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay; i-e_{j_1}}^j v_{(y, i-e_{j_1}+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xay; i-e_{j_1}}^j \right) v_{(y, i-e_{j_1})}^n \right) \right\} \leq \\
& \sum_y \lambda_{xa^*y} \left(\sum_{j=1}^m q_{xa^*y; i-e_{j_1}}^j v_{(y, i-e_{j_1}+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xa^*y; i-e_{j_1}}^j \right) v_{(y, i-e_{j_1})}^n \right) \stackrel{(3.6.2)}{\leq} \\
& \sum_y \lambda_{xa^*y} \left(\sum_{j=1}^m q_{xa^*y; i}^j v_{(y, i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xa^*y; i}^j \right) v_{(y, i)}^n \right) = \\
& \min_a \left\{ \sum_y \lambda_{xay} \left(\sum_{j=1}^m q_{xay; i}^j v_{(y, i+e_j)}^n + \left(1 - \sum_{j=1}^m q_{xay; i}^j \right) v_{(y, i)}^n \right) \right\}.
\end{aligned}$$

Let j_1^*, \dots, j_s^* be the optimal assignment in (x, i) . If j_1 does not belong to this action, we have

$$\mu_k v_{(x, i - e_{j_1} - e_k)}^n + (\mu - \mu_k) v_{(x, i - e_{j_1})}^n \stackrel{(3.6.2)}{\leq} \mu_k v_{(x, i - e_k)}^n + (\mu - \mu_k) v_{(x, i)}^n \quad (4.3.4)$$

for $k = j_1^*, \dots, j_s^*$. Summing gives the expression wanted. If j_1 does belong to the optimal action in (x, i) , say $j_1 = j_s^*$, take the suboptimal action $j_1^*, \dots, j_{s-1}^*, 0$ in $(x, i - e_{j_1})$. We have (4.3.4) for $k = j_1^*, \dots, j_{s-1}^*$. For the last server we have

$$\mu v_{(x, i - e_{j_1})}^n \stackrel{(3.6.2)}{\leq} \mu_{j_1} v_{(x, i - e_{j_1})}^n + (\mu - \mu_{j_1}) v_{(x, i)}^n.$$

Summing gives the expression for the suboptimal action. As the optimal action is even better, we have the inequality wanted. \square

Proof of lemma 3.7.1. By induction. Assume the lemma holds up to n . We start with the terms on arrivals of (3.7.1). We consider each customer separately, instead of each queue separately. All customers who are not in the queues in state i can enter the system in the states considered in (3.7.1). Their terms go directly with induction. Consider the extra customers in queue j_1 and j_2 . We have

$$\lambda_{j_2} \mu_{j_1} v_{i - e_{j_1}}^n + \lambda_{j_2} (\mu - \mu_{j_1}) v_i^n \stackrel{(3.7.1)}{\leq}$$

$$\lambda_{j_2} \mu_{j_2} v_{i - e_{j_2}}^n + \lambda_{j_2} (\mu - \mu_{j_2}) v_i^n \stackrel{(3.7.2)}{\leq}$$

$$\lambda_{j_1} \mu_{j_2} v_{i - e_{j_2}}^n + (\lambda_{j_2} \mu - \lambda_{j_1} \mu_{j_2}) v_i^n.$$

This gives

$$\lambda_{j_1} \mu_{j_1} v_i^n + \lambda_{j_2} \mu_{j_1} v_{i - e_{j_1}}^n + \lambda_{j_1} (\mu - \mu_{j_1}) v_i^n + \lambda_{j_2} (\mu - \mu_{j_1}) v_i^n \leq$$

$$\lambda_{j_1} \mu_{j_2} v_{i - e_{j_2}}^n + \lambda_{j_2} \mu_{j_2} v_i^n + \lambda_{j_1} (\mu - \mu_{j_2}) v_i^n + \lambda_{j_2} (\mu - \mu_{j_2}) v_i^n,$$

which are the terms on the extra customers.

The departure and dummy terms can be proved in a similar way as in the proof of lemma 1.11.5. We continue with (3.7.2). Again all terms follow directly by induction, with an exception for the extra customer in queue j_1 . \square

Uniformization

5.1. Introduction

The dynamic programming results of the previous chapters are obtained for discrete-time models. Here we establish, for the policies optimal in the discrete-time models, optimality at T in the continuous-time models, for all T . First we make a distinction between policies and decision rules. A decision rule is a function prescribing for each state which action to take (or more generally, for each state it is a distribution on the actions). A policy R is, in the discrete-time case, a sequence of decision rules (f_1, f_2, \dots) , with f_n the decision rule at time n . If the system is controlled continuously in $[0, \infty)$, R is a family $\{f_t, t \in [0, \infty)\}$ with f_t the decision rule at t .

We consider the following controllable model. We have a countable state space E . If action $a \in A(x)$ is chosen in $x \in E$, the system goes to y with intensity q_{xay} . We assume that there is a constant α such that $\sum_y q_{xay} \leq \alpha$ for all $a \in A(x)$, $x \in E$. A model satisfying this condition is called uniformizable. We will consider this model for various types of cost functions.

Unfortunately not all models considered in the previous chapters conform to this description. Particularly, in the customer assignment models we first choose an action in the arrival process. Then, immediately after a transition in the arrival process, the assignment action has to be chosen, possibly depending on the state of the arrival process just reached. To be able to use the results of the forthcoming sections, we rewrite it in the standard form as follows. In state (x, i) (with x the state of the arrival process) we have as possible actions $(a, j_z; z \in \Lambda)$ with a in the action set of the arrival process and $1 \leq j_z \leq m$ for all z , and j_z an allowable action in i . Here a is the action in the arrival process, and j_z is the queue to assign the arriving customer to if the arrival process moves to z . Thus each action has $|\Lambda| + 1$ components, giving the action sets $A((x, i))$. The non-negative transition intensities are for example in the model of section 2.2:

$$q_{(x,i)(a,j_z)(y,i+e_j)} = \lambda_{xay} q_{xay} \text{ if } j_y = j$$

$$q_{(x,i)(a,j_z)(y,i)} = \lambda_{xay} (1 - q_{xay})$$

$$q_{(x,i)(a,j_z)(x,i-e_j)} = \mu \text{ if } i_j > 0$$

$$q_{(x,i)(a,j_z)(x,i)} = 1 - \sum_y \lambda_{xay} - \mu \sum_j \delta_{i_j}$$

If the model is uniformizable, we can rewrite the dynamic programming equation of the embedded discrete-time chain in the form of (2.2.1):

$$\begin{aligned}
v_{(x,i)}^{n+1} = & \min_{(a,j_z; z \in \Lambda)} \left\{ \sum_y \left(q_{(x,i)(a,j_z)(y,i+e_{j_y})} v_{(y,i+e_{j_y})}^n + q_{(x,i)(a,j_z)(y,i)} v_{(y,i)}^n \right) + \right. \\
& \sum_j \delta_{i_j} q_{(x,i)(a,j_z)(x,i-e_j)} v_{(x,i-e_j)}^n + q_{(x,i)(a,j_z)(x,i)} v_{(x,i)}^n \left. \right\} = \\
& \min_a \left\{ \sum_y \min_{j_y} \left\{ \lambda_{xay} \left(q_{xay} v_{(y,i+e_{j_y})}^n + (1 - q_{xay}) v_{(y,i)}^n \right) \right\} \right\} + \\
& \sum_j \delta_{i_j} \mu v_{(x,i-e_j)}^n + (1 - \sum_y \lambda_{xay} - \mu \sum_j \delta_{i_j}) v_{(x,i)}^n = \\
& \min_a \left\{ \sum_y \lambda_{xay} \left(q_{xay} \min_j \{ v_{(y,i+e_j)}^n \} + (1 - q_{xay}) v_{(y,i)}^n \right) \right\} + \\
& \sum_j \delta_{i_j} \mu v_{(x,i-e_j)}^n + (1 - \sum_y \lambda_{xay} - \mu \sum_j \delta_{i_j}) v_{(x,i)}^n.
\end{aligned}$$

A disadvantage of this way of rewriting is the fact that models that originally had only finite action sets now have infinite action sets.

A way to get around this problem is to allow for 2 transitions immediately after each other at the jump times. We illustrate this idea again with the model of section 2.2. Let the jump times be exponentially distributed with rate $\alpha = \gamma + m\mu$. Assume that the process is in (x, i) . An action $a \in A(x)$ is selected and we have as transition probabilities p for the first jump:

$$\begin{aligned}
p_{(x,i)a(y,i,0)} &= \frac{\lambda_{xay}}{\alpha} q_{xay} \\
p_{(x,i)a(y,i,1)} &= \frac{\lambda_{xay}}{\alpha} (1 - q_{xay}) \\
p_{(x,i)a(x,i,2)} &= m \frac{\mu}{\alpha} \\
p_{(x,i)a(x,i,3)} &= 1 - \frac{\lambda_{xay}}{\alpha} - m \frac{\mu}{\alpha}
\end{aligned}$$

The third component of the state indicates the event at the immediate second transition, with probabilities denoted \tilde{p} . Take j as the assignment action.

$$\begin{aligned}
\tilde{p}_{(x,i,0)j(x,i+e_j)} &= 1 \\
\tilde{p}_{(x,i,1)j(x,i)} &= 1 \\
\tilde{p}_{(x,i,2)j(x,(i-e_{j_1})^+)} &= \frac{1}{m}
\end{aligned}$$

$$\tilde{p}_{(x,i,3)j(x,i)} = 1$$

Here we also show that equation (2.2.1) can be obtained. Let w^n be the value function for the present model. We will show that if $w^{2n} = v^n$ in states of the form (x, i) , then $w^{2n+2} = v^{n+1}$. For ease of notation, assume $\alpha = 1$.

$$\begin{aligned} w_{(x,i)}^{2n+2} &= \min_a \left\{ \sum_y \lambda_{xay} q_{xay} w_{(y,i,0)}^{2n+1} + \sum_y \lambda_{xay} (1 - q_{xay}) w_{(y,i,1)}^{2n+1} + \right. \\ &\quad \left. m\mu w_{(x,i,2)}^{2n+1} + (1 - \sum_y \lambda_{xay} - m\mu) w_{(x,i,3)}^{2n+1} \right\} = \\ &= \min_a \left\{ \sum_y \lambda_{xay} q_{xay} \min_j \{ w_{(y,i+e_j)}^{2n} \} + \sum_y \lambda_{xay} (1 - q_{xay}) w_{(y,i,1)}^{2n} + \right. \\ &\quad \left. \mu \sum_j w_{(x,(i-e_j)+)}^{2n} + (1 - \sum_y \lambda_{xay} - m\mu) w_{(x,i)}^{2n} \right\} \end{aligned}$$

This completes the induction step.

The results for the discrete-time models are of two types. First we have the models of chapter 1. The optimal policies obtained there are myopic, i.e. they have the same decision rule for all n . Continuous-time results for these models are obtained in the next section.

In section 5.3 models with horizon-dependent optimal policies are studied. Here extra conditions are necessary to obtain optimality at T .

5.2. Uniformization with fixed parameter

In this section we assume that we have a uniformizable problem, and that the optimal policy of the discrete-time model is myopic and independent of the uniformization parameter. Furthermore, assume that the costs are bounded, either from above or below. Consider a model in which there are only costs at T , thus the problem is how to control the model from 0 to T . We call the class of policies in the continuous-time model that only can change actions at the jump times the *semi-Markov policies*. Note that this is not a restriction for the customer assignment models, because there the only action of importance is the one taken at the jump times. In the server assignment models however, it is a restriction. Denote with ϕ^T ($\phi^T(R)$) the minimal costs (the costs using policy R) at T . Let R^* be the policy with $f_t^* = f^*$ for all t , with f^* the optimal decision rule in the discrete-time model. Then we have the following.

5.2.1. Theorem. $\phi^T(R^*) \leq \phi^T(R)$ for each semi-Markov policy R .

Proof. The evolution of the process is completely described by two independent random processes: the Poisson process generating the transition times and the embedded chain generating the actual transitions at the transition times. Note that the former process does not depend on the policy chosen,

the latter however does. We condition on the transition times, both for R and R^* . Let $\omega \in \Omega$ be a realization of the transition time process with probability space (Ω, P) , with $\omega = (t_1, \dots, t_u)$, $0 \leq t_1 \leq \dots \leq t_u \leq T$, the t 's being the transition times. Note that u is a realization of a Poisson distributed random variable. The decision rule used at t by R is completely determined by the jump times before t and the states at these moments. This induces a policy R_ω in the embedded chain. Note that R_ω^* does not depend on ω , therefore we use R^* also for the discrete-time policy. Denote by $v_x^n(R_\omega, \omega)$ the value function of the embedded chain (not necessarily in the standard form). By the optimality of R^* we have $v_x^n(R^*, \omega) = v_x^n \leq v_x^n(R_\omega, \omega)$. Denote by $\phi_x^T(R)$ the expected costs at T using R and starting in x . Then

$$\phi_x^T(R^*) = \int_{\Omega} v_x^u dP(\omega) \leq \int_{\Omega} v_x^u(R_\omega, \omega) dP(\omega) = \phi_x^T(R).$$

Note that, although $\phi_x^T(R)$ can be infinite, it is well defined due to the boundedness of the costs and therefore of the v^n . \square

The optimal policy also minimizes the costs from 0 up to T , because that is the integral over the costs from 0 to T . Thus, we do not need to introduce immediate costs in the dynamic programming equation.

