Monotonicity in Markov Reward and Decision Chains: Theory and Applications

Monotonicity in Markov Reward and Decision Chains: Theory and Applications

Ger Koole

Department of Mathematics
VU University Amsterdam
De Boelelaan 1081a
Amsterdam
1081 HV
The Netherlands
koole@few.vu.nl

now

the essence of knowledge

Boston – Delft

Foundations and Trends® in Stochastic Systems

Foundations and Trends® in Stochastic Systems

Volume 1 Issue 1, 2006

Editorial Board

Editorial Scope

Foundations and Trends® in Stochastic Systems will publish survey and tutorial articles in the following topics:

- Brownian Models
- Logistics and Supply Chain Applications
- Markov Decision Processes
- Mathematical Finance and Insurance
- Point Processes
- Queuing Theory

- Reliability Theory
- Risk Analysis
- Simulation
- Stochastic Control
- Stochastic Networks
- Stochastic Optimization
- Stochastic Programming
- Telecom and Internet Applications

now

the essence of knowledge

Monotonicity in Markov Reward and Decision Chains: Theory and Applications

Ger Koole

*Department of Mathematics, VU University Amsterdam, De Boelelaan
1081a, Amsterdam 1081 HV, The Netherlands, koole@few.vu.nl*

Abstract

This paper focuses on monotonicity results for dynamic systems that
take values in the natural numbers or in more-dimensional lattices. The
results are mostly formulated in terms of controlled queueing systems,
but there are also applications to maintenance systems, revenue man-
agement, and so forth. We concentrate on results that are obtained
by inductively proving properties of the dynamic programming value
function. We give a framework for using this method that unifies results
obtained for different models. We also give a comprehensive overview
of the results that can be obtained through it, in which we discuss
not only (partial) characterizations of optimal policies but also appli-
cations of monotonicity to optimization problems and the comparison
of systems.

Contents

1

Introduction

Dynamic programming (DP) is a versatile method that can be used to analyze dynamical systems with a possibly stochastic transition mechanism. Above all, it is a numerical method for so-called Markov decision chains (MDCs), in which there are actions to be taken. In systems without actions DP can also be used to find performance indicators (as an alternative to the method that computes the stationary distribution first). These systems are called Markov reward chains, where the reward refers to the performance measure. However, DP can also be used as a theoretical means to obtain certain properties of Markov decision/reward chains. That is the subject of this paper.

The focus in this paper is on dynamic programming methods to obtain monotonicity results for queueing and other types of systems. The results that we obtain for MDCs can be used to characterize, partially or completely, optimal policies. In the context of Markov reward chains we obtain certain monotonicity results in the states, which can then be used in other optimization problems or to compare systems that, for example, differ in certain parameter values. Relevant results that are not obtained through DP are mentioned (but not proven) to give a complete picture.

2

Typical Results

This chapter gives an idea of the structural results that can be obtained with dynamic programming. We discuss two representative models and their variants and generalizations.

2.1 Server-Assignment Models

Consider the following controlled queueing model. Customers of m different types arrive according to some process at a service station with a single server. Class-i customers require an exponentially distributed service time with parameter $\mu(i)$. Each time unit a class-i customer spends waiting costs $c(i)$, service may be interrupted. How to assign the server as to minimize the long-run average costs?

The answer to this question is the so-called μc *rule*: the customers should be served in decreasing order of $\mu(i)c(i)$. When the server is serving class j and a customer of a class i with $\mu(i)c(i) > \mu(j)c(j)$ enters then the service should be interrupted and the server moves to class i. This result can be shown in a number of ways, one of them being dynamic programming (DP).

It can be generalized in a number of ways. An example is the criterion: instead of long-run average costs, we can consider expected discounted costs or even the expected costs at some time T. For all these criteria the μc rule is optimal.

For a number of interesting generalizations, it is convenient to reformulate the original question as follows: what are the conditions to the model to make that the policy that serves preemptively the customer with the smallest index first is optimal? We just saw that this *smallest index policy* (SIP) is optimal in case that $\mu(1)c(1) \geq \cdots \geq \mu(m)c(m)$.

Now we add one or more servers. Under the additional assumption that $\mu(1) \leq \cdots \leq \mu(m)$ the SIP is again optimal. This can be proven with dynamic programming. The proof uses the same techniques as the proof for the single-server case, and a considerable part is completely the same. The result holds not only for the linear cost structure, but also for other objectives. One of the objectives that is allowed is the time until the system is empty for the first time, the *makespan*. This shows that the *longest expected processing times first* policy (LEPT) is optimal for the problem of minimizing makespan. In the case of linear costs, it follows that the condition $\mu(1) \leq \cdots \leq \mu(m)$ allows us to take the probability that the costs exceed a certain level as direct costs. Thus, under this assumption, the μc rule not only minimizes the expected costs at T, but it also minimizes the costs at T in distribution. A final generalization made possible by the condition $\mu(1) \leq \cdots \leq \mu(m)$ is the fact that we can allow a certain *dependence* of the arrival process on the queue length: the arrival process can also be controlled using decisions that may depend on the numbers of customers. The SIP remains optimal. With this we can for example model more than one server in tandem, in which case SIP should be used at the final server.

Server-assignment models are discussed in detail in Section 9.1.

2.2   Routing Models

Let us look at another type of controlled queueing model, with a single customer class and two servers. Customers are assigned to a server at the moment of arrival. We try to minimize the total expected number

of customers in the system. When service times are exponentially distributed with rate $\mu(i)$ for server i, then the optimal routing policy can be characterized as follows: when the faster server has less or as many customers in queue as the slower server, then a new arrival should be assigned to the faster server. This result can also be obtained using dynamic programming. By symmetry arguments it leads to the *send to the shortest queue* policy (SQP) in the case of equal service rates. Our result, for the asymmetric case, characterizes the optimal policy in half of all possible configurations, and can be generalized in a number of ways (more than 2 queues, finite buffers, other costs and criterion, controlled arrivals, and so forth). But how about the other states, where the faster server has the longer queue? One would expect that the optimal policy would be monotone in the following way: if in state (x_1, x_2) (with x_i denoting the number of customers at server i) it is optimal to assign to queue 1, then the same holds true in $(x_1, x_2 + 1)$ and consequently in all states $(x_1, x_2 + y)$ with $y > 0$. This can indeed be shown using DP arguments. By symmetry arguments this leads to an increasing switching curve in the (x_1, x_2)-plane indicating the boundary between states where it is optimal to assign to queue 1 or 2. This result is only proven for 2 queues and a single server at each queue, but otherwise it can be generalized in many directions. For example, we can allow for dedicated arrivals streams, finite buffers, and customers that move from one queue to the other if it is optimal to do so. This allows us to model the following system: arrivals to queue 1 only, and the possibility to move queued customers to a second (often slower) server, but not back. Thus the monotonicity also holds for this system: it is only optimal to move servers to the slower server when there are more than a certain number of customers in queue, the *threshold*.

Threshold policies are best known from one-dimensional controlled queueing systems, where arriving customers are rejected when the total number of customers in the system exceeds the threshold. One of the threshold policies minimizes the weighted sum of rejection and holding costs. The proof of this result is by showing convexity of the so-called *dynamic programming value function*, as a function of the number of customers in the system. This convexity also holds for the uncontrolled single-server queue. Actually, a generalization of convexity to

two-dimensional lattices is at the basis of the threshold result described above. We saw how this convexity can be used for deriving properties of the optimal policy. However, it can also be used to compare controlled or uncontrolled queueing systems. As an example, consider two single-server queues which differ only in their arrival processes. One has a regular (possibly inhomogeneous) Poisson process as arrival process. The other has queue has arrival according to a Poisson process with a lower rate and batch arrivals such that the average number of arrivals is equal in both queues. It can be shown using DP that the expected queue length in the queue with batch arrivals is higher. By taking the right limits this result can be used to show that a fluid queue has a lower expected workload then the corresponding M/G/1 queue. Crucial to the proof is the convexity of the value function.

In the next chapter, we start with a discussion of the technical details of the method, by looking at the M/M/1 queue with admission control. They are discussed in their full generality in Section 9.3. "First-order" results for routing models (such as the optimality of the SQP) are treated in Section 9.2, "second-order" results (such as the existence of a switching curve) for two dimensions are discussed in Section 9.5.

2.3   Bibliographical Notes

Over the years a number of overviews of monotonicity results have appeared. We mention Stidham and Weber [77] and Stidham [75].

Puterman [65] is the standard reference for Markov decision chains.

3

The Framework

In this chapter we introduce the general framework. We start with a motivating example.

3.1 A Motivating Example

Consider an M/M/1 queue with arrival rate λ and service rate μ. At each arrival instant we are allowed to reject the arriving customer, at costs R per rejected customer. On the other hand, every customer in the system costs 1 per unit of time. What is the policy that minimizes the long-run average costs? *Value iteration*, a standard numerical method from Markov decision theory, combined with *uniformization*, to make the continuous-time problem discrete, can be used to solve this problem.

To see how uniformization works, we should realize that in all states except the empty state 0, there are two events possible: an arrival (which is possibly rejected), and a departure. In the empty state only an arrival can occur. If we allow for transitions from a state to itself then the rate out of each state except 0 is $\lambda + \mu$. By adding a *dummy* of *fictitious* transition from state 0 to itself with rate μ we find that the rate out of each state is $\lambda + \mu$, without exception, for every policy. Now

we consider the *embedded* discrete-time Markov decision chain (MDC) by looking at the system only at transition instants. They occur according to a Poisson process with rate $\lambda + \mu$. The instantaneous holding costs for the embedded chain count for the whole period until the next transition, and should therefore be multiplied with $(\lambda + \mu)^{-1}$.