The process just described is called uniformization. It is essential that the rate out of each state is uniformly bounded (otherwise we cannot formulate the discrete-time dynamic programming equations) and that the policy R^* is the same for each n . In the models of the chapters 2 and 3 the latter condition is not satisfied, giving need for a limiting argument, which is the subject of the next section.

Note that if a policy is stochastically optimal in the discrete-time model, it is also stochastically optimal in the continuous-time model.

Summarizing, we have the following.

5.2.2. Corollary. *The policies minimizing the costs in the discrete-time models considered in chapter 1 minimize the costs at T (from 0 to T) in the continuous-time models in the class of semi-Markov policies, if the costs are bounded, either from above or below.*

The decisions are taken on the Poisson epochs, even if there is a dummy transition. By increasing the uniformization parameter we add decision epochs. This way we can approximate continuous-time control. Roughly speaking the class of limiting policies are called *strongly regular* in Hordijk & Van der Duyn Schouten [29]. More precisely, a policy is strongly regular if for almost all sample paths the time points at which the control is discontinuous has Lebesgue measure zero. See Hordijk & Van der Duyn Schouten [29] for details.

5.2.3. Corollary. *The policies minimizing the costs in the discrete-time models considered in chapter 1 minimize the costs at T (from 0 to T) in the*

continuous-time models in the class of strongly regular policies, if the costs are bounded, either from above or below.

Also results on discounted and average costs can be obtained. It is clear from theorem 5.2.1 that R^* minimizes $\int_0^T e^{-\beta t} \phi_x^t(R) dt$ and $\frac{1}{T} \int_0^T \phi_x^t(R) dt$ and their limits for $T \rightarrow \infty$, if they exist.

Usually definitions for discounted and average optimality other than the ones given above, using semi-Markov Decision Processes, are used. We can translate the continuous-time problems into discrete-time ones, like the ones we studied in chapter 1. See for example Serfoso [63] for this equivalence. Then, under suitable conditions guaranteeing the convergence of the successive approximation scheme, optimality of R^* for average and discounted optimality follows. Convergence of successive approximation can be proved for example using negative dynamic programming (Ross [60]) or by showing ν -geometric recurrence (Spieksma [69]).

A complication using successive approximation is that the analysis of chapter 1 only considers costs at the end of the horizon, as we took v^n of the form $v^n = \inf_f \{P(f)v^{n-1}\}$, $v^0 = c$, with $P(f)$ the transition matrix under decision rule f . However, we are interested in $w^n = \inf_f \{c + \beta P(f)w^{n-1}\}$ with $w^0 = 0$. By the assumption of this section that the optimal policy is myopic, we have $w^n = v^0 + \dots + \beta^{n-1}v^{n-1}$. If $R^* = (f, f, \dots)$ is the optimal policy, this gives us for arbitrary R

$$w^n(R^*) = w^n \leq w^n(R).$$

5.3. Continuous-time Bellman equation

In this section we give another approach to continuous-time control. We show that under mild conditions on the cost functions the solutions of the dynamic programming equations converge to the solution of the continuous-time Bellman equation. Hence the structure of the optimal value functions carry over to the continuous-time model, and therefore so does the structure of the optimal policy.

We can use this method not only in the models of chapter 1, but also in the (non-myopic) models of the chapters 2 and 3. The method is usually referred to as time-discretization. Our analysis is based on the results of Van Dijk [73], as he allows for both positive and negative unbounded costs. Besides this he considers salvage costs. We rewrite his results here for the simpler case of a uniformizable model. Denote with $G(f)$ the infinitesimal generator of the process if decision rule f is used.

For the customer assignment model we would have the following generator. Assume the current state is (x, i) and $f(x, i) = (a, j_z)$. Then, for example

$$G(f)_{(x,i)(y,i+e_j)} = \lambda_{xay}q_{xay} \text{ if } j = j_y,$$

$$G(f)_{(x,i)(x,i-e_j)} = \mu \text{ if } i_j > 0$$

and

$$G(f)_{(x,i)(x,i)} = -\left(\sum_y \lambda_{xay} + \mu \sum_{j=1}^m \delta_{ij}\right).$$

Basic to the analysis is the continuous-time Bellman equation. Heuristically, this equation can be derived as follows. Let ϕ^t denote again the expected costs for horizon t . We are interested in ϕ^T , thus ϕ^t are in fact the expected costs from $T - t$ to T . Assume that continuously over time costs with rate c are incurred, ϕ^0 are the costs at the end. Then we have:

$$\frac{d}{dt}\phi^s = \inf_f \{c + G(f)\phi^s\}.$$

Integrating from 0 to t gives

$$\phi^t - \phi^0 = \int_0^t \inf_f \{c + G(f)\phi^s\} ds,$$

the Bellman equation. Note that we have a model with immediate costs, as contrasted with the model used in uniformization. We need to do it this way because we cannot introduce immediate costs afterwards, for the same reason that we cannot use uniformization with a fixed parameter here. That is, for minimizing costs at different T we have different optimal policies.

Now we introduce our computational scheme. We assume that the model is uniformizable, i.e. there is a constant α such that for each state x and decision rule f we have $|G(f)_{xx}| \leq \alpha$. Let h be a positive number, $h \leq 1/\alpha$. Define $P^h(f) = hG(f) + I$. Thus $P^h(f)$ is the transition matrix of the discrete-time model obtained by uniformization with parameter $1/h$. Take hc as immediate costs. Now define

$$v^{h,n+1} = \inf_f \{hc + P^h(f)v^{h,n}\}, \quad h(n+1) \leq T,$$

$$v^{h,0} = \phi^0.$$

We will show that $v^{h,k}$ with $k = \lfloor t/h \rfloor$, $t \leq T$, converges as $h \rightarrow 0$ to the solution of the Bellman equation under certain conditions. Heuristically, when seen as uniformization, this can be explained by noting that the number of jumps before T converges to a constant as h decreases (which follows from lemma A.2). When seen as discretization, $v^{h,n}$ is the first order approximation of the costs at hn . By the infinitesimal properties, the transition rates converge to their first order approximations as $h \rightarrow 0$.

The conditions involve the weighted supremum norm, defined as follows: $\|b\|_\nu = \sup_x |b_x|/\nu_x$ with $\nu > 0$ the bounding vector. For a matrix the norm is defined as follows: $\|A\|_\nu = \sup_x \sum_y |A_{xy}| \nu_y / \nu_x$. We will often use that $\|Ab\|_\nu \leq \|A\|_\nu \|b\|_\nu$. We assume the following.

5.3.1. Assumption. There are $\nu \geq 1$, constants K_1 and K_2 such that

$$\|G(f)\|_\nu \leq K_1 \text{ for all } f, \|c\|_\nu \leq K_2 \text{ and } \|\phi^0\|_\nu \leq K_2.$$

We will check these conditions for various cost functions. If ϕ^0 is an indicator function and $c = 0$, we take $\nu = e$. Then $\|G(f)\|_\nu \leq 2\alpha$ and $\|\phi^0\|_\nu \leq 1$. In the customer assignment model with $c_{(x,i)}$ or $\phi_{(x,i)}^0 = i_1 + \dots + i_m$, take $\nu_{(x,i)} = i_1 + \dots + i_m \vee 1$. Then $\|G(f)\|_\nu \leq 2\alpha$ and $\|c\|_\nu, \|\phi^0\|_\nu \leq 1$. In the server assignment models with $c_{(x,i)}$ or $\phi_{(x,i)}^0 = i_1 c_1 + \dots + i_m c_m$ take $\nu_{(x,i)} = (i_1 |c_1| + \dots + i_m |c_m|) \vee 1$. Because $\nu_{(x,i+e_j)}/\nu_{(x,i)}$ and $\nu_{(x,i-e_j)}/\nu_{(x,i)} \leq 1 + \max_j |c_j|$, $\|G(f)\|_\nu \leq 2\alpha(1 + \max_j |c_j|)$ and $\|c\|_\nu, \|\phi^0\|_\nu \leq (1 + \max_j |c_j|)$. Similarly, if $c_{(x,i)}$ or $\phi_{(x,i)}^0 = (i_1 c_1 + \dots + i_m c_m)^n$ take $\nu_{(x,i)} = (i_1 |c_1| + \dots + i_m |c_m|)^n \vee 1$, giving $\|G(f)\|_\nu \leq 2\alpha(1 + \max_j |c_j|)^n$ and $\|c\|_\nu, \|\phi^0\|_\nu \leq (1 + \max_j |c_j|)^n$.

5.3.2. Theorem. There are $\hat{\phi}_x^t$ such that $v_x^{h,[t/h]} \rightarrow \hat{\phi}_x^t$ with $h = 2^{-m}$ as $m \rightarrow \infty$, for $t \leq T$ and all i .

Proof. First we show that all $v^{h,n}$ with $hn \leq T$ are ν -bounded:

$$\begin{aligned} \|v^{h,m}\|_\nu &\leq \sup_f \|hc + P^h(f)v^{h,m-1}\|_\nu \leq \\ h \sup_f \|c\|_\nu + \sup_f \|P^h(f)\|_\nu \|v^{h,m-1}\|_\nu &\leq hK_2 + (1 + hK_1)\|v^{h,m-1}\|_\nu \end{aligned} \quad (5.3.1)$$

Now we have, since $1 + c \leq e^c$,

$$\|v^{h,n}\|_\nu \leq \sum_{k=0}^{n-1} (1 + hK_1)^k hK_2 + (1 + hK_1)^n \|v^{h,0}\|_\nu \leq Te^{TK_1} K_2 + e^{TK_1} K_2. \quad (5.3.2)$$

Let us denote the r.h.s. by C_1 .

We will prove the convergence by first deriving a relation between $v^{h,n}$ and $v^{h/2,2n}$. By induction on n we prove

$$v^{h/2,2n} \leq v^{h,n} + \nu(1 + hK_1)^n (nh^2 + h)C_2 \quad (5.3.3)$$

for $C_2 \geq K_1 K_2 + K_1^2 C_1$. Assume the inequality holds up to k .

$$\begin{aligned} v^{h/2,2k+2} &= \inf_f \{h/2c + P^{h/2}(f)v^{h/2,2k+1}\} \leq \\ \inf_f \{h/2(I + P^{h/2}(f))c + P^{h/2}(f)P^{h/2}(f)v^{h/2,2k}\} &= \\ \inf_f \{(h + (h/2)^2 G(f))c + (P^h(f) + (h/2)^2 (G(f))^2)v^{h/2,2k}\} &\leq \\ \inf_f \{hc + P^h(f)v^{h/2,2k}\} + \sup_f \{(h/2)^2 G(f)c + \sup_f \{(h/2)^2 (G(f))^2 v^{h/2,2k}\}\} &\leq \end{aligned}$$

$$v^{h,k+1} + \nu(1 + hK_1)^{k+1}(kh^2 + h)C_2 + \sup_f \{(h/2)^2 G(f)c\} + \sup_f \{(h/2)^2 (G(f))^2 v^{h/2,2k}\}.$$

We have

$$\|(h/2)^2 G(f)c\|_\nu \leq (h/2)^2 K_1 K_2,$$

and, using (5.3.2),

$$\|(h/2)^2 (G(f))^2 v^{h/2,2k}\|_\nu \leq (h/2)^2 K_1^2 \|v^{h/2,2k}\|_\nu \leq h^2 K_1^2 C_1,$$

giving

$$\begin{aligned} \sup_f \{(h/2)^2 G(f)c\} + \sup_f \{(h/2)^2 (G(f))^2 v^{h/2,2k}\} &\leq \\ \nu h^2 (K_1 K_2 + K_1^2 C_1) &\leq \nu h^2 C_2. \end{aligned}$$

Thus, because $(1 + hK_1)^k \geq 1$, the inequality holds.

Because $\lfloor 2t/h \rfloor = 2\lfloor t/h \rfloor$ or $2\lfloor t/h \rfloor + 1$, we have

$$v^{h/2, \lfloor 2t/h \rfloor} \leq v^{h/2, 2\lfloor t/h \rfloor} + \nu h (K_1 C_1 + K_2),$$

by (5.3.1). Thus

$$v^{h/2, \lfloor 2t/h \rfloor} \leq v^{h, \lfloor t/h \rfloor} + \nu \left(\lfloor \frac{t}{h} \rfloor h^2 + h \right) C,$$

if $C \geq e^{TK_1} C_2 + K_1 C_1 + K_2$.

Iterating this last inequality k times, for h of the form 2^{-m} , gives

$$\begin{aligned} v^{2^{-(m+k)}, \lfloor t/2^{-(m+k)} \rfloor} &\leq v^{2^{-m}, \lfloor t/2^{-m} \rfloor} + \sum_{l=0}^{k-1} \nu (T+1) C 2^{-m} 2^{-l} \leq \\ &v^{2^{-m}, \lfloor t/2^{-m} \rfloor} + \nu (T+1) C 2^{-m+1}. \end{aligned}$$

Because the space of vectors with bounded ν -norm is a Banach space, we have that $v_x^{h, \lfloor t/h \rfloor}$ for each x has at least one limit point. To show that there is a unique limit point, suppose that, for fixed x , v'_x and v''_x are limit points, with $v'_x < v''_x$. Take $\varepsilon < (v''_x - v'_x)/3$, and m such that $|v_x^{2^{-m}, \lfloor t/2^{-m} \rfloor} - v'_x| < \varepsilon$ and $\nu_x (T+1) C 2^{-m+1} < \varepsilon$. Then $v_x^{2^{-(m+k)}, \lfloor t/2^{-(m+k)} \rfloor} < v''_x - \varepsilon$ for all k . Hence v''_x is not a limit point. \square

5.3.3. Theorem. *The function $\hat{\phi}^t$ is a solution of the Bellman equation.*

Proof. We have, for $h = 2^{-m}$,

$$v^{h, n+1} - v^{h, n} = h \inf_f \{G(f)v^{h, n}\},$$

and thus

$$v^{h, \lfloor t/h \rfloor} - v^{h, 0} = \int_0^{h \lfloor t/h \rfloor} \inf_f \{G(f)v^{h, \lfloor s/h \rfloor}\} ds.$$

The left hand side converges to $\hat{\phi}^t - \phi^0$ for each i . By dominated convergence the r.h.s. converges to $\int_0^t \inf_f \{G(f)\hat{\phi}^s\} ds$ for fixed i , giving the Bellman equation. For further details, we refer to Van Dijk [73]. \square

We will not go into the details of showing that this solution is unique. Due to the finite action sets of the models we consider the infima are always attained and optimal policies exist. And as the discrete-time value functions converges to the continuous-time value function, the inequalities we typically prove for the discrete-time models also hold for the continuous-time models. This means that the optimality results also hold for the continuous-time models. As we considered both terminal costs and costs over time, the results hold both for costs at T and for costs over time.

5.3.4. Corollary. *The policies minimizing the costs in the discrete-time models considered in chapters 2 and 3 minimize the costs at T (from 0 to T) in the continuous-time models, if the transition rates and costs satisfy assumption 5.3.1.*

Regarding discounted and average optimality, the same remarks as in the previous section apply here.

Appendix A

The approximation of point processes

Here we show that any marked arrival stream can be approximated in the sense of weak convergence by a series of MAP's. For stationary point processes this has already been proven by Herrmann [20]. Note that we used the marks in the previous chapters to indicate the class of arrivals and server vacations. We use the following definition of an MAP:

A.1. Definition. (Markov Arrival Process) Let Λ be the, possibly countable, state space of a Markov process with transition rates λ_{xy} , $x, y \in \Lambda$. When this process moves from x to y with probability q_{xy}^v an arrival with mark v occurs, with $\sum_{v \in B} q_{xy}^v \leq 1$ for all $x, y \in \Lambda$ and $B \subset \mathbb{R}_+$. The triple (Λ, λ, q) is an MAP.

A series of random variables $X^m = \{(X_n^m)\}_{n \leq N}$ on \mathbb{R}^N converges weakly to $X = \{(X_n)\}_{n \leq N}$, notated as $X^m \xrightarrow{\mathcal{D}} X$, if $\mathbb{E}f(X^m) \rightarrow \mathbb{E}f(X)$ for all continuous and bounded f . Assume X^m (X) has distribution function F_m (F).

In Schassberger [62] it is shown, for $N = 1$, that to have $X^m \xrightarrow{\mathcal{D}} X$, we can take X^m such that

$$F_m(x) = F(0) + \sum_{k=1}^{\infty} \left(F\left(\frac{k}{m}\right) - F\left(\frac{k-1}{m}\right) \right) E_m^k(x), \quad (\text{A.1})$$

where $E_m^k(x)$ is the d.f. of a gamma distributed r.v. with k phases and intensity m , i.e. $E_m^k(x) = \sum_{l=k}^{\infty} e^{-mx} \frac{(mx)^l}{l!}$, the probability that a Poisson(mx) distributed r.v. has k or more successes. The result holds also if the mass at 0 is omitted and if the mixture is taken finite, e.g.

$$F_m(x) = F\left(\frac{1}{m}\right) E_m^1(x) + \sum_{k=2}^{m^2-1} \left(F\left(\frac{k}{m}\right) - F\left(\frac{k-1}{m}\right) \right) E_m^k(x) + \left(1 - F\left(\frac{m^2-1}{m}\right) \right) E_m^{m^2}(x).$$

An heuristic explanation is easily given. The mass in a small interval, in the limit each point, is approximated by a series of gamma distributions with equal mean and increasing intensity. Such a series converge to their mean a.s.

A similar result can be obtained for finite-dimensional r.v.'s. This has already been shown in lemma 6.1 of Hordijk & Schassberger [28]. We give a

different proof here. First we construct X^m . Define

$$C_m(k) = \begin{cases} [0, \frac{1}{m}] & \text{if } k = 1, \\ (\frac{k-1}{m}, \frac{k}{m}] & \text{if } k \in \{2, \dots, m^2 - 1\}, \\ (\frac{m^2-1}{m}, \infty) & \text{if } k = m^2. \end{cases}$$

Now we have as approximation

$$F_m(x) = \sum_{\substack{1 \leq k_j \leq m^2 \\ j=1, \dots, N}} \mathbb{P}(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)) \prod_{j=1}^N E_m^{k_j}(x_j). \quad (\text{A.2})$$

We see that the mass of each cube with length of the sides $\frac{1}{m}$ is put on the upper corner, say x' . Then each component x'_j of this vector is approximated by an independent gamma distribution with parameter m and mx'_j phases, giving an expectation of x'_j .

A.2. Lemma. $X^m \xrightarrow{\mathcal{D}} X$.

Proof. It is well known that weak convergence is equivalent with convergence of the d.f. in each continuity point of F . Take such a point x . Choose an $\varepsilon > 0$. By continuity there exists a $\delta > 0$ such that $|F(x+s) - F(x)| \leq \frac{\varepsilon}{N+4}$ if $|s| < \delta$.

Now assume the integer l , a power of 2, is large enough such that $\sqrt{N}/l < \delta$ and $\frac{l^2-1}{l} > \max_i x_i$. The first condition guarantees there are vectors $\hat{x} = (\hat{x}_1, \dots, \hat{x}_N)$ and $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_N)$ such that $\hat{x}_j l$ and $\tilde{x}_j l$ are integer, $\hat{x}_j < x_j < \tilde{x}_j$ and the product set $\prod_{j=1}^N [\hat{x}_j, \tilde{x}_j]$ is contained in the ball around x with radius δ . By the integer condition \hat{x} and \tilde{x} lie at the top corner of the cube $\prod_{j=1}^N C_l(\hat{x}_j l)$ and $\prod_{j=1}^N C_l(\tilde{x}_j l)$. As we only consider powers of 2, \hat{x} and \tilde{x} lie at corners as well if $m > l$. The second condition assures that $C_m(x_j m)$ is bounded for all j if $m \geq l$.

The sum in the definition of F_m can be split in $N+2$ parts, namely $\{k|1 \leq k_j \leq \hat{x}_j m\}$, $\{k|1 \leq k_j \leq \tilde{x}_j m, \exists j : k_j > \hat{x}_j m\}$, $\{k|k_1 > \hat{x}_1 m\}, \dots, \{k|k_N > \hat{x}_N m\}$. Note that these sets are not disjoint, the last N overlap, and that $\{k|k_j \leq \tilde{x}_j m, \exists j : k_j > \hat{x}_j m\} = \{k|1 \leq k_j \leq \tilde{x}_j m\} \setminus \{k|1 \leq k_j \leq \hat{x}_j m\}$. Now we have if $m \geq l$:

$$\begin{aligned} & \left| F_m(x) - F(x) \right| \leq \left| F(\hat{x}) - F(x) \right| + \\ & \left| \sum_{1 \leq k_j \leq \hat{x}_j m} \mathbb{P}(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)) \left(1 - \prod_{j=1}^N E_m^{k_j}(x_j) \right) \right| + \\ & \left| \sum_{\substack{1 \leq k_j \leq \tilde{x}_j m \\ \exists j : k_j > \hat{x}_j m}} \mathbb{P}(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)) \prod_{j=1}^N E_m^{k_j}(x_j) \right| + \end{aligned}$$

$$\left| \sum_{k_1 > \hat{x}_1 m} \mathbb{P}\left(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)\right) \prod_{j=1}^N E_m^{k_j}(x_j) \right| + \dots +$$

$$\left| \sum_{k_N > \hat{x}_N m} \mathbb{P}\left(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)\right) \prod_{j=1}^N E_m^{k_j}(x_j) \right|.$$

The inequality holds because the probabilities of the second term of the r.h.s. sum to $F(\hat{x})$.

We give a bound for each term. It is easily seen that $|F(\hat{x}) - F(x)| \leq \frac{\varepsilon}{N+4}$ if $m \geq l$. Consider the second term of the r.h.s. If $k_j \leq \hat{x}_j m$ then $E_m^{k_j}(x) \geq E_m^{\hat{x}_j m}(x)$. Because $\hat{x} < x$, $\lim_{m \rightarrow \infty} E_m^{\hat{x}_j m}(x) = 1$. We choose m large enough such that $F(\hat{x})(1 - \prod E_m^{k_j}(x)) \leq (1 - \prod E_m^{\hat{x}_j m}(x)) \leq \frac{\varepsilon}{N+4}$. The probabilities in the next term summed gives $F(\hat{x}) - F(\hat{x})$, therefore this term is bounded by $2\frac{\varepsilon}{N+4}$. For the last N terms we have the following. If $k_j > \hat{x}_j m$ then $E_m^{k_j}(x) \leq E_m^{\hat{x}_j m}(x)$. By choosing m large enough we have $E_m^{\hat{x}_j m}(x) \leq \frac{\varepsilon}{N+4}$ which gives the inequality wanted. All equalities summed gives $|F_m(x) - F(x)| \leq \varepsilon$. \square

We are interested in the convergence of $\{(X_n^m)\}_{n \in \mathbb{N}}$ to $\{(X_n)\}_{n \in \mathbb{N}}$. We are working in the product topology. Then the following holds (e.g. by Billingsley [6]):

$$\{(X_n^m)\}_{n \in \mathbb{N}} \xrightarrow{\mathcal{D}} \{(X_n)\}_{n \in \mathbb{N}}$$

if and only if

$$\{(X_n^m)\}_{n \leq N} \xrightarrow{\mathcal{D}} \{(X_n)\}_{n \leq N}$$

for all finite N . However, first we have to check whether $\{(X_n)\}_{n \in \mathbb{N}}$ is well defined. This is done by checking consistency of the finite dimensional r.v.s $\{(X_n^m)\}_{n \leq N}$ (see Loève [43], p. 94).

A.3. Lemma. $\{(X_n^m)\}_{n \leq N}$ are consistent for all N .