Now we define the dynamic programming (DP) value function. Let $V_n(x)$ be the minimal expected total costs for the embedded decision chain if there are n more jumps to go, $n > 0$. We take $V_0(x) = 0$. Then we can express $V_n(x)$ in $V_{n-1}(x)$ in the following way. First the holding costs until the next jump $x/(\lambda + \mu)$ are incurred. Then one of two things can happen: a departure, with probability $\mu/(\lambda + \mu)$, or an arrival, with probability $\lambda/(\lambda + \mu)$. In the former case, we move to $x - 1$ (or to x, if $x = 0$). In the latter case we have to decide whether to accept the arriving customer. As $V_n(x)$ are the minimal *expected* total costs, we can take the expectation with respect to the next state:

$$
\begin{aligned}
V_n(x) \;=\; & \frac{x}{\lambda + \mu} + \frac{\mu}{\lambda + \mu} V_{n-1}((x-1)^+) \\
& + \frac{\lambda}{\lambda + \mu} \min\{R + V_{n-1}(x), V_{n-1}(x+1)\}. \qquad (3.1)
\end{aligned}
$$

For each $n > 0$ and every state $x \in \mathbb{N}_0$ there is a minimizing action: rejection or admission. For fixed n we call this function $\mathbb{N}_0 \to \{\text{rejection}, \text{admission}\}$ a *policy*. For $n \to \infty$ this policy converges to the *average optimal policy*, that is, the policy that minimizes the long-run expected average costs (see Section 4.1 for more details). Now consider threshold policies. A threshold policy is characterized by the fact that if rejection is optimal in x, then rejection is also optimal in $x + 1$. A sufficient condition for this is

$$
R + V_{n-1}(x) \leq V_{n-1}(x+1) \Rightarrow R + V_{n-1}(x+1) \leq V_{n-1}(x+2).
$$

This is satisfied if

$$
R + V_{n-1}(x) - V_{n-1}(x+1) \geq R + V_{n-1}(x+1) - V_{n-1}(x+2),
$$

which is equivalent to convexity on $\mathbb{N}_0$:

$$
2V_{n-1}(x+1) \leq V_{n-1}(x) + V_{n-1}(x+2).
$$

The crucial step in showing the optimality of a threshold policy is showing the convexity of the value function. This convexity holds for every n, and therefore the average optimal policy is also of threshold type. The convexity can be proven by induction to n.

The inductive proof consists of proving the induction step for every component of the value function. Thus, assuming that $V_{n-1}(x)$ is convex, we prove that the direct costs $x/(\lambda + \mu)$, the departure term $V_{n-1}((x-1)^+)$, and the arrival term $\min\{R + V_{n-1}(x), V_{n-1}(x+1)\}$ are all convex functions of x. Then we use the fact that convex combinations and sums of convex functions are again convex. From this it follows that $V_n(x)$ is convex and the induction step is done.

It should be noted that to prove that $V_{n-1}((x-1)^+)$ is convex we also need that $V_{n-1}(x)$ is increasing in x.

An important observation is the fact that the proof of the convexity is largely the same if we change part of the model. Take for example the uncontrolled M/M/1 queue with linear holding costs. Although there are no actions, we can again formulate the value function, by replacing $\min\{R + V_{n-1}(x), V_{n-1}(x+1)\}$ by $V_{n-1}(x+1)$ in the value function (3.1). The convexity and increasingness can again be proven inductively. What changes are the arguments used for proving the convexity for the arrival term $V_{n-1}(x+1)$.

In this example we clearly see the central role of the functional properties convexity and increasingness, and of propagating these properties through the building blocks of the value function, which are related to the possible *events* in the queueing system. New events can easily be added by checking whether increasingness and/or convexity propagate. This way we get a set of events for each set of functional properties. Each value function then has the properties belonging to the corresponding set of events. Up to now we encountered the events that model a single server, admission control, and uncontrolled arrivals. The costs can also be interpreted as an event, as well as taking linear combinations. Now the value function consists entirely of event *operators*, and its properties follow directly from the properties of the composing events. Studying the equivalent discrete-time model becomes almost trivial: we simply replace the uniformization operator by a concatenation of operators as they occur on the transition times. It follows

immediately that the results for the continuous-time model hold as well for this discrete-time model. Many other models can be studied by adding new events. Events that we consider include: an event for an environment that allows us to have general arrival processes, controlled departures, and multiple servers.

3.2   Operators and Events

We now define formally the operators and the value functions that we consider. We consider models with m dimensions, often queues, and a separate environment. The queues take values in $\mathbb{N}_0 = \{0, 1, \ldots\}$. The same holds for the environment. (Other types of state spaces are shortly discussed in Chapter 10.) Event operators map a function from the states to $\mathbb{R}$ to a new function from the states to $\mathbb{R}$, and value functions are basically concatenations of event operators. This leads to the following definitions.

Definition 3.1. (States, operators, and value function)

— For some $m \in \mathbb{N}$ we define the state space $\mathcal{X} = \mathbb{N}_0^{m+1}$;
— Let $\mathcal{V}$ be the set of functions from $\mathcal{X}$ to $\mathbb{R}$ (elements of $\mathcal{V}$ can be thought of as real-valued vectors the size of the state space);
— For some $k \in \mathbb{N}$ we define *event operators* $T_0, \ldots, T_{k-1}$ as follows: $T_i : \mathcal{V}^{l_i} \to \mathcal{V}$, $l_i \in \mathbb{N}$, for $i = 1, \ldots, k$;
— The *value function* V_n for $n \in \mathbb{N}$ is defined recursively using the auxiliary $V_n^{(0)}, \ldots, V_n^{(k)}$:

  — $V_n^{(k)} = V_{n-1}$;

  — $V_n^{(j)} = T_j(V_n^{(k_1)}, \ldots, V_n^{(k_{l_j})})$, $j = 0, \ldots, k-1$, for $k_1, \ldots, k_{l_j}$ such that $j < k_1, \ldots, k_{l_j} \leq k$;

  — $V_n = V_n^{(0)}$.

We model (3.1) to make the ideas behind the definition clear. Take $m = 1$. Then the states are two-dimensional: the 0th component is the environment, the 1st component the actual queue length. Take $k = 4$,

and define

$$T_0 f(x) \;=\; \frac{x_1}{\lambda + \mu} + f(x) \text{ (the direct costs)},$$

$$T_1(f_1, f_2)(x) \;=\; \frac{\mu}{\lambda + \mu} f_1(x) + \frac{\lambda}{\lambda + \mu} f_2(x) \text{ (uniformization)},$$

$$T_2 f(x) \;=\; f((x - e_1)^+) \text{ (departures)},$$

and

$$T_3 f(x) = \min\{R + f(x), f(x + e_1)\} \text{ (admission control)}.$$

Now take $V_n^{(0)} = T_0 V_n^{(1)}$, $V_n^{(1)} = T_1(V_n^{(2)}, V_n^{(3)})$, $V_n^{(2)} = T_2 V_n^{(4)}$, and $V_n^{(3)} = T_3 V_n^{(4)}$. This leads to

$$V_n(x) = T_0(T_1[T_2(V_{n-1}), T_3(V_{n-1})]). \tag{3.2}$$

Once written out this is completely equivalent to (3.1), except for the fact that there is an environment state that remains constant. This will become relevant as soon as we model non-Poissonian arrival streams.

The central idea of how to use the value function as we defined it in Definition 3.1 is summarized in the following theorem:

Theorem 3.1. Let $\mathcal{F}$ be some class of functions from $\mathcal{X}$ to $\mathbb{R}$, V_n as defined in Definition 3.1, and $V_0 \in \mathcal{F}$. If, for all i, for $f_1, \ldots, f_{l_i} \in \mathcal{F}$ it holds that $T_i(f_1, \ldots, f_{l_i}) \in \mathcal{F}$, then also $V_n \in \mathcal{F}$ for all n.

We illustrate the use again with the value function (3.1). Take $\mathcal{F}$ the set of increasing and convex functions. It can be shown (see Section 9.3) that $\mathcal{F}$ propagates through the operators $T_0, \ldots, T_3$, as defined below Definition 3.1.

Depending on the combination of operators and the results we try to obtain we use different classes of functions. For each class $\mathcal{F}$ and operator T we derive the "biggest" $\mathcal{F}'$ with $\mathcal{F}' \subset \mathcal{F}$ such that if $f_1, \ldots, f_l \in \mathcal{F}'$ holds that $T(f_1, \ldots, f_l) \in \mathcal{F}$. This is notated as follows: $T \colon \mathcal{F}' \to \mathcal{F}$. Let us apply this to the queue with admission control. Denote the class of increasing function with I and the class of convex function with Cx.

It is shown in Section 9.3 that:

— T_0, T_1, T_3: $I \to I$, $Cx \to Cx$;
— T_2: $I \to I$, $I \cap Cx \to Cx$.

In fact, for T_2 convexity by itself does not propagate: $\mathcal{F}'$ should be restricted to the class of convex and increasing functions. To obtain the optimality of the threshold policy convexity is sufficient, but to propagate this property through T_2 we also need I. Luckily, I propagates through all operators. Thus the class to take is $I \cap Cx$, leading to:

— T_0, T_1, T_2, T_3: $I \cap Cx \to I \cap Cx$.

Now we can use Theorem 3.1 to show that V_n is increasing and convex. This shows the optimality of a threshold policy for this model, using the reasoning on Page 7.

In the next chapters, we will look at different groups of classes of functions. We do this in a systematic way. Chapter 4 deals with the criteria and the role the environment (the 0th state variable) plays. In Chapter 5, we define all the operators we consider, and in Chapter 6 we consider all the inequalities we are interested in, and some of their relations. In Chapter 7, we consider all operators one by one and we determine under which addtional conditions the operators preserve the functional inequalities defined in Chapter 6. In Chapter 8, we determine for the controlled operators what the consequences are for the optimal actions as functions of the states (e.g., threshold optimality). Then, in Chapter 9, we consider sets of functions and determine which set of operators propagate for each set. As an example, in Section 9.3 we look at one-dimensional convex functions, for which the model taken as example in this chapter is the main representative.