Proof. By the symmetry of (A.2) it suffices to show that the projection of $\{(X_n^m)\}_{n \leq N}$ on \mathbb{R}_+^{N-1} is equally distributed as $\{(X_n^m)\}_{n \leq N-1}$. We have, as $\cup_{k_N=1}^{m^2} C_m(k_N) = \mathbb{R}_+$,

$$\mathbb{P}\left(X_1^m \leq x_1, \dots, X_{N-1}^m \leq x_{N-1}, X_N^m \in \mathbb{R}_+\right) =$$

$$\sum_{\substack{1 \leq k_j \leq m^2 \\ j=1, \dots, N}} \mathbb{P}\left(X_1 \in C_m(k_1), \dots, X_N \in C_m(k_N)\right) \prod_{j=1}^{N-1} E_m^{k_j}(x_j) =$$

$$\sum_{\substack{1 \leq k_j \leq m^2 \\ j=1, \dots, N-1}} \mathbb{P}\left(X_1 \in C_m(k_1), \dots, X_{N-1} \in C_m(k_{N-1})\right) \prod_{j=1}^{N-1} E_m^{k_j}(x_j). \quad \square$$

It is easily seen that the lemmas remain valid if we replace (A.2) by

$$F_m(x) = \sum_{\substack{1 \leq k_j \leq m^2 \\ j=1, \dots, N}} \mathbb{P}(\dots) \prod_{\substack{j \leq N \\ j \text{ odd}}} E_m^{k_j}(x_j) \prod_{\substack{j \leq N \\ j \text{ even}}} \mathbf{I}\left\{x_j \leq \frac{k_j}{m}\right\}. \quad (\text{A.3})$$

Now consider the ∞ -dimensional r.v. $\{(S_n, V_n)\}_{n \in \mathbb{N}}$. Here S_n is the n th interarrival time, V_n is the mark belonging to the n th arrival. As $\{(S_n, V_n)\}_{n \leq N}$ is a $2N$ -dimensional r.v. we can apply the results obtained above by taking $X_{2n-1}^{(m)} = S_n^{(m)}$ and $X_{2n}^{(m)} = V_n^{(m)}$. With the superscript (m) we mean that the expression holds both with and without the superscript m . Thus $\{(S_n^m, V_n^m)\}_{n \in \mathbb{N}}$ is well defined by lemma A.3 and

$$\{(S_n^m, V_n^m)\}_{n \in \mathbb{N}} \xrightarrow{\mathcal{D}} \{(S_n, V_n)\}_{n \in \mathbb{N}}$$

holds.

Note that if $V_n \in \{1, \dots, l\}$, as in the server assignment model, then also $X_{2n}^{(m)} \in \mathbb{N}$, and $X_{2n}^{(m)} = V_n$ for m large enough, thus avoiding non-integer class numbers.

We continue by constructing an MAP (Λ, λ, q) which generates the interarrival times and marks $\{(S_n^m, V_n^m)\}_{n \in \mathbb{N}}$ for an arbitrary m . First we construct Λ . Take for each $N \in \mathbb{N}$ all vectors of the form $(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})$ with $s_n, v_n \in \{1, \dots, m^2\}$, $1 \leq \beta \leq s_N$ and $\mathbb{P}(S_n \in C_m(s_n), n \leq N; V_n \in C_m(v_n), n \leq N-1) > 0$.

Being in state $(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})$ sojourn time N is produced. The integer β indicates the current phase of the gamma distribution. The transition rates and arrival probabilities are:

$$\begin{aligned} \underline{\beta < s_N} : \quad & \lambda_{(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})(\beta+1, s_1, \dots, s_N, v_1, \dots, v_{N-1})} = m, \\ & q_{(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})(\beta+1, s_1, \dots, s_N, v_1, \dots, v_{N-1})}^v = 0. \\ \underline{\beta = s_N} : \quad & \lambda_{(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})(1, s_1, \dots, s_{N+1}, v_1, \dots, v_N)} = \\ & m \frac{\mathbb{P}(S_n \in C_m(s_n), n \leq N+1; V_n \in C_m(v_n), n \leq N)}{\mathbb{P}(S_n \in C_m(s_n), n \leq N; V_n \in C_m(v_n), n \leq N-1)}, \\ & q_{(\beta, s_1, \dots, s_N, v_1, \dots, v_{N-1})(1, s_1, \dots, s_{N+1}, v_1, \dots, v_N)}^{\frac{v_{N-1}}{m}} = 1. \end{aligned}$$

All other transition intensities are 0. Note that the transition rate out of each state is equal to m . The transition mechanism is illustrated in figure A.1. The transition marked I (III) corresponds to an arrival of a customer with mark $\frac{v_{N-1}}{m}$ ($\frac{v_N}{m}$); the next arrival will take place after s_N (s_{N+1}) phases. At transitions marked II no arrivals occur. The result proved above can easily be extended to multi-dimensional and not necessarily positive marks.

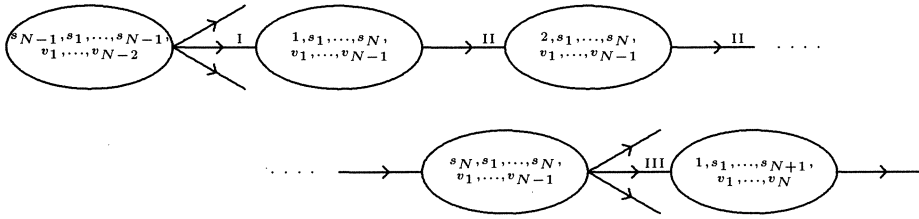


Figure A.1.

The analysis so far has to do with interarrival times and is thus in the *customer time scale*. However, weak convergence of (marked) point processes is, in general, defined in the *physical time scale*. To complete the analysis we have to prove that weak convergence of the interarrival times entails weak convergence of the point process. This result can be found in Asmussen & Koole [3].

Appendix B

Phase-type distributions of DFR/IFR distributions

Consider phase-type distributions of the form (A.1). The objective of this appendix is to give a characterization of these distributions if the approximated distribution is DFR or IFR. We will see, in the case of a decreasing (increasing) failure rate distribution, that the probability that a phase-type distribution consists of k phases, conditional that it consists of k or more phases, is decreasing (increasing) in k . (Decreasing and increasing are used in the non-strict sense.) For the DFR case our result is a special case of the characterization of Hordijk & Ridder [27].

Let F be a non-negative distribution function. For fixed m we define $\beta_1 = F(1/m)$ and $\beta_k = F(k/m) - F((k-1)/m)$ for $k > 1$. Again, let $E_m^k(x)$ be the d.f. of the gamma distribution with k phases and intensity m . Now take

$$F_m(x) = \sum_{k=1}^{\infty} \beta_k E_m^k(x).$$

It is clear, by lemma A.2, that F_m converges weakly to F . Now F_m can be seen as the time until absorption of a Markov process with initial distribution $(0, \beta_1, \beta_2, \dots)$ and transitions depicted as follows

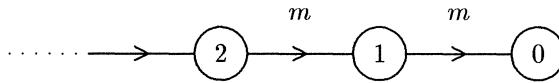


Figure B.1.

Consider the following Markov process which starts in state 1:

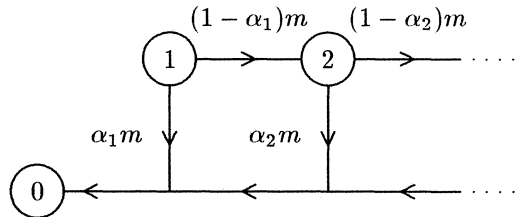


Figure B.2.

Now take

$$\alpha_n = \begin{cases} \frac{\beta_n}{1 - \sum_{k=1}^{n-1} \beta_k} & \text{if } \sum_{k=1}^{n-1} \beta_k < 1, \\ 1 & \text{if } \sum_{k=1}^{n-1} \beta_k = 1. \end{cases}$$

Then it is easily seen that the time until absorption in both processes is equally distributed. Vice versa, $\beta_n = (1 - \alpha_1) \cdots (1 - \alpha_{n-1}) \alpha_n$.

We can define a distribution to be DFR or IFR if the failure rate (defined as $f(t)/(1 - F(t))$, with f the density of F) is decreasing or increasing. However, then we implicitly assume that the failure rate, and thus the density, exists. To avoid this, we prefer to use the definition of Barlow & Prochan [5], which is only in terms of $\bar{F}(t) = 1 - F(t)$. Then it follows for example that F with $F(t) = \mathbf{I}\{t \geq x\}$, the deterministic distribution, is also IFR, although its failure rate does not exist.

B.1. Definition. (DFR and IFR) A non-negative distribution function is: DFR if $\bar{F}(t + s)/\bar{F}(t)$ is increasing in $t \geq 0$ with $\bar{F}(t) > 0$, for each $s \geq 0$; IFR if $\bar{F}(t + s)/\bar{F}(t)$ is decreasing in $-\infty < t < \infty$ with $\bar{F}(t) > 0$, for each $s \geq 0$.

Now we can formulate the main result of this appendix:

B.2. Theorem. *If F is DFR (IFR) then α_n is decreasing (increasing) in n , for all m .*

Proof. First we consider the DFR case. Take $s = 1/m$ and $t = 1/m, 2/m, \dots$. Then, according to the definition of DFR, $\bar{F}(n/m)/\bar{F}((n-1)/m)$ is increasing. Therefore $(F(n/m) - F((n-1)/m))/\bar{F}((n-1)/m)$ is decreasing in n . By the definition of β_n , $\bar{F}((n-1)/m) = 1 - F((n-1)/m) = 1 - \sum_{k=1}^{n-1} \beta_k$. Because $\beta_n = F(n/m) - F((n-1)/m)$, α_n is decreasing in n if $n \geq 2$. As $\beta_1 = F(1/m) \geq F(1/m) - F(0)$, we also have $\alpha_1 \geq \alpha_2$.

Concerning IFR distributions, the analysis goes completely analogous, except for β_1 . We show that $F(0) = 0$ or 1 . Assume $F(0) = a$, $0 < a < 1$. By the right-continuity of distribution functions we can find t_1 and ε such that $\bar{F}(t_1 + \varepsilon)/\bar{F}(t_1) > 1 - a$, and $\bar{F}(t_1) > 0$. Because $\bar{F}(0)/\bar{F}(-\varepsilon) = 1 - a$, we have a contradiction with the IFR assumption. Thus $F(0) = 0$ or 1 , in the former case giving $\beta_1 = F(1/m) - F(0)$, and in the latter case $\alpha_n = 1$ for all n . \square

A disadvantage of this method is that we need an infinite number of states. Therefore we change the process of figure B.2, making the state space finite, as shown in figure B.3.

This corresponds with changing the approximation into

$$\hat{F}_m(x) = \sum_{k=1}^{m^2} \beta_k E_m^k(x) + (1 - \sum_{k=1}^{m^2} \beta_k) \sum_{k=1}^{\infty} (1 - \beta_{m^2})^{k-1} \beta_{m^2} E_m^{m^2+k}(x).$$

It is easily checked that the approximation lemma A.2 also holds for \hat{F}_m .

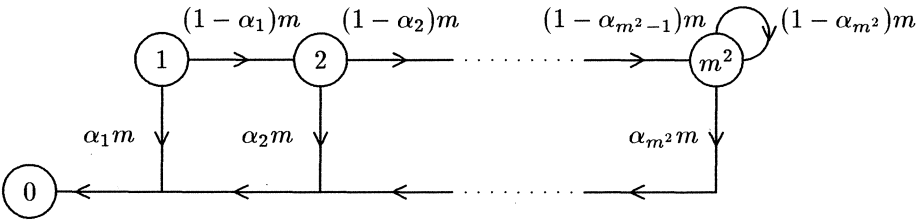


Figure B.3.

Appendix C

Majorization

In the customer assignment models the class of all allowable cost functions can often be characterized with the help of majorization. For two types of orderings, originating from the symmetric case (see e.g. section 1.2) and the case $B = 1$ (see e.g. section 1.3), we have a complete characterization. For the more general model of section 3.2 we give a conjecture for the correct ordering.

In the first ordering, all vectors considered are componentwise smaller than the buffer vector B . Consider the ordering \prec , with $i \prec i^*$ if there are i^1, \dots, i^n , $i^0 = i$ and $i^n = i^*$, such that

$$i^k = i^{k-1} - e_{j_1} + e_{j_2} \text{ if } 0 < i_{j_1}^{k-1} < i_{j_2}^{k-1} + 1 \quad (\text{C.1})$$

or

$$i^k = i^{k-1} + e_j \quad (\text{C.2})$$

or

$$i^k \text{ is a permutation of } i^{k-1}. \quad (\text{C.3})$$

Now consider the *weak submajorization ordering* \prec_w (see Marshall & Olkin [45]). We write $i \prec_w i^*$ if $\sum_{j=1}^k i_{[j]} \leq \sum_{j=1}^k i_{[j]}^*$ for all k , with $i_{[1]} \geq \dots \geq i_{[m]}$ the decreasing rearrangement of i . Thus, the sum of the k th largest components of i is smaller than that of i^* .

C.1. Theorem. *The orderings \prec and \prec_w are equivalent.*

Proof. $i \prec i^* \Rightarrow i \prec_w i^*$. Take i^0, \dots, i^n as in (C.1), (C.2) or (C.3). It is easy to see that $i^{k-1} \prec_w i^k$ for all k . Because \prec_w is a preordering transitivity holds and $i \prec_w i^*$.

$i \prec_w i^* \Rightarrow i \prec i^*$. We construct i^0, \dots, i^n such that $i = i^0 \prec \dots \prec i^n = i^*$. Assume that the k largest components of i^k are equal to, and in the same place as, the k largest components of i^* , and $i = i^0 \prec \dots \prec i^k \prec_w i^*$. We construct i^{k+1} with the property that either i^{k+1} has the $k+1$ largest components equal to i^* and $i^k \prec i^{k+1} \prec_w i^*$, or $i^{k+1} = i^*$. Repeating this gives the result. For simplicity of notation assume that $k = 0$.

Take the largest component of i^0 , say queue j_1 , and interchange it with the component of i^0 with the index of the longest queue of i^* , say queue j_2 . Call the resulting vector i' . Then, as $i_{[1]}^0 \leq i_{[1]}^*$, $i_{[1]}^0$ fits in the buffer of queue

j_2 . Because $i_{j_1}^0 \geq i_{j_2}^0$, $i_{j_2}^0$ fits in the buffer of queue j_1 , thus $i' \leq B$. We have by symmetry $i^0 \prec i'$ and trivially $i' \prec_w i^*$.

If $i'_{[2]} = 0$, the result follows by (C.2), because all components except $i'_{[1]}$ are 0. Thus, assume $i'_{[2]} > 0$.

Now we transfer a customer from $i'_{[2]}$ to $i'_{[1]}$. Call the resulting vector i'' . By (C.1) we have $i' \prec i''$. To show $i'' \prec_w i^*$ we distinguish the following two cases.

In case $i'_{[2]} > i'_{[3]}$, then $i'_{[1]} + i'_{[2]} = i''_{[1]} + i''_{[2]}$ and $i'' \prec_w i^*$ follows immediately.

In case $i'_{[2]} = \dots = i'_{[k]} > i'_{[k+1]}$ then $i''_{[1]} + \dots + i''_{[l]} = 1 + i'_{[1]} + \dots + i'_{[l]}$, $l < k$. However, it is straightforward to see that $i'_{[1]} + \dots + i'_{[l]} < i^*_{[1]} + \dots + i^*_{[l]}$. Thus $i'' \prec_w i^*$.

Repeat this until either $i''_{[1]} = i^*_{[1]}$ (and call the resulting vector i^1 , repeat the argument) or $i''_{[1]} = 0$ (which case is already handled). \square

The equivalence of theorem C.1 gives that the class of functions satisfying for example (1.2.2), (1.2.3) and (1.2.4) is precisely the class of functions preserving weak submajorization. These functions are called the *weak Schur convex functions*, cf. Marshall & Olkin [45]. According to Marshall & Olkin [45], a similar result has been shown by Muirhead [48]. He shows (presumably for $B = \infty$) that the ordering obtained by transfers of the form (C.1) and (C.2) is equivalent to the majorization ordering. This ordering is like the weak majorization ordering, but with the additional constraint $\sum_{j=1}^n i_{[j]} = \sum_{j=1}^n i^*_{[j]}$.

In much the same way we can give the generalization of the results of section 1.3. There we took $B = 1$. It appears that the result can easily be extended to arbitrary buffers. Then the ordering agrees with the *partial sum ordering* of Chang et al. [12], used there in the context of a server assignment model. Again, all vectors considered are smaller than B . Define the partial ordering \prec as follows: $i \prec i^*$ if there are i^0, \dots, i^n with $i^0 = i$ and $i^n = i^*$ such that

$$i^k = i^{k-1} - e_{j_1} + e_{j_2} \text{ if } i_{j_1}^{k-1} > 0 \text{ and } j_1 < j_2 \quad (\text{C.4})$$

or

$$i^k = i^{k-1} + e_j. \quad (\text{C.5})$$

It is easily seen that the class of cost functions satisfying (1.3.2) and (1.3.3) is precisely the class of functions preserving the ordering \prec .

We show that the ordering \prec is equivalent to an ordering \prec' , defined by: $i \prec' i^*$ if $\sum_{j=k}^m i_j \leq \sum_{j=k}^m i^*_j$ for $k = 1, \dots, m$.

C.2. Theorem. *The orderings \prec and \prec' are equivalent.*

Proof. $i \prec i^* \Rightarrow i \prec' i^*$. Take $i = i^0 \prec \dots \prec i^n = i^*$ as in the definition of \prec . It is easy to see that $i^{k-1} \prec' i^k$ for all k . Due to the transitivity of \prec' we have $i \prec' i^*$.

$i \prec' i^* \Rightarrow i \prec i^*$. We construct i^0, \dots, i^n such that $i = i^0 \prec \dots \prec i^n = i^*$. First add $\sum_{j=1}^m i_j^* - \sum_{j=1}^m i_j$ customers to state i^0 , adding them to the queues with smallest indices, without passing the buffer sizes. Call this state i^1 . Then $i \prec i^1 \prec' i^*$. Clearly, if $i^1 = i^*$, we are ready. If not, let j_2 be the highest numbered queue with $i_{j_2}^1 < i_{j_2}^*$. Now, construct $i' = i^1 - e_{j_1} + e_{j_2}$, with j_1 the highest numbered queue with $j_1 < j_2$ and $i_{j_1}' > 0$. Since $\sum_{j=k}^m i_j^1 < \sum_{j=k}^m i_j^*$ for $k = j_1 + 1, \dots, j_2$ we have $i' \prec' i^*$. Repeat this construction until we have $i_{j_1}' = i_{j_1}^*$. Choose a new j_1 and repeat.

Finally, $i^1 \prec \dots \prec i^n$ and transitivity gives $i \prec i^*$. \square

Let us look at the \prec' -preserving functions. Allowed cost functions are $c_i = \sum_{j=k}^m i_j$, for all k , the total number of customers in the $m - k$ queues with slowest servers. Hence the FQP minimizes the total number of customers in the $m - k$ queues with slowest servers stochastically for all $k = 1, \dots, m$. It is clear that there are other interesting \prec' -preserving functions, e.g. the weighted total number of customers with increasing weights.

Finally, consider the model of section 3.2. To study the allowable cost functions, define the ordering \prec as follows: $i \prec i^*$ if there are i^1, \dots, i^n , $i^0 = i$ and $i^n = i^*$, such that

$$i^k = i^{k-1} - e_{j_1} + e_{j_2} \text{ if } 0 < i_{j_1}^{k-1} < i_{j_2}^{k-1} + 1 \text{ and } j_1 < j_2 \quad (\text{C.6})$$

or

$$i^k = i^{k-1} + e_j \quad (\text{C.7})$$

or

$$i^k \text{ a permutation of } i^{k-1} \text{ with } j_1 \text{ and } j_2 \text{ exchanged, } i_{j_1}^{k-1} > i_{j_2}^{k-1} \text{ and } j_1 < j_2. \quad (\text{C.8})$$

Now define an ordering as follows: $i \prec' i^*$ if $\sum_{j=k}^n (i_j - l)^+ \leq \sum_{j=k}^n (i_j^* - l)^+$ for all k and l .

C.3. Conjecture. *The orderings \prec and \prec' are equivalent.*

Appendix D

Computational issues

In this section we compare different computational methods, mostly from a practical point of view. We consider value iteration (also called successive approximation), both for discounted and average costs and uniformization (with a large parameter).

We introduce some notation. Let λ_{xay} be the transition intensity from x to y using action a and c_x the costs in state x . Assume that the state space is finite (which simplifies the analysis but is also necessary for the computations) and that $\sum_y \lambda_{xay} = \alpha$ for all x and a . Consider the following iteration scheme:

$$v_x^{n+1} = \min_a \left\{ c_x + \hat{\beta} \sum_y \frac{\lambda_{xay}}{\alpha} v_y^n \right\}$$

It is well known that v_x^n converges to the minimal discounted costs if $\hat{\beta} \in [0, 1)$ and that, under an aperiodicity assumption, $v_x^{n+1} - v_x^n$ converges to the minimal average costs if $\hat{\beta} = 1$.

We start by motivating the choices for the discount factor made in table 2.3.1. Assume that the costs are continuously incurred over time, meaning that the costs at t are multiplied with β^t . Then we take in the discrete model $\hat{\beta} = \alpha / (\log(\beta^{-1}) + \alpha)$ as discount factor. In table D.1 the values of $\hat{\beta}$ are given for the choices of β taken in table 2.3.1, for the typical value $\alpha = 5$. It is surprising to compare the values of β and $\hat{\beta}$.

β	$\hat{\beta}$
0.01	0.52
0.1	0.68
0.25	0.78
0.5	0.88
0.75	0.95

Table D.1. Discount factors

Our computations were done on workstations. For the model of section 2.3 we give in table D.2 the computer time in seconds and the number of iterations, for $\lambda = 1$, $B = 20$ and an accuracy of 10^{-10} , needed to calculate the discounted and average costs ($\beta = 1$) under the SQP with value iteration. Each iteration takes about 4 seconds. Note that the number of states is approximately $1.6 \cdot 10^5$. When λ (or B) is increased the number of iterations sharply increases. For example, if $\lambda = 1.9$ and $B = 30$, 43288 iterations were needed.

β	time	iterations
0.01	172	42
0.1	277	69
0.25	419	105
0.5	758	192
0.75	1708	436
1	5761	1224

Table D.2. Value iteration

Our experience is that it is easy to overlook some of the possible transitions. In the discounted case the value function still converges, but to the wrong solution. However, in the average cost case, in a model which is irreducible under each policy, the costs converge to 0. Therefore it is preferable also to program the average cost case, even if we are only interested in discounted costs.

We continue with uniformization. Consider the following iteration scheme:

$$v_x^{h,n+1} = \min_a \left\{ hc_x + \sum_y h \lambda_{xy} v_y^n + (1 - h\alpha) v_x^{h,n} \right\}$$

As we showed in section 5.3, $v^{h, \lfloor T/h \rfloor}$ converges to the costs from 0 to T as $h \rightarrow 0$. Little is known about the convergence of this method. Some bounds on the speed of convergence can be found in Van Dijk [72] and [73]. Our computational experience is summarized in table D.3. There for various values of T and h the total costs from 0 to T are given, again for the model of section 2.3, starting from the states with in each queue 10 customers. Other starting states give similar results. The number of iterations is h^{-1} . Each iteration takes a little less time than for discounted and average costs, because we do not have to check whether we are finished iterating. (Because we did computations on several computers, some of which were faster than others, we do not supply computer times.) Note that because $\alpha = 5$, h needs to be smaller than 0.2. For $T = 1$, it seems that an accuracy of 10^{-10} is obtained for $h = 10^{-10}$, meaning 10^{10} iterations. If each iteration takes 3 seconds, this takes approximately 950 years. This explains why we computed, for several models, discounted and average costs, but not costs from 0 to T .

	$T = 1$	10	100
$h = 0.1$	39.550000	350.510743	1033.855844
0.01	39.505000	350.062022	1033.855578
0.001	39.500500	350.017155	
0.0001	39.500050		

Table D.3. Uniformization with $\alpha \rightarrow \infty$

References

- [1] I.J.B.F. Adan, J. Wessels & W.H.M. Zijm (1989). Analysis of the asymmetric shortest queue problem. *Queueing Systems, Theory and Applications* **8**: 1–58.
- [2] E. Altman & G.M. Koole (1995). On submodular value functions of dynamic programming. Technical report, INRIA Sophia Antipolis.
- [3] S. Asmussen & G.M. Koole (1993). Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability* **30**: 365–372.
- [4] J.S. Baras, D.-J. Ma & A.M. Makowski (1985). K competing queues with geometric service requirements and linear costs: the μc -rule is always optimal. *Systems & Control Letters* **6**: 173–180.
- [5] R.E. Barlow & F. Proschan (1975). *Statistical Theory of Reliability and Life Testing*. Holt, Rinehart and Winston, New York.
- [6] P. Billingsley (1968). *Convergence of Probability Measures*. Wiley, New York.
- [7] J. Bruno, P. Downey & G.N. Frederickson (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *Journal of the ACM* **28**: 100–113.
- [8] C. Buyukkoc, P. Varaiya & J. Walrand (1985). The $c\mu$ rule revisited. *Advances in Applied Probability* **17**: 237–238.
- [9] R. Chakka & I. Mitrani (1994). Heterogeneous multiprocessor systems with breakdowns: Performance and optimal repair strategies. *Theoretical Computer Science* **125**: 91–109.
- [10] C.S. Chang (1992). A new ordering for stochastic majorization: Theory and applications. *Advances in Applied Probability* **24**: 604–634.
- [11] C.S. Chang, X. Chao & M. Pinedo (1990). A note on queues with Bernoulli routing. Proceedings of the 29th Conference on Decision and Control, Hawaii, 897–902.
- [12] C.S. Chang, X. Chao, M. Pinedo & R.R. Weber (1992). On the optimality of LEPT and $c\mu$ -rules for machines in parallel. *Journal of Applied Probability* **29**: 667–681.
- [13] C.S. Chang, A. Hordijk, R. Richter & G. Weiss (1994). The stochastic optimality of SEPT in parallel machine scheduling. *Probability in the Engineering and Informational Sciences* **8**: 179–188.
- [14] R.B. Cooper & S. Palakurthi (1989). Heterogeneous-server loss systems with ordered entry: an anomaly. *Operations Research Letters* **8**: 347–349.