3.3   Bibliographical Notes

In several papers unifying frameworks for the study of monotonicity results have been introduced. Veatch and Wein [81], Glasserman and Yao [26] and Altman and Koole [6] all consider a class of functions related to submodularity (see Sections 6.3 and 9.4), resulting in the

fact that the occurence of an event increases the optimal rate for any other event. This limits the number of models covered by these papers.

References [6] and [26] both give the *score space* a central place. Basically, the score space counts the number of times that different events have occurred, and the actual state is a function of the current score. As a result, the score space has more dimensions than the state space. For this reason stronger results can often be obtained by working directly in the state space. See Section 9.4 for more details.

The current framework was introduced in Koole [40] (where it was called *event-based DP*).

4

Criteria and the Environment

The main objective of this paper is to study certain classes of value functions of the form given in Definition 3.1, to look which classes of functions propagate through the operators used in the value functions, in order to obtain $V_n \in \mathcal{F}$ for certain classes of functions $\mathcal{F}$, using Theorem 3.1.

In this chapter we look at the relation between the original system and the value function. That is, we look how the structural result (of the type $V_n \in \mathcal{F}$, or some consequence of that concerning the policy at stage n) translates back to the original problem, for example in terms of the average cost long-run optimal policy.

We look at different cost criteria, discrete and continuous-time models, and the possibility of an environment (the 0th state variable) that regulates the rates at which operators occur. We also pay attention to the special role that direct cost play.

4.1 Cost Criteria and Markov Decision Processes

For the moment we assume a discrete-time setting. In the literature, we encounter for (controlled) stochastic processes different ways to

evaluate performance. Concentrating on costs, we differentiate between long-run average costs, long-run discounted costs, total costs over a finite (possibly random) horizon, and costs at a (possibly random) time instant.

We assume that the system under consideration can be described by a (time-homogeneous) Markov decision chain (MDC) or process. Then any of the criteria above can be evaluated by the use of a value function of the following form, $n \in \mathbb{N}$:

$$V_n(x) = \min_{a \in \mathcal{A}} \left\{ c(x,a) + \alpha_n \sum_{y \in \mathcal{X}} p(x,a,y) V_{n-1}(y) \right\}, \qquad (4.1)$$

with $p(x,a,\cdot)$ a probability distribution for state $x \in \mathcal{X}$ and action $a \in \mathcal{A}$, $c(x,a) \in \mathbb{R}$, $\alpha_n \in [0,1]$ for all n, and V_0 given.

Let us look at the criteria one by one. If $\alpha_n = 1$ and $V_0(x) = 0$, then $V_n(x)$ represents the minimal total costs for a horizon of length n for initial state $x \in \mathcal{X}$. On the other hand, if $c(x,a) = 0$ but $V_0(x) = C(x)$ for some cost function C, then $V_n(x)$ represents the total minimal expected costs after n time units.

By choosing all $\alpha_n = 1$ the costs at every time instant are equally weighted in the total costs. By taking $\alpha_n < 1$ early costs are weighted more (and vice versa). For a horizon n, costs after k time units are weighted with a factor $\alpha_n \cdots \alpha_{n-k+1}$. It is common to take $\alpha_n = \alpha$, costs after k time units are weighted with a factor α^k. This is called discounting, with the interest rate r is related to α as follows: $\alpha = (1+r)^{-1}$, i.e., $r = (1-\alpha)/\alpha$. An alternative way to look at discounting is as total costs with a random horizon: $1 - \alpha_n$ is the probability of an horizon of length 1, $\alpha_n(1 - \alpha_{n-1})$ is the probability of an horizon of length 2, etc.

More common objectives are the long-run discounted and long-run average costs. For $\alpha_n = \alpha < 1$ the value function V_n converges to the minimal expected long-run discounted costs, under certain conditions. Discounting can be seen as a random geometrically distributed horizon. If $\alpha_n = 1$ for all n then V_n usually does not converge, not even for well-constructed models. Instead, $V_{n+1}(x) - V_n(x)$ converges to the minimal long-run average costs, under certain conditions (which are more restrictive than those for the discounted case). Elaborating on

these conditions falls outside the scope of this paper; see for example Puterman [65, Section 8.10] for more details.

It is tempting to derive results only for costs at a fixed horizon, thus $c(x,a) = 0$ and $V_0(x) = C(x)$. Results involving (discounted) costs over multiple periods can then be derived by summing in the adequate way over the horizon. However, an important complication of using finite-horizon criteria is the fact that optimal policies depend on the horizon, they are time-dependent. Thus considering costs at the horizon does not suffice, unless we have a system without actions or if the optimal policy is the same for all horizons. If the latter is the case then the policy is called *myopic*, because the optimal policy for horizon 1 is optimal for all n.

As an example, consider the queue with controlled admissions of Equation (3.1), with parameter values $\lambda = \mu = 1$ and $R = 4$. For small n admission is always optimal: the costs of holding a customer in queue over a short period is cheaper than rejecting a customer. For $n = 10$ it is optimal to reject in all states except $\{0,1,2\}$, and from $n = 11$ on the optimal policy is equal to the long-run optimal policy: accept only in states 0 and 1.

However, Equation (3.1), and, in general, the class of value functions defined in Definition 3.1, is not in the standard form of Equation (4.1). But, if we choose the operators T_i of the form:

$$T_i(f_1, \ldots, f_{l_i})(x) = \min_{a \in \mathcal{A}_i} \left\{ c_i(x,a) + \sum_{j=1}^{l_i} \sum_{y \in \mathbb{N}_0^m} p_i^j(x,a,y) f_j(y) \right\}, \quad (4.2)$$

with $\sum_j \sum_y p_i^j(x,a,y) = 1$, and one operator of the form $f \to C + \alpha f$ then it is possible to choose c and p in (4.1) such that both definitions are equivalent (see Theorem 2.2 of [40]).

It is this latter operator that represents the discounting.

Definition 4.1. The operator T_{disc} is defined by:

   — $T_{\text{disc}} f(x) = C(x) + \alpha f(x)$ for some real-valued function C

The current definition is for a constant discount rate. To model discount factors that depend on n the operators should depend on n

as well (or, alternatively, α could depend on the environment, with the environment "counting" time: see Section 4.2).

As an example of the use of T_{disc}, consider Equation (3.1) with discounting. Then in Equation (3.2) operator T_0 gets replaced by T_{disc} with $C(x) = x_1/(\lambda + \mu)$.

4.2   The Environment

Dimension 0 of our state space will be used for the environment. The environment will allow us to model all kinds of variations in the model. Examples are server vacations and varying probabilities by which events occur. As long as the evolution of the environment is not influenced by the state of dimensions 1 upto m then all the monotonicity results of the following chapters still hold.

An important application of an environment is for modeling continuous-time models with general "inter-event" distributions; see Section 4.3.

Definition 4.2. The operator T_{env} is defined by:

$$- T_{\text{env}}(f_1, \ldots, f_l)(x) = \sum_{y \in \mathbb{N}_0} \lambda(x_0, y) \sum_{j=1}^{l} q^j(x_0, y) f_j(x^*),$$

where x^* is equal to x with the 0th component replaced by y.

In this definition $\lambda(x, y)$ is the probability with which the environment moves from x to y, and $q^j(x, y)$ is the probability of a type j arrival or *mark* during such a transition.

In general the monotonicity results do not hold anymore when the environment depends on the state of dimensions 1 upto m. Counterexamples are easily constructed. However, there is one type of dependency that can be propagated through certain useful classes of functions, and that is when the dependence is modeled through actions taken in the environment. Usually we work in a setting where it is the objective to minimize costs. The environment can either be cost-minimizing, or cost-maximizing. The latter situation makes it a zero-sum game. (Note by the way that there are no methodological objections of including maximizations in other operators.)

This leads to the following definitions:

Definition 4.3. The operators $T_{\min}$ and $T_{\max}$ are defined by:

$$- \quad T_{\min}(f_1,\ldots,f_l)(x) = \min_a \left\{ \sum_{y \in \mathbb{N}_0} \lambda(x_0,a,y) \sum_{j=1}^{l} q^j(x_0,a,y) f_j(x^*) \right\};$$

$$- \quad T_{\max}(f_1,\ldots,f_l)(x) = \max_a \left\{ \sum_{y \in \mathbb{N}_0} \lambda(x_0,a,y) \sum_{j=1}^{l} q^j(x_0,a,y) f_j(x^*) \right\}.$$

4.3 Continuous Time

In this chapter we considered up to now discrete-time models. The current section will show how these models can be used to model certain continuous-time systems. In the literature there are three different methods by which we can model continuous-time systems in discrete time: uniformization, discretization, or embedding. We discuss all three, starting with uniformization (see also Section 3.1). The framework in which we discuss it is as follows. We assume we have a number of events (arrivals, departures, and so forth), each controlled by a point process (or a marked point process, the mark indicating the type of event). (Note the similarities with semi-Markov (decision) processes.) The exponential distribution plays a crucial role in uniformization, and indeed, the simplest case is when the time between two events of the same type has an exponential distribution, independent of all other inter-event times, with a parameter depending on the event type, say λ_j for event type j. Now the time until the next event is exponentially distributed with parameter $\sum_i \lambda_i$. The probability that the event j happens at a jump time is $\lambda_j / \sum_i \lambda_i$. This leads to the following operator that models uniformization, which is actually a special case of T_{env}.

Definition 4.4. The operator T_{unif} is defined by:

$$- \quad T_{\mathrm{unif}}(f_1,\ldots,f_l)(x) = \sum_j p(j) f_j(x) \quad \text{with } p(j) > 0 \text{ for all } j.$$

In the definition, $p(j) = \lambda_j / \sum_i \lambda_i$; normally they would sum to 1.