- [15] D.J. Daley (1987). Certain optimality properties of the first-come first-served discipline for $G|G|s$ queues. *Stochastic Processes and their Applications* **25**: 301–308.
- [16] R. Dekker & A. Hordijk (1991). Denumerable semi-Markov decision chains with small interest rates. *Annals of Operations Research* **28**: 185–212.
- [17] C. Derman, G.J. Lieberman & S.M. Ross (1980). On the optimal assignment of servers and a repairman. *Journal of Applied Probability* **17**: 577–581.
- [18] S.G. Foss (1982). Extremal problems in queueing theory. Ph.D. thesis, Novosibirsk State University (in Russian).
- [19] B. Hajek (1984). Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control* **29**: 491–499.
- [20] U. Herrmann (1965). Ein Approximationssatz für Verteilungen stationärer zufälliger Punktfolgen. *Mathematische Nachrichten* **30**: 377–381.
- [21] A. Hordijk & G.M. Koole (1990). On the optimality of the generalized shortest queue policy. *Probability in the Engineering and Informational Sciences* **4**: 477–487.
- [22] A. Hordijk & G.M. Koole (1992). On the shortest queue policy for the tandem parallel queue. *Probability in the Engineering and Informational Sciences* **6**: 63–79.
- [23] A. Hordijk & G.M. Koole (1992). The μc -rule is not optimal in the second node of the tandem queue: a counterexample. *Advances in Applied Probability* **24**: 234–237.
- [24] A. Hordijk & G.M. Koole (1992). On the assignment of customers to parallel queues. *Probability in the Engineering and Informational Sciences* **6**: 495–511.
- [25] A. Hordijk & G.M. Koole (1993). On the optimality of LEPT and μc rules for parallel processors and dependent arrival processes. *Advances in Applied Probability* **25**: 979–996.
- [26] A. Hordijk & G.M. Koole (1995). On suboptimal policies in multi-class tandem models. To appear in *Probability in the Engineering and Informational Sciences*.
- [27] A. Hordijk & A. Ridder (1987). Stochastic inequalities for an overflow model. *Journal of Applied Probability* **24**: 696–708.
- [28] A. Hordijk & R. Schassberger (1982). Weak convergence for generalized semi-Markov processes. *Stochastic Processes and their Applications* **12**: 271–291.
- [29] A. Hordijk & F. van der Duyn Schouten (1985). Markov decision drift processes; conditions for optimality obtained by discretization. *Mathematics of Operations Research* **10**: 160–173.
- [30] D.J. Houck (1987). Comparison of policies for routing customers to parallel queueing systems. *Operations Research* **35**: 306–310.
- [31] P.K. Johri (1989). Optimality of the shortest line discipline with state-dependent service times. *European Journal of Operations Research* **41**: 157–161.

- [32] M.N. Katehakis & A. Levine (1985). A dynamic routing problem—Numerical procedures for light traffic conditions. *Applied Mathematics and Computation* **17**: 267–276.
- [33] M.N. Katehakis & C. Melolidakis (1989). On the optimal maintenance of systems and control of arrivals in queues. Technical report, Graduate School of Management, Rutgers University.
- [34] J.F.C. Kingman (1961). Two similar queues in parallel. *Annals of Mathematical Statistics* **32**: 1314–1323.
- [35] G.M. Koole (1991). Assigning multiple customer classes to parallel servers. Technical report TW-91-07, Leiden University.
- [36] G.M. Koole (1992). On the optimality of FCFS for networks of multi-server queues. Technical report BS-R923, CWI, Amsterdam.
- [37] G.M. Koole (1993). Optimal server assignment in the case of service times with monotone failure rates. *Systems and Control Letters* **20**: 233–238.
- [38] G.M. Koole (1994). On the pathwise optimal Bernoulli routing policy for homogeneous parallel servers. Technical report 2443, INRIA Sophia Antipolis.
- [39] G.M. Koole (1995). Optimal Repairman Assignment in Two Symmetric Maintenance Models. *European Journal of Operations Research* **82**: 295–301.
- [40] G.M. Koole and M. Vrijenhoek (1995). Scheduling a repairman in a finite source system. Technical report TW-95-03, Leiden University.
- [41] V.G. Kulkarni & Y. Serin (1995). Optimal implementable policies: discounted cost case. In: W.J. Stewart (ed.), *Computations with Markov Chains*. Kluwer, Boston.
- [42] Z. Liu and D. Towsley (1994). Optimality of the round robin routing policy. *Journal of Applied Probability* **31**: 466–475.
- [43] M. Loève (1977). *Probability Theory I*, 4th ed. Springer-Verlag, New York.
- [44] A. Loeve & M. Pols (1990). Optimale toewijzing van klanten in wachtrijmodellen met twee bedieningscentra. Master thesis, Leiden University.
- [45] A.W. Marshall & I. Olkin (1979). *Inequalities: Theory of Majorization and its Applications*. Academic Press, New York.
- [46] R. Menich & R.F. Serfozo (1991). Monotonicity and optimality of symmetric parallel processing systems. *Queueing Systems, Theory and Applications* **9**: 403–418.
- [47] A. van Moorsel & E. de Vries (1989). Optimale toewijzingsstrategieën bij eenvoudige netwerken van wachtrijen. Master thesis, Leiden University.
- [48] R.F. Muirhead (1903). Some methods applicable to identities and inequalities of symmetric algebraic functions of n letters. *Proceedings Edinburgh Mathematical Society* **21**: 144–157.
- [49] P. Nain (1989). Interchange arguments for classical scheduling problems in queues. *Systems and Control Letters* **12**: 177–184.
- [50] P. Nain, P. Tsoucas & J. Walrand (1989). Interchange arguments in stochastic scheduling. *Journal of Applied Probability* **27**: 815–826.

- [51] M.F. Neuts (1979). A versatile Markovian point process. *Journal of Applied Probability* **16**: 764–779.
- [52] M.F. Neuts (1981). *Matrix-Geometric Solutions in Stochastic Models; an Algorithmic Approach*. John Hopkins, Baltimore.
- [53] M.F. Neuts (1989). *Structured Stochastic Matrices of $M|G|1$ Type and their Applications*. Marcel Dekker, New York.
- [54] R. Nobel & H.C. Tijms (1990). Optimal routing of customers to parallel service groups. Memorandum 90-47, Vrije Universiteit Amsterdam.
- [55] M. Pinedo & L. Schrage (1982). Stochastic shop scheduling: A survey. In: M.A.H. Dempster, J.K. Lenstra & A.H.G. Rinnooy Kan (eds.), *Deterministic and Stochastic Scheduling*. Reidel, Dordrecht.
- [56] R. Righter (1995). Optimal policies for scheduling repairs and allocating heterogeneous servers. To appear in *Journal of Applied Probability*.
- [57] R. Righter & J.G. Shanthikumar (1989). Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures. *Probability in the Engineering and Informational Sciences* **3**: 323–333.
- [58] R. Righter & J.G. Shanthikumar (1991). Extension of the bivariate characterization for stochastic orders. *Advances in Applied Probability* **24**: 506–508.
- [59] R. Righter & J.G. Shanthikumar (1991). Extremal properties of the FIFO discipline in queueing networks. *Journal of Applied Probability* **29**: 967–978.
- [60] S.M. Ross (1983). *Introduction to Stochastic Dynamic Programming*. Academic Press, New York.
- [61] M. Rudemo (1973). Point processes generated by transitions of Markov chains. *Advances in Applied Probability* **5**: 262–286.
- [62] R. Schassberger (1973). *Warteschlangen*. Springer-Verlag, Wien.
- [63] R.F. Serfozo (1979). An equivalence between continuous and discrete time Markov decision processes. *Operations Research* **27**: 616–620.
- [64] K. Seth (1977). Optimal service policies, just after idle periods, in two-server heterogeneous queueing systems. *Operations Research* **25**: 356–360.
- [65] S. Shenker & A. Weinrib (1989). The optimal control of heterogeneous queueing systems: a paradigm for load-sharing and routing. *IEEE Transactions on Computers* **38**: 1724–1735.
- [66] M.J. Sobel (1990). Throughput maximization in a loss queueing system with heterogeneous servers. *Journal of Applied Probability* **27**: 693–700.
- [67] M.J. Sobel & C. Srivastava (1991). Full-service policy optimality with heterogeneous servers. Working paper.
- [68] P.D. Sparagis & D. Towsley (1993). Optimal routing in systems with ILR service time distributions. CMPSCI Technical Report 93-13, University of Massachusetts at Amherst.
- [69] F.M. Spieksma (1990). Geometrically ergodic Markov chains and the optimal control of queues. Ph.D. thesis, Leiden University.

- [70] F.M. Spieksma (1991). The existence of sensitive optimal policies in two multi-dimensional queueing models. *Annals of Operations Research* **28**: 273–296.
- [71] D. Towsley, P.D. Sparaggis & C.G. Cassandras (1992). Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control* **37**: 1446–1451.
- [72] N.M. van Dijk (1983). Controlled Markov processes; time-discretization — networks of queues. Ph.D. thesis, Leiden University.
- [73] N.M. van Dijk (1988). On the finite horizon Bellman equation for controlled Markov jump models with unbounded characteristics: existence and approximations. *Stochastic Processes and their Applications* **28**: 141–157.
- [74] J. Walrand (1988). *An Introduction to Queueing Networks*. Prentice-Hall, Englewood Cliffs.
- [75] R.R. Weber (1978). On the optimal assignment of customers to parallel queues. *Journal of Applied Probability* **15**: 406–413.
- [76] R.R. Weber (1982). Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime. *Journal of Applied Probability* **19**: 167–182.
- [77] R.R. Weber (1992). The interchangeability of tandem queues with heterogeneous customers and dependent service times. *Advances in Applied Probability* **24**: 727–737.
- [78] J. Weishaupt (1994). Optimal myopic policies and index policies for stochastic scheduling problems. *Zeitschrift für Operations Research* **40**: 75–89.
- [79] G. Weiss (1982). Multiserver stochastic scheduling. In: M.A.H. Dempster, J.K. Lenstra & A.H.G. Rinnooy Kan (eds.), *Deterministic and Stochastic Scheduling*. Reidel, Dordrecht.
- [80] G. Weiss & M. Pinedo (1980). Scheduling tasks with exponential service times on nonidentical processors to minimize various cost functions. *Journal of Applied Probability* **17**: 187–202.
- [81] W. Whitt (1986). Deciding which queue to join: some counterexamples. *Operations Research* **34**: 55–62.
- [82] W. Winston (1977). Optimality of the shortest line discipline. *Journal of Applied Probability* **14**: 181–189.
- [83] R.W. Wolff (1977). An upper bound for multi-channel queues. *Journal of Applied Probability* **14**: 884–888.
- [84] R.W. Wolff (1987). Upper bounds on work in system for multichannel queues. *Journal of Applied Probability* **24**: 547–551.