The operator T_{env} can be used to model systems with general "inter-event" distributions. This follows from the fact that Markov Arrival Processes (MAPs), or marked point processes, can be modeled with an environment. An MAP is a Markov process that at transitions generates events of different types. In the current context these arrivals generate occurrences of operators operating on dimensions 1 upto m. It can be shown that the set of MAPs is dense in the class of all arrival processes. This way monotonicity results for general arrival processes can be obtained, almost without additional effort. What is crucial here is that the events occur independent of the state components $x_1, \ldots, x_m$. Thus, in general, this cannot be applied to departure processes, because when a queue is empty then the departure process is temporarily stopped. Note that for exponential service times this did not pose a problem: we worked with a continuing process of potential departures, that only became real departures when the queue was non-empty. Thus an M/G/1 queue cannot be modeled with this; two parallel exponential queues (even with dependent arrival processes) can.

Uniformization is an elegant and powerful method, but it has a number of limitations. When used for modeling non-exponential inter-event times, then the state space has an additional component that can be computationally burdensome. A way around this is embedding if there is only one non-exponential inter-event time. Next to that, it is hard to use uniformization for finite-horizon models. Then discretization can be a better alternative. Let us discuss embedding first.

Consider a system with a single generally distributed inter-event distribution and one exponential inter-event distribution. Instead of approximating the generally distributed inter-event distribution by a phase-type distribution and making the phase dimension 0, which is essentially what uniformization does, we can also take the arrival epochs as the transition epochs of the value function. The consequence is that between jump times, i.e., during each inter-arrival time, a geometric number of potential departures occur.

As an example, consider the controlled GI/M/1 queue, the basic model of Stidham [73]. This can be modeled by

$$V_{n+1} = T_{\text{disc}}\big(T_{\text{CA}} T_{\text{unif}} \big[V_n, T_{\text{D}} V_n, T_{\text{D}}^2 V_n, \ldots\big]\big),$$

with T_{CA} the controlled arrival operator and with T_{D}^{j} the j-fold con-volution of the departure operator T_{D}, and with $p(j)$, the coefficient of $T_{\mathrm{D}}^{j} V_n$, the probability that j (potential) departures occur during an inter-arrival time. (See Chapter 5 for the definitions of T_{CA} and T_{D}.)

Embedding can also be useful if the transition rates are unbounded, as in the M/M/∞ queue. Note that in the M/M/∞ queue there is no lower bound for the expected time that we stay in a state. Therefore uniformization cannot be used. A way around is only looking at arrival instances, again with the consequence that there is a random number of departures during each period. This approach was used in Hariharan *et al.* [30]. In Liang and Kulkarni [49] another approach was chosen for a retrial queue that can have an unbounded number of customers in orbit: the total rate out of the orbit is bounded from above.

Uniformization is the obvious choice for long-run average and dis-counted cost problems. It can also be used for finite-horizon criteria, but in that case the total number of jumps becomes finite and random. In general, optimal policies depend on the stage n; which n to use? For this reason we can only use this method for systems in which the optimal policy does not depend on n, i.e., uncontrolled systems or sys-tems in which the optimal policy is myopic (i.e., for all n equal, see Page 17). An alternative method for non-myopic systems in this case is time-discretization, i.e., looking at the systems at discrete points in time with short times in between. As the lengths of these intervals approaches 0 the performance approaches that of the continuous-time system. At the same time the probability of more than 1 event approaches 0. Dis-cretization has an operator equal to uniformization, but with one of the functions f_j equal to the identity function.

There is an important difference between discounting and embed-ding in the case of controlled models. Optimal policies depend in gen-eral on the entire state. If there is an environment, then there will be a policy for every state of the environment. In the case of embedding without environment there will be a single policy.

There is one final issue to be discussed concerning continuous-time systems. There are systems where actions are taken continuously over time and those where actions are only taken at jump times. To illustrate the different types of actions consider again the queue with admission

control. The action whether or to admit a customer plays only at the arrival instant which is a jump moment. Now suppose we can also control the service rate, depending on the number of customers in queue. It is interesting to consider as well the possibility to change the service rate between jump times, leading to a situation where one continuously wants to determine the optimal policy. However, uniformization and embedding do not allow for actions changing inbetween epochs. Discretization does neither, but the time between decision epochs goes to 0, thus it approaches continuous control. Note however that for controlled models uniformization is mostly used for the infinite-horizon criteria. Whether or not the action can be changed between jumps is not an issue: the optimal policy is time-homogeneous, so there is no reason to change action between jumps. For the finite horizon and a myopic optimal policy there is neither a reason to change actions.

4.4   Bibliographical Notes

Uniformization was first used in the setting of proving monotonicity in Lippman [51]. Serfozo [70] made some useful extensions to this method. Discretization results that can be applied in the current context were obtained in Van Dijk [79].

The concept of the controlled environment is introduced in Hordijk and Koole [33]. Putting this in a game-theory setting, with both minimization and maximization in the same value function, was done in Altman and Koole [4].

The result that MAPs are dense in the class of all arrival processes can be found in Asmussen and Koole [11].

5

Operators

In this chapter we define all operators that are used in the subsequent chapters. We start by repeating the generic operators from Chapter 4.

Definition 5.1. We define the following generic operators, where x^* is equal to x with the 0th component replaced by y:

— $T_{\text{disc}} f(x) = C(x) + \alpha f(x)$ for some real-valued function C;

— $T_{\text{env}}(f_1, \ldots, f_l)(x) = \sum_{y \in \mathbb{N}_0} \lambda(x_0, y) \sum_{j=1}^{l} q^j(x_0, y) f_j(x^*)$;

— $T_{\min}(f_1, \ldots, f_l)(x) = \min_a \left\{ \sum_{y \in \mathbb{N}_0} \lambda(x_0, a, y) \sum_{j=1}^{l} q^j(x_0, a, y) f_j(x^*) \right\}$;

— $T_{\max}(f_1, \ldots, f_l)(x) = \max_a \left\{ \sum_{y \in \mathbb{N}_0} \lambda(x_0, a, y) \sum_{j=1}^{l} q^j(x_0, a, y) f_j(x^*) \right\}$.

Next we define operators that are used to model arrivals of customers.

Definition 5.2. We define the following arrival operators, for $1 \leq i \leq m$:

— $T_{\mathrm{A}(i)} f(x) = f(x + e_i)$;
— $T_{\mathrm{CA}(i)} f(x) = \min\{c + f(x), c' + f(x + e_i)\}$, $c, c' \in \mathbb{R}$;
— $T_{\mathrm{FS}(i)} f(x) = \begin{cases} (1 - \frac{x_i}{B}) f(x + e_i) + \frac{x_i}{B} f(x) & \text{if } x_i \leq B, \\ f(x) & \text{otherwise}; \end{cases}$
— $T_{\mathrm{CAF}} f(x) = \min\{c + f(x), f(x + \sum_{i=1}^{m} e_i)\}$;
— $T_{\mathrm{R}} f(x) = \min_{1 \leq i \leq m} f(x + e_i)$.

Operator $T_{\mathrm{A}(i)}$ models arrivals to dimension ("queue") i. Depending on the actual form of the value function, especially the environment, this can be used to model for example Poisson arrivals. The CA in $T_{\mathrm{CA}(i)}$ stands for "controlled arrivals," it can be used to model admission control to a queue. $T_{\mathrm{FS}(i)}$ is the finite-source operator to model Engset-type models. T_{CAF} models controlled simultaneous arrivals (a "fork"), and T_{R} models routing to one of the queues in I.

Note that the set I can be different for every operator (or even every time an operator is used).

Next we define operators that are used to model departures of customers.

Definition 5.3. We define the following departure operators, $1 \leq i \leq m$, and $I \subset \{1, \ldots, m\}$ some index set:

— $T_{\mathrm{D}1(i)} f(x) = f((x - e_i)^+)$;
— $T_{\mathrm{D}(i)} f(x) = \mu(x_i) f((x - e_i)^+) + (1 - \mu(x_i)) f(x)$ with $0 \leq \mu(x) \leq 1$ for all $x \in \mathbb{N}$ and $\mu(0) = 0$;
— $T_{\mathrm{PD}} f(x) = \sum_{1 \leq i \leq m} \mu(i) f((x - e_i)^+)$ with $\sum_{1 \leq i \leq m} \mu(i) = 1$;
— $T_{\mathrm{CD}(i)} f(x)$
$$= \begin{cases} \min_{\mu \in [0,1]} \{c(\mu) + \mu f(x - e_i) + (1 - \mu) f(x)\} & \text{if } x_i > 0, \\ c(0) + f(x) & \text{otherwise}, \end{cases}$$
with $c(\mu) \in \mathbb{R}$ for all $\mu \in [0,1]$, assuming that the minimum always exists;

— $T_{\mathrm{MS}}f(x)$

$$= \begin{cases} \min_{j \in I : x_j > 0} \{\mu(j)f(x - e_j) + (1 - \mu(j))f(x)\} & \text{if } \sum_{j \in I} x_j > 0, \\ f(x) & \text{otherwise,} \end{cases}$$

for $\mu(j) \leq 1$;

— $T_{\mathrm{MMS}}f(x)$

$$= \begin{cases} \dfrac{1}{s} \min_{i_1, \dots, i_s \in I : \sum_{k \in I} I\{i_k = j\} \leq x_j} \left\{ \sum_{k=1}^{s} (\mu(i_k)f(x - e_{i_k}) + (1 - \mu(i_k))f(x)) \right\} \\ \hfill \text{if } \sum_{j \in I} x_j \geq s, \\ \dfrac{1}{s} \sum_{j \in I} x_j \big(\mu(j)f(x - e_j) + (1 - \mu(j))f(x)\big) + \dfrac{s - \sum_{j \in I} x_j}{s} f(x) \\ \hfill \text{otherwise,} \end{cases}$$

for $\mu(j) \leq 1$.