Index

Bellman equation	108
Cyclic Assignment Policy (CAP)	20
Dependent Markov Decision Arrival Process (DMDAP)	57
DFR	119
discretization	107
Equal Splitting Policy (ESP)	20
Fastest Queue Policy (FQP)	9
FCFS	19
failure rate	119
IFR	119
LAST	27
LEPT	29
list policy	73
majorization	121
Markov Arrival Process (MAP)	5, 113
Markov Decision Arrival Process (MDAP)	33
Markov Modulated Poisson Process (MMPP)	6
MAST	27
partial information	38
partial sum ordering	122
pathwise optimality	13
Phase-type distribution	5, 113
Schur convexity	122
Shorter Faster Queue Policy (SFQP)	59
Shortest Queue Policy (SQP)	7
Shortest Workload Policy (SWP)	17
Smallest Index Policy (SIP)	24
static policy	38
stochastic optimality	8
tandem μc -rule	49
uniformization	6, 105, 107
μc -rule	23

CWI TRACTS

- 1 D.H.J. Epema. *Surfaces with canonical hyperplane sections*. 1984.
- 2 J.J. Dijkstra. *Fake topological Hilbert spaces and characterizations of dimension in terms of negligibility*. 1984.
- 3 A.J. van der Schaft. *System theoretic descriptions of physical systems*. 1984.
- 4 J. Koene. *Minimal cost flow in processing networks, a primal approach*. 1984.
- 5 B. Hoogenboom. *Intertwining functions on compact Lie groups*. 1984.
- 6 A.P.W. Böhm. *Dataflow computation*. 1984.
- 7 A. Blokhuis. *Few-distance sets*. 1984.
- 8 M.H. van Hoorn. *Algorithms and approximations for queueing systems*. 1984.
- 9 C.P.J. Koymans. *Models of the lambda calculus*. 1984.
- 10 C.G. van der Laan, N.M. Temme. *Calculation of special functions: the gamma function, the exponential integrals and error-like functions*. 1984.
- 11 N.M. van Dijk. *Controlled Markov processes; time-discretization*. 1984.
- 12 W.H. Hundsdorfer. *The numerical solution of nonlinear stiff initial value problems: an analysis of one step methods*. 1985.
- 13 D. Grune. *On the design of ALEPH*. 1985.
- 14 J.G.F. Thiemann. *Analytic spaces and dynamic programming: a measure theoretic approach*. 1985.
- 15 F.J. van der Linden. *Euclidean rings with two infinite primes*. 1985.
- 16 R.J.P. Groothuizen. *Mixed elliptic-hyperbolic partial differential operators: a case-study in Fourier integral operators*. 1985.
- 17 H.M.M. ten Eikelder. *Symmetries for dynamical and Hamiltonian systems*. 1985.
- 18 A.D.M. Kester. *Some large deviation results in statistics*. 1985.
- 19 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 1: Philosophy, framework, computer science*. 1986.
- 20 B.F. Schriever. *Order dependence*. 1986.
- 21 D.P. van der Vecht. *Inequalities for stopped Brownian motion*. 1986.
- 22 J.C.S.P. van der Woude. *Topological dynamix*. 1986.
- 23 A.F. Monna. *Methods, concepts and ideas in mathematics: aspects of an evolution*. 1986.
- 24 J.C.M. Baeten. *Filters and ultrafilters over definable subsets of admissible ordinals*. 1986.
- 25 A.W.J. Kolen. *Tree network and planar rectilinear location theory*. 1986.
- 26 A.H. Veen. *The misconstrued semicolon: Reconciling imperative languages and dataflow machines*. 1986.
- 27 A.J.M. van Engelen. *Homogeneous zero-dimensional absolute Borel sets*. 1986.
- 28 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 2: Applications to natural language*. 1986.
- 29 H.L. Trentelman. *Almost invariant subspaces and high gain feedback*. 1986.
- 30 A.G. de Kok. *Production-inventory control models: approximations and algorithms*. 1987.
- 31 E.E.M. van Berkum. *Optimal paired comparison designs for factorial experiments*. 1987.
- 32 J.H.J. Einmahl. *Multivariate empirical processes*. 1987.
- 33 O.J. Vrieze. *Stochastic games with finite state and action spaces*. 1987.
- 34 P.H.M. Kersten. *Infinitesimal symmetries: a computational approach*. 1987.
- 35 M.L. Eaton. *Lectures on topics in probability inequalities*. 1987.
- 36 A.H.P. van der Burgh, R.M.M. Mattheij (eds.). *Proceedings of the first international conference on industrial and applied mathematics (ICIAM 87)*. 1987.
- 37 L. Stougie. *Design and analysis of algorithms for stochastic integer programming*. 1987.
- 38 J.B.G. Frenk. *On Banach algebras, renewal measures and regenerative processes*. 1987.
- 39 H.J.M. Peters, O.J. Vrieze (eds.). *Surveys in game theory and related topics*. 1987.
- 40 J.L. Geluk, L. de Haan. *Regular variation, extensions and Tauberian theorems*. 1987.
- 41 Sape J. Mullender (ed.). *The Amoeba distributed operating system: Selected papers 1984-1987*. 1987.
- 42 P.R.J. Asveld, A. Nijholt (eds.). *Essays on concepts, formalisms, and tools*. 1987.
- 43 H.L. Bodlaender. *Distributed computing: structure and complexity*. 1987.
- 44 A.W. van der Vaart. *Statistical estimation in large parameter spaces*. 1988.
- 45 S.A. van de Geer. *Regression analysis and empirical processes*. 1988.
- 46 S.P. Spekrijse. *Multigrid solution of the steady Euler equations*. 1988.
- 47 J.B. Dijkstra. *Analysis of means in some non-standard situations*. 1988.
- 48 F.C. Drost. *Asymptotics for generalized chi-square goodness-of-fit tests*. 1988.
- 49 F.W. Wubs. *Numerical solution of the shallow-water equations*. 1988.
- 50 F. de Kerf. *Asymptotic analysis of a class of perturbed Korteweg-de Vries initial value problems*. 1988.
- 51 P.J.M. van Laarhoven. *Theoretical and computational aspects of simulated annealing*. 1988.
- 52 P.M. van Loon. *Continuous decoupling transformations for linear boundary value problems*. 1988.
- 53 K.C.P. Machielsens. *Numerical solution of optimal control problems with state constraints by sequential quadratic programming in function space*. 1988.
- 54 L.C.R.J. Willenborg. *Computational aspects of survey data processing*. 1988.
- 55 G.J. van der Steen. *A program generator for recognition, parsing and transduction with syntactic patterns*. 1988.
- 56 J.C. Ebergen. *Translating programs into delay-insensitive circuits*. 1989.
- 57 S.M. Verduyn Lunel. *Exponential type calculus for linear delay equations*. 1989.
- 58 M.C.M. de Gunst. *A random model for plant cell population growth*. 1989.
- 59 D. van Dulst. *Characterizations of Banach spaces not containing ℓ^1* . 1989.
- 60 H.E. de Swart. *Vacillation and predictability properties of low-order atmospheric spectral models*. 1989.
- 61 P. de Jong. *Central limit theorems for generalized multilinear forms*. 1989.
- 62 V.J. de Jong. *A specification system for statistical software*. 1989.
- 63 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part I*. 1989.
- 64 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part II*. 1989.
- 65 B.M.M. de Weger. *Algorithms for diophantine equations*. 1989.
- 66 A. Jung. *Cartesian closed categories of domains*. 1989.
- 67 J.W. Polderman. *Adaptive control & identification: Conflict or conflux?*. 1989.
- 68 H.J. Woerdeman. *Matrix and operator extensions*. 1989.
- 69 B.G. Hansen. *Monotonicity properties of infinitely divisible distributions*. 1989.
- 70 J.K. Lenstra, H.C. Tijms, A. Volgenant (eds.). *Twenty-five years of operations research in the Netherlands: Papers dedicated to Gijb de Leve*. 1990.
- 71 P.J.C. Spreij. *Counting process systems. Identification and stochastic realization*. 1990.
- 72 J.F. Kaashoek. *Modeling one dimensional pattern formation by anti-diffusion*. 1990.
- 73 A.M.H. Gerard. *Graphs and polyhedra. Binary spaces and cutting planes*. 1990.
- 74 B. Koren. *Multigrid and defect correction for the steady Navier-Stokes equations. Application to aerodynamics*. 1991.
- 75 M.W.P. Savelsbergh. *Computer aided routing*. 1992.

- 76 O.E. Flippo. *Stability, duality and decomposition in general mathematical programming*. 1991.
- 77 A.J. van Es. *Aspects of nonparametric density estimation*. 1991.
- 78 G.A.P. Kindervater. *Exercises in parallel combinatorial computing*. 1992.
- 79 J.J. Lodder. *Towards a symmetrical theory of generalized functions*. 1991.
- 80 S.A. Smulders. *Control of freeway traffic flow*. 1993.
- 81 P.H.M. America, J.J.M.M. Rutten. *A parallel object-oriented language: design and semantic foundations*. 1992.
- 82 F. Thuijsman. *Optimality and equilibria in stochastic games*. 1992.
- 83 R.J. Kooman. *Convergence properties of recurrence sequences*. 1992.
- 84 A.M. Cohen (ed.). *Computational aspects of Lie group representations and related topics. Proceedings of the 1990 Computational Algebra Seminar at CWI, Amsterdam*. 1991.
- 85 V. de Valk. *One-dependent processes*. 1994.
- 86 J.A. Baars, J.A.M. de Groot. *On topological and linear equivalence of certain function spaces*. 1992.
- 87 A.F. Monna. *The way of mathematics and mathematicians*. 1992.
- 88 E.D. de Goede. *Numerical methods for the three-dimensional shallow water equations*. 1993.
- 89 M. Zwaan. *Moment problems in Hilbert space with applications to magnetic resonance imaging*. 1993.
- 90 C. Vuik. *The solution of a one-dimensional Stefan problem*. 1993.
- 91 E.R. Verheul. *Multimedians in metric and normed spaces*. 1993.
- 92 J.L.M. Maubach. *Iterative methods for non-linear partial differential equations*. 1994.
- 93 A.W. Ambergen. *Statistical uncertainties in posterior probabilities*. 1993.
- 94 P.A. Zegeling. *Moving-grid methods for time-dependent partial differential equations*. 1993.
- 95 M.J.C. van Pul. *Statistical analysis of software reliability models*. 1993.
- 96 J.K. Scholma. *A Lie algebraic study of some integrable systems associated with root systems*. 1993.
- 97 J.L. van den Berg. *Sojourn times in feedback and processor sharing queues*. 1993.
- 98 A.J. Koning. *Stochastic integrals and goodness-of-fit tests*. 1993.
- 99 B.P. Sommeijer. *Parallelism in the numerical integration of initial value problems*. 1993.
- 100 J. Molenaar. *Multigrid methods for semiconductor device simulation*. 1993.
- 101 H.J.C. Huijberts. *Dynamic feedback in nonlinear synthesis problems*. 1994.
- 102 J.A.M. van der Weide. *Stochastic processes and point processes of excursions*. 1994.
- 103 P.W. Hemker, P. Wesseling (eds.). *Contributions to multigrid*. 1994.
- 104 I.J.B.F. Adan. *A compensation approach for queueing problems*. 1994.
- 105 O.J. Boxma, G.M. Koole (eds.). *Performance evaluation of parallel and distributed systems - solution methods. Part 1*. 1994.
- 106 O.J. Boxma, G.M. Koole (eds.). *Performance evaluation of parallel and distributed systems - solution methods. Part 2*. 1994.
- 107 R.A. Trompert. *Local uniform grid refinement for time-dependent partial differential equations*. 1995.
- 108 M.N.M. van Lieshout. *Stochastic geometry models in image analysis and spatial statistics*. 1995.
- 109 R.J. van Glabbeek. *Comparative concurrency semantics and refinement of actions*. 1995.
- 110 W. Vervaat (ed.). *Probability and lattices*. 1995.
- 111 I. Helsloot. *Covariant formal group theory and some applications*. 1995.
- 112 R.N. Bol. *Loop checking in logic programming*. 1995.
- 113 G.J.M. Koole. *Stochastic scheduling and dynamic programming*. 1995.