Operator $T_{\mathrm{D1}(i)}$ models a single server. It is a special case of $T_{\mathrm{D}(i)}$, where the departure probability is an arbitrary function of the number of customers x_i. This can be used to model for example a multi-server queue. Sometimes we require additional conditions for the departure probability function, for example concavity. PD means parallel departures, T_{PD} is in fact of the form $T_{\mathrm{unif}}(T_{\mathrm{D1}(1)}, \dots, T_{\mathrm{D1}(m)})$. $T_{\mathrm{CD}(i)}$ is the controlled departure operator, MS stands for movable server. Finally MMS stand for multiple movable servers. What makes the definition of T_{MMS} complicated is the fact that different servers cannot work at the same customer.

Finally we define operators for moving from one queue to the next.

Definition 5.4. We define the tandem departure operators, for $1 \leq i \leq m$:

— $T_{\mathrm{TD1}(i)}f(x) = \begin{cases} f(x - e_i + e_{i+1(\mathrm{mod}\,m)}) & \text{if } x_i > 0, \\ f(x) & \text{otherwise}; \end{cases}$

— $T_{\mathrm{CTD}(i)}f(x)$

$$= \begin{cases} \min_{\mu \in [0,1]} \{c(\mu) + \mu f(x - e_i + e_{i+1(\mathrm{mod}\,m)}) + (1 - \mu)f(x)\} & \text{if } x_i > 0, \\ c(0) + f(x) & \text{otherwise}; \end{cases}$$

$$
- \ T_{\mathrm{TD}(i)}f(x) = \begin{cases} \frac{x_i}{S}f(x - e_i + e_{i+1(\mathrm{mod}m)}) + \frac{S-x_i}{S}f(x) & \text{if } x_i < S, \\ f(x - e_i + e_{i+1(\mathrm{mod}m)}) & \text{otherwise;} \end{cases}
$$

$$
- \ T_{\mathrm{MTS}}f(x)
$$
$$
= \begin{cases} \displaystyle\min_{j \in I : x_j > 0}\left\{ \sum_{k=0}^{m} \mu(i,k)f(x - e_i + e_k) \right\} & \text{if } \sum_{j \in I} x_j > 0, \\ f(x) & \text{otherwise,} \end{cases}
$$
where $\sum_{k=0}^{m} \mu(i,k) = 1$ for all i, $\mu(i,j) = 0$ for all i and $0 < j < i - 1$, and $e_0 = 0$.

Operator $T_{\mathrm{TD1}(i)}$ is an uncontrolled single server that feeds the next queue. $T_{\mathrm{TD1}(i)}$ feeds queue 1, making a loop possible. $T_{\mathrm{CTD1}(i)}$ adds the possibility of control, $T_{\mathrm{TD}(i)}$ multiple servers. T_{MTS} is a generalization of T_{MS}: it is a movable tandem server. Here $\mu(i,k)$ represents the probability that a customer in queue i which is being served moves to queue k. $k = 0$ means leaving the system. Note that feedback is possible to lower numbered queues and the next higher queue.

6

Inequalities

In this chapter we define all inequalities that we use, and we show the relations between different (combinations of) inequalities. We use increasing and decreasing in the non-strict sense.

All functions we define have as domain $\mathbb{N}_0^{m+1}$, where the dimensions are numbered $0, \ldots, m$. Recall that dimension 0 is the environment.

6.1   First-Order Conditions

In this section we define the first-order inequalities.

Definition 6.1. We define the following sets of functions:

— $f \in I(i)$ if

$$f(x) \leq f(x + e_i)$$

for all $x \in \mathbb{N}_0^{m+1}$, $1 \leq i \leq m$;
— $I = I(1) \cap \cdots \cap I(m)$;
— $f \in \mathrm{UI}(i)$ if

$$f(x + e_{i+1}) \leq f(x + e_i)$$

for all $x \in \mathbb{N}_0^{m+1}$, $1 \leq i < m$;

— UI = UI(1) $\cap \cdots \cap$ UI($m - 1$);
— $f \in w\text{UI}(i)$ if

$$\mu(i)f(x + e_{i+1}) + (1 - \mu(i))f(x + e_i + e_{i+1})$$
$$\leq \mu(i+1)f(x + e_i) + (1 - \mu(i+1))f(x + e_i + e_{i+1})$$

for all $x \in \mathbb{N}_0^{m+1}$, $1 \leq i < m$, for given constants $0 < \mu(j) \leq 1$, $1 \leq j \leq m$;
— $w\text{UI} = w\text{UI}(1) \cap \cdots \cap w\text{UI}(m - 1)$;
— $f \in g\text{UI}(i)$ if

$$\sum_{k=0}^{m} \mu(i,k)f(x + e_{i+1} + e_k) \leq \sum_{k=0}^{m} \mu(i+1,k)f(x + e_i + e_k),$$

$1 \leq i < m$, $e_0 = 0$, for given constants $\mu(j,k)$ such that $\sum_{k=1}^{m} \mu(j,k) = 1$, $1 \leq j \leq m$;
— $g\text{UI} = g\text{UI}(1) \cap \cdots \cap g\text{UI}(m - 1)$.

The class I speaks for itself: $f \in I$ if f is non-decreasing in all components.

When $f \in \text{UI}$ then it is advantageous to have customers in higher-numbered queues, further down the line in case of a tandem system. UI stands for "upstream increasing." It should be noted that

$$\text{UI} \cap I(m) \subset \bigcap_{i=1}^{m} I(i).$$

Indeed, summing the last i inequalities of UI and $I(m)$ gives $I(m - i)$. Another interpretation of UI is that we give priority to serve customers in low-indexed queues. $\text{UI} \cap I$ is exactly the class of functions characterized by the partial sum ordering of Chang *et al.* [19]; see Koole [39, Appendix C].

$w\text{UI}$ can be seen as a weighted version of UI. It is equivalent to

$$\mu(i)f(x + e_j) + (1 - \mu(i))f(x + e_j + e_{i+1})$$
$$\leq \mu(j)f(x + e_i) + (1 - \mu(j))f(x + e_i + e_{+1})$$

for all $1 \leq i < j \leq m$. An interesting special case for $w\text{UI}$ is $f(x) = \sum_{k=1}^{m} c(k)x_k$, i.e., f is linear. Then the following holds:

$$f \in w\text{UI} \Leftrightarrow \mu(1)c(1) \geq \cdots \geq \mu(m)c(m). \tag{6.1}$$

To interpret gUI we go back to the interpretation of UI of giving priority to serving upstream (low-numbered) queues in a tandem system: gUI generalizes this to general networks where feedback to all queues is possible. Queue 0 represents departures (which has nothing to do with the fact that dimension 0 represents the environment).

6.2 Schur Convexity

The inequalities defined in Definition 6.1 have a simple interpretation: less customers is preferable (I) or customers in high-numbered queues are preferable over customers in low-numbered queues (UI and generalizations). The word "preferable" should be interpreted as "having a lower value," as if we try to minimize costs. These conditions do not take the length of the queue into account. Under UI, even if a high-numbered queue has many customers, adding a customer there is still preferable over adding a customer to queue 1. In this section, we study conditions that do take the length into account, that try the balance the queue lengths in some way.

Definition 6.2. We define the following sets of functions:

— $f \in$ SC if

$$f(x + e_i) \leq f(x + e_j)$$

for all x and i, j with $i \neq j$ and $x_i \leq x_j$ and

$$f(x + ke_i) = f(x + ke_j)$$

for all x and i, j with $i \neq j$, $x_i = x_j$, and all $k > 0$;

— $f \in$ ASC if

$$f(x + e_i) \leq f(x + e_j)$$

for all x and i, j with $i < j$ and $x_i \leq x_j$ and

$$f(x + ke_i) \leq f(x + ke_j)$$

for all x and i, j with $i < j$, $x_i = x_j$, and all $k > 0$.

Thus, if $f \in \mathrm{SC}$, then the value of $f(x)$ is lower if x is more balanced. Note also that $f(x)$ does not change if x is permuted: the order of the queues does not matter. In Hordijk and Koole [32], it is shown that SC is actually the class of *Schur convex functions*. $I \cap \mathrm{SC}$ is the class of *weak Schur convex functions*. See Marshall and Olkin [54] for more information on Schur convexity and the corresponding majorization ordering.

ASC is a generalization of SC, where we allow asymmetry in the queues. Now the order does matter. If we require ASC for any possible ordering of the queues, then it is equivalent to SC. There is no simple characterization known for ASC (see Müller [57]).

6.3   Convexity

We define component-wise convexity and related multi-dimensional second-order properties.

Definition 6.3. We define the following sets of functions:

— $f \in Cx(i)$ if

$$2f(x + e_i) \le f(x) + f(x + 2e_i)$$

for all x and $1 \le i \le m$;
— $Cx = Cx(1) \cap \cdots \cap Cx(m)$;
— $f \in Cv(i)$ if

$$f(x) + f(x + 2e_i) \le 2f(x + e_i)$$

for all x and $1 \le i \le m$;
— $Cv = Cv(1) \cap \cdots \cap Cv(m)$;
— $f \in \mathrm{Super}(i, j)$ if

$$f(x + e_i) + f(x + e_j) \le f(x) + f(x + e_i + e_j)$$

for all x and $1 \le i < j \le m$;
— $\mathrm{Super} = \cap_{1 \le i < j \le m} \mathrm{Super}(i, j)$;
— $f \in \mathrm{Sub}(i, j)$ if

$$f(x) + f(x + e_i + e_j) \le f(x + e_i) + f(x + e_j)$$

for all x and $1 \le i < j \le m$;

— Sub $= \cap_{1 \leq i < j \leq m} \mathrm{Sub}(i,j)$;
— $f \in \mathrm{Super}C(i,j)$ if

$$f(x + e_i) + f(x + e_i + e_j) \leq f(x + e_j) + f(x + 2e_i)$$

for all x and $1 \leq i, j \leq m$, $i \neq j$;
— $\mathrm{Super}C = \cap_{1 \leq i,j \leq m : i \neq j} \mathrm{Super}C(i,j)$;
— $f \in \mathrm{Sub}C(i,j)$ if

$$f(x + e_i) + f(x + e_i + e_j) \leq f(x) + f(x + 2e_i + e_j)$$

for all x and $1 \leq i,\ j \leq m$, $i \neq j$;
— $\mathrm{Sub}C = \cap_{1 \leq i,j \leq m : i \neq j} \mathrm{Sub}C(i,j)$;
— $f \in \mathrm{MM}(i,j)$ if

$$f(x) + f(x + d_i + d_j) \leq f(x + d_i) + f(x + d_j)$$

for all x and $0 \leq i < j \leq m$ such that $x + d_i, x + d_j \in \mathbb{N}_0^{m+1}$, with $d_0 = e_1$, $d_k = -e_k + e_{k+1}$, $k = 1, \ldots, m - 1$, and $d_m = -e_m$;
— $\mathrm{MM} = \cap_{1 \leq i < j \leq m} \mathrm{MM}(i,j)$.

Cx and Cv are evidently component-wise convexity and concavity. Next comes super- and submodularity. $\mathrm{Super}C$ and $\mathrm{Sub}C$ stand for super- and subconvexity. No appropriate terms existed; the current names are inspired by (but not exactly the same as) terminology used in Ghoneim and Stidham [25]. Note that all four properties are symmetric in the two dimensions. Combining the right inequalities implies component-wise convexity:

$$\mathrm{Super}(i,j) \cap \mathrm{Super}C(i,j) \subset Cx(i) \qquad (6.2)$$

and

$$\mathrm{Sub}(i,j) \cap \mathrm{Sub}C(i,j) \subset Cx(i). \qquad (6.3)$$

This can easily be seen by adding the inequalities.

MM stands for multimodularity. For $m = 2$ it holds that $\mathrm{MM} = \mathrm{Super} \cap \mathrm{Super}C$ because they consist of exactly the same equations. For higher dimensions MM consists of more equations.

Another property related to convexity is *directional convexity*, defined next.

Definition 6.4. A function f is called directionally convex if for all $x, y, z \in \mathbb{N}_0^{m+1}$, $y_0 = z_0 = 0$,

$$f(x + y) + f(x + z) \leq f(x) + f(x + y + z).$$

The class of directionally convex functions is denoted as DC.

There is a simple alternative characterization of directional convexity:

$$Cx \cap \text{Super} = \text{DC}. \tag{6.4}$$

See Koole [44] for a proof.

Note also that DC is weaker than MM for $m = 2$, using Equation (6.2). This also holds for general m, see Koole [44]. A two-dimensional function which is in DC but not in $\text{Super}C(1,2)$ is $f(x) = x_1 x_2$; thus $\text{Super} \cap \text{Super}C \subsetneq \text{DC}$.

In conclusion:

$$\text{MM} \subset \text{Super} \cap \text{Super}C \subsetneq \text{Super} \cap Cx = \text{DC} \cap Cx. \tag{6.5}$$

6.4   Bibliographical Notes

MM was introduced in Hajek [29] for the study of assignment rules to a queue. Its relation to monotonicity results was recognized in Stidham and Weber [77, on top of page 295], although it was used earlier in Hajek [28] (for $m = 2$) and Weber and Stidham [83] for controlled queueing systems, without making the connection with [29].

Altman *et al.* [1] is a book that is completely dedicated to MM and its many applications.

DC was, according to Müller and Scarsini [58], introduced in Shaked and Shanthikumar [71].

7

Propagation Results

In this chapter we combine the operators with the function sets we just defined. We use the same notation as on Page 11. When we write $T\colon A \to B$ then we mean to say: if $f \in A$, then $Tf \in B$. The ideal situation is that $A = B$, but often we need additional inequalities. In all cases $A \subset B$.

Although we defined concavity, we will not propagate it, we only use it in the definition of some of the operators. We focus on convexity, often in combination with increasingness.

We discuss one by one the different sets of operators.

7.1 Environment Operators

Theorem 7.1. The following relations hold:

— T_{disc} (with C in the left-sided set), $T_{\mathrm{env}}\colon I \to I$, $\mathrm{UI} \to \mathrm{UI}$, $w\mathrm{UI} \to w\mathrm{UI}$, $g\mathrm{UI} \to g\mathrm{UI}$, $\mathrm{SC} \to \mathrm{SC}$, $\mathrm{ASC} \to \mathrm{ASC}$, $Cx \to Cx$, $\mathrm{Super} \to \mathrm{Super}$, $\mathrm{Sub} \to \mathrm{Sub}$, $\mathrm{Super}C \to \mathrm{Super}C$, $\mathrm{Sub} \to \mathrm{Sub}$, $\mathrm{MM} \to \mathrm{MM}$;

— $T_{\min}$: $I \to I$, UI $\to$ UI, wUI $\to w$UI when $\mu(1) \leq \cdots \leq \mu(m)$, SC $\to$ SC, ASC $\to$ ASC;

— $T_{\max}$: $I \to I$, UI $\to$ UI, wUI $\to w$UI when $\mu(1) \geq \cdots \geq \mu(m)$, SC $\to$ SC, ASC $\to$ ASC, $Cx \to Cx$.

Thus T_{disc} and T_{env} are "ideal" in the way we just discussed, i.e., $T : A \to A$ holds for any combination of sets defined in Chapter 6. Note that for T_{disc} we need in all situations that $C \in A$. The operators $T_{\min}$ and $T_{\max}$ are less "ideal," $T : A \to A$ does not hold in cases where the inequality has multiple terms on the left- or right-hand side. For details, see the proof below.

Proof. For all inequalities we defined in Chapter 6 it holds that they are closed under convex combinations. From this the results for T_{disc} and T_{env} follow directly.

$T_{\min}$ and $T_{\max}$ are more complicated. Suppose we have $a_i + b_i \leq c_i$ for certain numbers a_i, b_i, and c_i. Then $\min_i a_i + \min_i b_i \leq \min_i c_i$. Indeed, suppose $j = \arg\min_i c_i$. Then, $\min_i a_i + \min_i b_i \leq a_j + b_j = c_j = \min_i c_i$. Thus $T_{\min}$ propagates when the right-hand side of an inequality consists of one term. By the same argument $T_{\max}$ propagates when the left-hand side of an inequality consists of a single term. This explains $T_{\max}$: $Cx \to Cx$, and the absence of the same result for $T_{\min}$. (Actually, $T_{\min}$: $Cv \to Cv$ holds.) It also explains the conditions for wUI. □

7.2   Arrival Operators

Theorem 7.2. The following relations hold, $1 \leq i, j \leq m$, $i \neq j$:

— $T_{\mathrm{A}(i)}$: $I \to I$, UI $\to$ UI, wUI $\to w$UI, gUI $\to g$UI, $Cx \to Cx$, Super $\to$ Super, Sub $\to$ Sub, Super$C \to$ Super$C$, Sub $\to$ Sub, MM $\to$ MM;

— $T_{\mathrm{CA}(i)}$: $I \to I$, UI $\to$ UI, wUI $\to w$UI when $\mu(1) \leq \cdots \leq \mu(m)$, $Cx(i) \to Cx(i)$,   Super$(i,j) \to$ Super(i,j),   Sub $\to$ Sub, Super$(i,j) \cap$ Super$C(i,j) \to$ Super$C(i,j)$,   Super$(i,j) \cap$ Super

$$C(j,i) \to \mathrm{Super}C(j,i), \qquad \mathrm{Sub}(i,j) \cap \mathrm{Sub}C(i,j) \to \mathrm{Sub}C(i,j),$$
$$\mathrm{Sub}\,(i,j) \cap \mathrm{Sub}C(j,i) \to \mathrm{Sub}C(j,i),\ \mathrm{MM} \to \mathrm{MM} \text{ for } i = 1;$$

— $T_{\mathrm{FS}(i)}$: $I \to I$, $Cx \to Cx$, Super $\to$ Super, Sub $\to$ Sub;

— T_{CAF}: $I \to I$, $\mathrm{Sub}(i,j) \cap \mathrm{Sub}C(i,j) \to \mathrm{Sub}(i,j)$ if $m = 2$, $\mathrm{Sub}C(i,j) \to \mathrm{Sub}C(i,j)$ if $m = 2$;

— T_{R}: $I \to I$, $\mathrm{SC} \to \mathrm{SC}$, $\mathrm{ASC} \to \mathrm{ASC}$, $\mathrm{Super}(i,j) \cap \mathrm{Super} \, C(i,j) \cap \mathrm{Super}C(j,i) \to \mathrm{Super}(i,j)$ if $m = 2$, Super $(i,j) \cap \mathrm{Super}C(i,j) \to \mathrm{Super}C(i,j)$ if $m = 2$.

The operator $T_{\mathrm{A}(i)}$ propagates through almost all inequalities, because these inequalities also hold with x replaced by $x + e_i$. The only exceptions are SC and ASC, because of the additional conditions on x. The results for $T_{\mathrm{CA}(i)}$ are much more limited because of the minimization. Because the minimization has a certain structure (the choice between $f(x)$ and $f(x + e_i)$) the results are stronger than for the general minimization operator $T_{\min}$. For $T_{\mathrm{FS}(i)}$ only limited results hold; note that DC also propagates, because of Equation (6.4). It is interesting to compare this with T_{CAF}. Now Cx does not propagate by itself; we need the stronger condition Sub $\cap$ Sub$C$, from which convexity follows. The last operator, T_{R}, propagates through Schur convex, supermodular, and superconvex functions.

Proof. For $T_{\mathrm{A}(i)}$ all the mentioned inequalities still hold when translated by e_i.