MATHEMATICAL CENTRE TRACTS

- 1 T. van der Walt. *Fixed and almost fixed points*. 1963.
- 2 A.R. Bloemena. *Sampling from a graph*. 1964.
- 3 G. de Leve. *Generalized Markovian decision processes, part 1: model and method*. 1964.
- 4 G. de Leve. *Generalized Markovian decision processes, part II: probabilistic background*. 1964.
- 5 G. de Leve, H.C. Tijms, P.J. Weeda. *Generalized Markovian decision processes, applications*. 1970.
- 6 M.A. Maurice. *Compact ordered spaces*. 1964.
- 7 W.R. van Zwet. *Convex transformations of random variables*. 1964.
- 8 J.A. Zonneveld. *Automatic numerical integration*. 1964.
- 9 P.C. Baayen. *Universal morphisms*. 1964.
- 10 E.M. de Jager. *Applications of distributions in mathematical physics*. 1964.
- 11 A.B. Paalman-de Miranda. *Topological semigroups*. 1964.
- 12 J.A.Th.M. van Berckel, H. Brandt Corstius, R.J. Mokken, A. van Wijngaarden. *Formal properties of newspaper Dutch*. 1965.
- 13 H.A. Lauwerier. *Asymptotic expansions*. 1966, out of print: replaced by MCT 54.
- 14 H.A. Lauwerier. *Calculus of variations in mathematical physics*. 1966.
- 15 R. Doornbos. *Slippage tests*. 1966.
- 16 J.W. de Bakker. *Formal definition of programming languages with an application to the definition of ALGOL 60*. 1967.
- 17 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 1*. 1968.
- 18 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 2*. 1968.
- 19 J. van der Slot. *Some properties related to compactness*. 1968.
- 20 P.J. van der Houwen. *Finite difference methods for solving partial differential equations*. 1968.
- 21 E. Wattel. *The compactness operator in set theory and topology*. 1968.
- 22 T.J. Dekker. *ALGOL 60 procedures in numerical algebra, part 1*. 1968.
- 23 T.J. Dekker, W. Hoffmann. *ALGOL 60 procedures in numerical algebra, part 2*. 1968.
- 24 J.W. de Bakker. *Recursive procedures*. 1971.
- 25 E.R. Paërl. *Representations of the Lorentz group and projective geometry*. 1969.
- 26 European Meeting 1968. *Selected statistical papers, part 1*. 1968.
- 27 European Meeting 1968. *Selected statistical papers, part II*. 1968.
- 28 J. Oosterhoff. *Combination of one-sided statistical tests*. 1969.
- 29 J. Verhoeff. *Error detecting decimal codes*. 1969.
- 30 H. Brandt Corstius. *Exercises in computational linguistics*. 1970.
- 31 W. Molenaar. *Approximations to the Poisson, binomial and hypergeometric distribution functions*. 1970.
- 32 L. de Haan. *On regular variation and its application to the weak convergence of sample extremes*. 1970.
- 33 F.W. Steutel. *Preservations of infinite divisibility under mixing and related topics*. 1970.
- 34 I. Juhász, A. Verbeek, N.S. Kroonenberg. *Cardinal functions in topology*. 1971.
- 35 M.H. van Emden. *An analysis of complexity*. 1971.
- 36 J. Grasman. *On the birth of boundary layers*. 1971.
- 37 J.W. de Bakker, G.A. Blaauw, A.J.W. Duijvestijn, E.W. Dijkstra, P.J. van der Houwen, G.A.M. Kamsteeg-Kemper, F.E.J. Kruseman Aretz, W.L. van der Poel, J.P. Schaap-Kruseman, M.V. Wilkes, G. Zoutendijk. *MC-25 Informatica Symposium*. 1971.
- 38 W.A. Verloren van Themaat. *Automatic analysis of Dutch compound words*. 1972.
- 39 H. Bavinck. *Jacobi series and approximation*. 1972.
- 40 H.C. Tijms. *Analysis of (s,S) inventory models*. 1972.
- 41 A. Verbeek. *Superextensions of topological spaces*. 1972.
- 42 W. Vervaat. *Success epochs in Bernoulli trials (with applications in number theory)*. 1972.
- 43 F.H. Ruymgaart. *Asymptotic theory of rank tests for independence*. 1973.
- 44 H. Bart. *Meromorphic operator valued functions*. 1973.
- 45 A.A. Balkema. *Monotone transformations and limit laws*. 1973.
- 46 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 1: the language*. 1973.
- 47 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 2: the compiler*. 1973.
- 48 F.E.J. Kruseman Aretz, P.J.W. ten Hagen, H.L. Oudshoorn. *An ALGOL 60 compiler in ALGOL 60, text of the MC-compiler for the EL-X8*. 1973.
- 49 H. Kok. *Connected orderable spaces*. 1974.
- 50 A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens, R.G. Fisker (eds.). *Revised report on the algorithmic language ALGOL 68*. 1976.
- 51 A. Hordijk. *Dynamic programming and Markov potential theory*. 1974.
- 52 P.C. Baayen (ed.). *Topological structures*. 1974.
- 53 M.J. Faber. *Metrizability in generalized ordered spaces*. 1974.
- 54 H.A. Lauwerier. *Asymptotic analysis, part 1*. 1974.
- 55 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 1: theory of designs, finite geometry and coding theory*. 1974.
- 56 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 2: graph theory, foundations, partitions and combinatorial geometry*. 1974.
- 57 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 3: combinatorial group theory*. 1974.
- 58 W. Albers. *Asymptotic expansions and the deficiency concept in statistics*. 1975.
- 59 J.L. Mijnheer. *Sample path properties of stable processes*. 1975.
- 60 F. Göbel. *Queueing models involving buffers*. 1975.
- 63 J.W. de Bakker (ed.). *Foundations of computer science*. 1975.
- 64 W.J. de Schipper. *Symmetric closed categories*. 1975.
- 65 J. de Vries. *Topological transformation groups, I: a categorical approach*. 1975.
- 66 H.G.J. Pijs. *Logically convex algebras in spectral theory and eigenfunction expansions*. 1976.
- 68 P.P.N. de Groen. *Singularly perturbed differential operators of second order*. 1976.
- 69 J.K. Lenstra. *Sequencing by enumerative methods*. 1977.
- 70 W.P. de Roever, Jr. *Recursive program schemes: semantics and proof theory*. 1976.
- 71 J.A.E.E. van Nunen. *Contracting Markov decision processes*. 1976.
- 72 J.K.M. Jansen. *Simple periodic and non-periodic Lamé functions and their applications in the theory of conical waveguides*. 1977.
- 73 D.M.R. Leivant. *Absoluteness of intuitionistic logic*. 1979.
- 74 H.J.J. te Riele. *A theoretical and computational study of generalized aliquot sequences*. 1976.
- 75 A.E. Brouwer. *Treelike spaces and related connected topological spaces*. 1977.
- 76 M. Rem. *Associations and the closure statements*. 1976.
- 77 W.C.M. Kallenberg. *Asymptotic optimality of likelihood ratio tests in exponential families*. 1978.
- 78 E. de Jonge, A.C.M. van Rooij. *Introduction to Riesz spaces*. 1977.
- 79 M.C.A. van Zuijlen. *Empirical distributions and rank statistics*. 1977.
- 80 P.W. Hemker. *A numerical study of stiff two-point boundary problems*. 1977.
- 81 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 1*. 1976.
- 82 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 2*. 1976.
- 83 L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH system*. 1979.
- 84 H.L.L. Busard. *The translation of the elements of Euclid from the Arabic into Latin by Hermann of Carinthia (?)*, books vii-xii. 1977.
- 85 J. van Mill. *Supercompactness and Wallmann spaces*. 1977.
- 86 S.G. van der Meulen, M. Veldhorst. *Torrix I, a programming system for operations on vectors and matrices over arbitrary fields and of variable size*. 1978.
- 88 A. Schrijver. *Matroids and linking systems*. 1977.
- 89 J.W. de Roever. *Complex Fourier transformation and analytic functionals with unbounded carriers*. 1978.
- 90 L.P.J. Groenewegen. *Characterization of optimal strategies in dynamic games*. 1981.

- 91 J.M. Geysel. *Transcendence in fields of positive characteristic*. 1979.
- 92 P.J. Weeda. *Finite generalized Markov programming*. 1979.
- 93 H.C. Tijms, J. Wessels (eds.). *Markov decision theory*. 1977.
- 94 A. Bijsma. *Simultaneous approximations in transcendental number theory*. 1978.
- 95 K.M. van Hee. *Bayesian control of Markov chains*. 1978.
- 96 P.M.B. Vitányi. *Lindenmayer systems: structure, languages, and growth functions*. 1980.
- 97 A. Federgruen. *Markovian control problems; functional equations and algorithms*. 1984.
- 98 R. Geel. *Singular perturbations of hyperbolic type*. 1978.
- 99 J.K. Lenstra, A.H.G. Rinnooy Kan, P. van Emde Boas (eds.). *Interfaces between computer science and operations research*. 1978.
- 100 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 1*. 1979.
- 101 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 2*. 1979.
- 102 D. van Dulst. *Reflexive and superreflexive Banach spaces*. 1978.
- 103 K. van Harn. *Classifying infinitely divisible distributions by functional equations*. 1978.
- 104 J.M. van Wouwe. *GO-spaces and generalizations of metrizability*. 1979.
- 105 R. Helmers. *Edgeworth expansions for linear combinations of order statistics*. 1982.
- 106 A. Schrijver (ed.). *Packing and covering in combinatorics*. 1979.
- 107 C. den Heijer. *The numerical solution of nonlinear operator equations by imbedding methods*. 1979.
- 108 J.V. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 1*. 1979.
- 109 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 2*. 1979.
- 110 J.C. van Vliet. *ALGOL 68 transput, part I: historical review and discussion of the implementation model*. 1979.
- 111 J.C. van Vliet. *ALGOL 68 transput, part II: an implementation model*. 1979.
- 112 H.C.P. Berbee. *Random walks with stationary increments and renewal theory*. 1979.
- 113 T.A.B. Snijders. *Asymptotic optimality theory for testing problems with restricted alternatives*. 1979.
- 114 A.J.E.M. Janssen. *Application of the Wigner distribution to harmonic analysis of generalized stochastic processes*. 1979.
- 115 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 1*. 1979.
- 116 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 2*. 1979.
- 117 P.J.M. Kallenberg. *Branching processes with continuous state space*. 1979.
- 118 P. Groeneboom. *Large deviations and asymptotic efficiencies*. 1980.
- 119 F.J. Peters. *Sparse matrices and substructures, with a novel implementation of finite element algorithms*. 1980.
- 120 W.P.M. de Ruyter. *On the asymptotic analysis of large-scale ocean circulation*. 1980.
- 121 W.H. Haemers. *Eigenvalue techniques in design and graph theory*. 1980.
- 122 J.C.P. Bus. *Numerical solution of systems of nonlinear equations*. 1980.
- 123 I. Yuhász. *Cardinal functions in topology - ten years later*. 1980.
- 124 R.D. Gill. *Censoring and stochastic integrals*. 1980.
- 125 R. Eising. *2-D systems, an algebraic approach*. 1980.
- 126 G. van der Hoek. *Reduction methods in nonlinear programming*. 1980.
- 127 J.W. Klop. *Combinatory reduction systems*. 1980.
- 128 A.J.J. Talman. *Variable dimension fixed point algorithms and triangulations*. 1980.
- 129 G. van der Laan. *Simplicial fixed point algorithms*. 1980.
- 130 P.J.W. ten Hagen, T. Hagen, P. Klint, H. Noot, H.J. Singt, A.H. Veen. *ILP: intermediate language for pictures*. 1980.
- 131 R.J.R. Back. *Correctness preserving program refinements: proof theory and applications*. 1980.
- 132 H.M. Mulder. *The interval function of a graph*. 1980.
- 133 C.A.J. Klaassen. *Statistical performance of location estimators*. 1981.
- 134 J.C. van Vliet, H. Wupper (eds.). *Proceedings international conference on ALGOL 68*. 1981.
- 135 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part I*. 1981.
- 136 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part II*. 1981.
- 137 J. Telgen. *Redundancy and linear programs*. 1981.
- 138 H.A. Lauwerier. *Mathematical models of epidemics*. 1981.
- 139 J. van der Wal. *Stochastic dynamic programming, successive approximations and nearly optimal strategies for Markov decision processes and Markov games*. 1981.
- 140 J.H. van Geldrop. *A mathematical theory of pure exchange economies without the no-critical-point hypothesis*. 1981.
- 141 G.E. Welters. *Abel-Jacobi isogenies for certain types of Fano threefolds*. 1981.
- 142 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 1*. 1981.
- 143 J.M. Schumacher. *Dynamic feedback in finite- and infinite-dimensional linear systems*. 1981.
- 144 P. Eigenraam. *The solution of initial value problems using interval arithmetic; formulation and analysis of an algorithm*. 1981.
- 145 A.J. Brentjes. *Multi-dimensional continued fraction algorithms*. 1981.
- 146 C.V.M. van der Mee. *Semigroup and factorization methods in transport theory*. 1981.
- 147 H.H. Tigelaar. *Identification and informative sample size*. 1982.
- 148 L.C.M. Kallenberg. *Linear programming and finite Markovian control problems*. 1983.
- 149 C.B. Huijsmans, M.A. Kaashoek, W.A.J. Luxemburg, W.K. Vietsch (eds.). *From A to Z, proceedings of a symposium in honour of A.C. Zaanen*. 1982.
- 150 M. Veldhorst. *An analysis of sparse matrix storage schemes*. 1982.
- 151 R.J.M.M. Does. *Higher order asymptotics for simple linear rank statistics*. 1982.
- 152 G.F. van der Hoeven. *Projections of lawless sequences*. 1982.
- 153 J.P.C. Blanc. *Application of the theory of boundary value problems in the analysis of a queueing model with paired services*. 1982.
- 154 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part I*. 1982.
- 155 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part II*. 1982.
- 156 P.M.G. Apers. *Query processing and data allocation in distributed database systems*. 1983.
- 157 H.A.W.M. Kneppers. *The covariant classification of two-dimensional smooth commutative formal groups over an algebraically closed field of positive characteristic*. 1983.
- 158 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part I*. 1983.
- 159 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 2*. 1983.
- 160 A. Rezus. *Abstract AUTOMATH*. 1983.
- 161 G.F. Helminck. *Eisenstein series on the metaplectic group, an algebraic approach*. 1983.
- 162 J.J. Dik. *Tests for preference*. 1983.
- 163 H. Schippers. *Multiple grid methods for equations of the second kind with applications in fluid mechanics*. 1983.
- 164 F.A. van der Duyn Schouten. *Markov decision processes with continuous time parameter*. 1983.
- 165 P.C.T. van der Hoeven. *On point processes*. 1983.
- 166 H.B.M. Jonkers. *Abstraction, specification and implementation techniques, with an application to garbage collection*. 1983.
- 167 W.H.M. Zijm. *Nonnegative matrices in dynamic programming*. 1983.
- 168 J.H. Evertse. *Upper bounds for the numbers of solutions of diophantine equations*. 1983.
- 169 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 2*. 1983.