$T_{\mathrm{CA}(i)}$ can be written as $T_{\min}(T_{\mathrm{A}}(i), I)$, with I the identity operator. Thus all results that hold for $T_{\min}$ and for $T_{\mathrm{A}}(i)$ hold for $T_{\mathrm{CA}(i)}$ as well. Additionally, due to the special structure of $T_{\mathrm{CA}(i)}$, certain convexity results hold as well. We prove T_{CA}: $Cx \to Cx$. We need to check the different possibilities for the minimizers in x and $x + 2e_i$, denoted by a_1 and a_2, respectively. The only non-trivial case is $a_1 \neq a_2$. It is readily seen that $2\min\{c + f(x + e_i), c' + f(x + 2e_i)\} \leq c' + f(x + e_i) + c + f(x + 2e_i)$, which corresponds to the case that a_1 corresponds to admission and a_2 to rejection, and $2\min\{c + f(x + e_i), c' + f(x + 2e_i)\} \leq c + f(x) + c' + f(x + 3e_i)$ (by using convexity twice), which corresponds to the remaining case (which in fact cannot occur, as we will see later).

To prove $T_{\mathrm{CA}(i)}$: Super $\rightarrow$ Super, we need to show:

$$T_{\mathrm{CA}(i)}f(x + e_i) + T_{\mathrm{CA}(i)}f(x + e_j)$$
$$\leq T_{\mathrm{CA}(i)}f(x) + T_{\mathrm{CA}(i)}f(x + e_i + e_j).$$

If the optimal actions on the right-hand side are the same, for example, rejection, then the inequality follows easily by induction:

$$T_{\mathrm{CA}(i)}f(x + e_i) + T_{\mathrm{CA}(i)}f(x + e_j)$$
$$\leq c + f(x + e_i) + c + f(x + e_j)$$
$$\leq c + f(x) + c + f(x + e_i + e_j)$$
$$= T_{\mathrm{CA}(i)}f(x) + T_{\mathrm{CA}(i)}f(x + e_i + e_j).$$

When the optimal action are different then we should be more careful. If admission is optimal in x and rejection in $x + e_i + e_j$ then we have:

$$T_{\mathrm{CA}(i)}f(x + e_i) + T_{\mathrm{CA}(i)}f(x + e_j)$$
$$\leq c + f(x + e_i) + c' + f(x + e_i + e_j)$$
$$= T_{\mathrm{CA}(i)}f(x) + T_{\mathrm{CA}(i)}f(x + e_i + e_j).$$

If rejection is optimal in x and admission in $x + e_i + e_j$ then we have, by using induction twice:

$$T_{\mathrm{CA}(i)}f(x + e_i) + T_{\mathrm{CA}(i)}f(x + e_j)$$
$$\leq c' + f(x + 2e_i) + c + f(x + e_j)$$
$$\leq c + f(x) + c' + f(x + 2e_i + e_j)$$
$$= T_{\mathrm{CA}(i)}f(x) + T_{\mathrm{CA}(i)}f(x + e_i + e_j).$$

For proving that Sub propagates we refer to Lemma 3.1 of [6]. Now consider the first inequality of Super$C$. Again, the case that the optimal action for the right-hand side are equal is simple. Assume they are different. If rejection is optimal in $x + e_j$ then it suffices to show

$$f(x + 2e_i) + f(x + e_j) \leq f(x + e_j) + c + f(x + 3e_i).$$

This holds as it is the sum of Super$C$ and Super$C$ with x replaced by $x + e_i$. In case admission is optimal in $x + e_j$ then the actions in the left-hand side can be chosen such that an equality results. Consider

the second inequality of Super$C$, and the case that the optimal actions at the right-hand side are different. If rejection is optimal in x then it suffices to have $Cx(j)$. This explains why we require Super next to Super$C$. If admission is optimal in x then we can choose the actions at the left-hand side such that is the sum of both inequalities from the definition of Super$C$.

For the proof of $T_{\mathrm{CA}(1)} \colon \mathrm{MM} \to \mathrm{MM}$ we refer to [44].

Proving $T_{\mathrm{FS}(i)} \colon I(i) \to I(i)$ means showing:

$$(B_i - x_i)f(x + e_i) + x_i f(x) \le (B_i - x_i - 1)f(x + 2e_i)$$
$$+(x_i + 1)f(x + e_i)$$

for $x_i \le B_i$. This holds indeed for $f \in I(i)$. For $x_i > B_i$ the inequality to show is $B_i f(x) \le B_i f(x + e_i)$, which holds by induction. $T_{\mathrm{FS}(i)} \colon I(j) \to I(j)$ is trivial to show.

$T_{\mathrm{FS}(i)} \colon Cx(i) \to Cx(i)$ is, for $x_i \le B_i$, equivalent to:

$$2(B_i - x_i - 1)f(x + 2e_i) + 2(x_i + 1)f(x + e_i)$$
$$\le (B_i - x_i)f(x + e_i) + x_i f(x) + (B_i - x_i - 2)f(x + 3e_i)$$
$$+(x_i + 2)f(x + 2e_i),$$

which holds trivially for $f \in Cx$. For $x_i > B_i$ the inequality follows directly by induction. $T_{\mathrm{FS}(i)} \colon Cx(j) \to Cx(j)$ is trivial to show. The proofs for Super and Sub are similar.

Consider T_{CAF} and $\mathrm{Sub}(i,j)$. The only interesting cases arise when the optimizing action in the right-hand side of $\mathrm{Sub}(i,j)$ are different. Consider the case where $T_{\mathrm{CAF}}f(x + e_i) + T_{\mathrm{CAF}}f(x + e_j) = c + f(x + 2e_i + e_j) + f(x + e_j)$. Then

$$T_{\mathrm{CAF}}f(x) + T_{\mathrm{CAF}}f(x + e_i + e_j)$$
$$\le c + f(x + e_i + e_j) + f(x + e_i + e_j)$$
$$\le f(x + 2e_i + e_j) + f(x + e_j),$$

the last inequality by Equation (6.3).

We continue with $\mathrm{Sub}C(i,j)$. It is equal to:

$$f(x + e_i + e_j) - f(x) \le f(x + 2e_i + e_j) - f(x + e_i).$$

Note that $x + e_i + e_j - x = x + 2e_i + e_j - (x + e_i) = e_i + e_j$. Thus if rejection is optimal in x, i.e., $f(x + e_i + e_j) - f(x) \geq c$, then it is also optimal to reject in $x + e_i$, because of Sub$C$. Thus there is only one possibility for which the actions on the right-hand side of Sub$C(i,j)$ are different: admission in x, rejection in $x + e_i$. Then

$$
\begin{aligned}
T_{\mathrm{CAF}} f(x + e_i) &+ T_{\mathrm{CAF}} f(x + e_i + e_j) \\
&\leq c + f(x + 2e_i + e_j) + f(x + e_i + e_j) \\
&= T_{\mathrm{CAF}} f(x) + T_{\mathrm{CAF}} f(x + 2e_i + e_j).
\end{aligned}
$$

The property $T_{\mathrm{R}} \colon I \to I$ is easy to prove. For ASC $\to$ ASC we refer to [32]. SC $\to$ SC is a special case of ASC $\to$ ASC. The other results for T_{R} can be found in [40]. □

7.3   Departure Operators

Theorem 7.3. The following relations hold, $1 \leq i, j \leq m$:

— $T_{\mathrm{D}(i)}$: $I \to I$, $I \cap \mathrm{UI} \to \mathrm{UI}$ for $i = m$, $I(i) \cap Cx(i) \to Cx(i)$ if $\mu(x) \in I \cap Cv$, $Cx(j) \to Cx(j)$ for $j \neq i$, Super $\to$ Super, Sub $\to$ Sub;

— $T_{\mathrm{D1}(i)}$: $I \to I$, $I \cap \mathrm{UI} \to \mathrm{UI}$ for $i = m$, $I(i) \cap Cx(i) \to Cx(i)$, $Cx(j) \to Cx(j)$ for $j \neq i$,   Super $\to$ Super,   Sub $\to$ Sub, Super$C(j,k) \to$ Super$C(j,k)$ $(j, k \neq i)$, $I(i) \cap$ Super$C(i,j) \to$ Super$C(i,j)$ $(j \neq i)$, $Cx(j) \cap$ Super$C(j,i) \to$ Super$C(j,i)$, Sub $C(j,k) \to$ Sub$C(j,k)$ $(j, k \neq i)$, $I(i) \cap$ Sub$C(i,j) \to$ Sub$C(i,j)$ $(j \neq i)$, $Cx(j) \cap$ Sub$C(j,i) \to$ Sub$C(j,i)$, $\mathrm{UI} \cap \mathrm{MM} \to \mathrm{MM}$ for $i = m$;

— T_{PD}: $I \to I$, $\mathrm{UI} \to \mathrm{UI}$, $I \cap \mathrm{SC} \to \mathrm{SC}$ for $\mu(i) = \mu(j)$, $I \cap \mathrm{ASC} \to \mathrm{ASC}$ if $\mu(i) \geq \mu(j)$ if $i < j$, $I(i) \cap Cx(i) \to Cx(i)$, $Cx(j) \to Cx(j)$ for $j \neq i$,   Super $\to$ Super,   Sub $\to$ Sub, Super$C(j,k) \to$ Super$C(j,k)$ $(j, k \neq i)$, $I(i) \cap$ Super$C(i,j) \to$ Super$C(i,j)$ $(j \neq i)$, $Cx(j) \cap$ Super$C(j,i) \to$ Super$C(j,i)$, Sub $C(j,k) \to$ Sub$C(j,k)$ $(j, k \neq i)$, $I(i) \cap$ Sub$C(i,j) \to$ Sub$C(i,j)$ $(j \neq i)$, $Cx(j) \cap$ Sub$C(j,i) \to$ Sub$C(j,i)$, $\mathrm{UI} \cap \mathrm{MM} \to \mathrm{MM}$ for $i = m$;

— $T_{\mathrm{CD}(i)}$: $I \to I$ if $c(0) = \min_{\mu \in [0,1]} c(\mu)$, $I \cap \mathrm{UI} \to \mathrm{UI}$ for $i = m$ if $c(0) = \min_{\mu \in [0,1]} c(\mu)$, $Cx(i) \to Cx(i)$, $\mathrm{Super}(i,j) \to \mathrm{Super}(i,j)$, $\mathrm{Sub} \to \mathrm{Sub}$ if $c(0) = \min_{\mu \in [0,1]} c(\mu)$, $Cx(i) \cap \mathrm{Super}C(i,j) \to \mathrm{Super}C(i,j)$, $\mathrm{Super}(i,j) \cap \mathrm{Super}C(j,i) \to \mathrm{Super}C(j,i)$, $Cx(i) \cap \mathrm{Sub}C(i,j) \to \mathrm{Sub}C(i,j)$, $\mathrm{Sub}(i,j) \cap \mathrm{Sub}C(j,i) \to \mathrm{Sub}C(j,i)$, $\mathrm{MM} \to \mathrm{MM}$ for $i = m$;

— T_{MS}: $I \to I$, $w\mathrm{UI} \to w\mathrm{UI}$ for μ as in T_{MS}, $\mathrm{Super} \cap \mathrm{Super}C \to \mathrm{Super}$ for $m = 2$ and $\mu(1) = \mu(2)$, $\mathrm{Super}C \to \mathrm{Super}C$ for $m = 2$ and $\mu(1) = \mu(2)$;

— T_{MMS}: $I \to I$, $I \cap w\mathrm{UI} \to w\mathrm{UI}$ for μ as in T_{MMS} and $\mu(i) \le \mu(j)$, $i < j \in I$.

Note that T_{D1} is a special case of T_{D}. The results of the form $I \cap Cx \cap \mathrm{Super}C \to \mathrm{Super}C$ imply $I \cap \mathrm{Super} \cap \mathrm{Super}C \to \mathrm{Super}C$, but we prefer this more general formulation (because $I \cap \mathrm{Super} \cap \mathrm{Super}C \subset I \cap Cx \cap \mathrm{Super}C$), although in the models of the next sections we use $\mathrm{Super}C$ mostly in combination with Super.

Also for T_{PD} many results follow directly, as T_{PD} can be seen as a combination of T_{unif} and $T_{\mathrm{D1}(i)}$. Only the results for SC and ASC do not follow from results for T_{unif} and $T_{\mathrm{D1}(i)}$, and therefore need a separate proof.

Proof. Consider $T_{\mathrm{D}(i)}$. The increasingness, UI, $Cx(j)$ $(j \ne i)$, $\mathrm{Super}(j,k)$ and $\mathrm{Sub}(j,k)$ $(j,k \ne i)$ follow readily, without further conditions on the form of $\mu(x)$. We prove $Cx(i)$. We write $\bar{\mu}(x) = \bar{\mu} - \mu(x)$. Using that $\mu \in Cv$ and $f \in I$ in the first inequality and $f \in Cx$ in the last inequality we find:

$$\begin{aligned}
2T_{\mathrm{D}}f(x + e_i) &= 2\mu(x_i + 1)f(x) + 2\bar{\mu}(x_i + 1)f(x + e_i) \\
&\le [\mu(x_i) + \mu(x_i + 2)]f(x) \\
&\quad + [\bar{\mu}(x_i) + \bar{\mu}(x_i + 2)]f(x + e_i) \\
&= 2\mu(x_i)f(x) + (\mu(x_i + 2) - \mu(x_i))f(x + e_i) \\
&\quad + 2\bar{\mu}(x_i + 2)f(x + e_i) + (\bar{\mu}(x_i) - \bar{\mu}(x_i + 2))f(x) \\
&\le \mu(x_i)f(x - e_i) + \mu(x_i + 2)f(x + e_i) + \bar{\mu}(x_i)f(x) \\
&\quad + \bar{\mu}(x_i + 2)f(x + 2e_2) \\
&= T_{\mathrm{D}}f(x) + T_{\mathrm{D}}f(x + 2e_i).
\end{aligned}$$

We continue with Super(i,j). We have to differentiate between $\mu(x) \leq \mu(x + e_i)$ and $\mu(x) > \mu(x + e_i)$. The first case follows readily, for the second case we use that

$$
\begin{aligned}
f(x + e_j) - f(x) &\leq f(x + e_i + e_j) - f(x + e_i) \\
&\leq (x + 2e_i + e_j) - f(x + 2e_i).
\end{aligned}
$$

The final inequality to prove is Sub(i,j). Its proof is similar to that of Super(i,j).

For $T_{\mathrm{D1}(i)}$ many results are already covered by T_{D}. The rest follows easily. For the last result concerning MM we refer to [44].

For T_{PD} we only need to show the results for SC and ASC (the others follow from the results for other operators). For the proofs we refer to [32].

For T_{CD} we should, as for T_{CA}, consider the different possibilities for the minimizer at the right-hand side of the inequality.

Finally, for the proofs of I and $w\mathrm{UI}$ for T_{MS} and T_{MMS} we refer to [33]. The proofs for Super and Super$C$ for T_{MS} can be found in [28] or [17].   □

7.4   Transfer Operators

Finally we consider the operators that model customer transfers from one queue to the other.

Theorem 7.4. The following relations hold, $1 \leq i \leq m$:

— $T_{\mathrm{TD1}(i)}$: $I \to I$, UI $\to$ UI for $i < m$, UI $\cap$ MM $\to$ MM for $i < m$, UI $\cap Cx \cap$ Super $\to Cx$ for $i < m$, UI $\cap Cx \cap$ Super $\to$ Super for $i < m$;
— $T_{\mathrm{CTD}(i)}$: $I \to I$ if $c(0) = \min_{\mu \in [0,1]} c(\mu)$, UI $\to$ UI for $i < m$, MM $\to$ MM;
— $T_{\mathrm{TD}(i)}$: $I \to I$, UI $\to$ UI for $i < m$, UI $\cap Cx \cap$ Super $\to Cx$ for $i < m$, UI $\cap Cx \cap$ Super $\to$ Super for $i < m$;
— T_{MTS}: $I \to I$, $g\mathrm{UI} \to g\mathrm{UI}$ for μ as in T_{MTS}.

Proof. For the proofs for $T_{\mathrm{TD1}(i)}$, $T_{\mathrm{CTD}(i)}$, and $T_{\mathrm{TD}(i)}$ we refer to [44]. For the proofs for T_{MTS} we refer to [39, 42].   □

8

Monotonicity of Control

Certain combinations of operators that have some form of control and inequalities give structural results about the optimal policies, seen as functions of the state.

8.1 Arrival Operators

Theorem 8.1. For the minimizer $a(x)$ in $T_{\mathrm{CA}(i)}f(x)$ (action 0 is rejection, 1 is acceptance) holds:

— If $f \in Cx(i)$ then $a(x)$ is decreasing in x_i;
— If $f \in \mathrm{Super}(i,j)$ then $a(x)$ is decreasing in x_j;
— If $f \in \mathrm{Sub}(i,j)$ then $a(x)$ is increasing in x_i;
— If $f \in \mathrm{Super}C(i)$ then $a(x)$ is decreasing in the direction $e_i - e_j$.

If $f \in Cx(i)$ then a policy as in Theorem 8.1 is called a *threshold policy*, with the threshold the level of component i at which the policy switches from 1 to 0. Note that the threshold can be different for all values of x_j, $j \neq i$. If $f \in \mathrm{Super}(i,j)$ then the threshold for admitting a class-i customer is in component j. Let $m > 1$. Let $f \in Cx(i)$,

and consider the thresholds in i as a function of x_j. This is called a *switching curve*. If $f \in Cx \cap \text{Super} = \text{DC}$ then the switching curve is decreasing. If $f \in \text{Super} \cap \text{Super}C$, then the switching curve is even strictly decreasing (until it reaches 0).

Note that the latter result also holds for $f \in \text{MM}$, thanks to Equation (6.5). For $f \in \text{Sub} \cap \text{Sub}C$ the switching curve is increasing.

Proof. $f \in Cx(i)$ means that $f(x + e_i) - f(x)$ is increasing in x_i. As $T_{\text{CA}(i)}f(x) = f(x) + \min\{c, c' + f(x + e_i) - f(x)\}$, this means that if $a(x) = 0$, then $a(x + e_i) = 0$ as well. The other results follow in a similar way. $\qquad\square$

Theorem 8.2. In case $m = 2$, then for the minimizer $a(x)$ in $T_{\text{CAF}}f(x)$ (action 0 is rejection, 1 is acceptance) holds that it is decreasing in x_i if $f \in \text{Sub}C(i, j)$.

The proof of Theorem 8.2 is similar to that of Theorem 8.1

If $f \in \text{Sub}C = \text{Sub}C(i, j) \cap \text{Sub}C(j, i)$, then $a(x)$ is decreasing in x_1 and x_2 and it follows that a decreasing switching curve exists.

8.2   Departure Operators

Theorem 8.3. For the minimizer $\mu^*(x)$ in $T_{\text{CD}(i)}f(x)$ holds:

— If $f \in Cx(i)$ then $\mu^*(x)$ is increasing in x_i;
— If $f \in \text{Super}(i, j)$ then $\mu^*(x)$ is increasing in x_j;
— If $f \in \text{Sub}(i, j)$ then $\mu^*(x)$ is decreasing in x_i;
— If $f \in \text{Super}C(i)$ then $\mu^*(x)$ is increasing in the direction $e_i - e_j$.

The proof is similar to that of Theorem 8.1.

It depends on the form of $c(\mu)$ whose values of μ can be minimizer. If c is linear then only 0 and 1 need to be considered. Then, for each of the results, there is a single threshold. If c is for example piecewise linear and convex, then all discontinuities are possible minimizers, and there is a threshold for each value.

Theorem 8.4. If $f \in w$UI, then

— $T_{\text{MS}} f(x)$ is minimized by $\min_j \{x_j > 0\}$, for $x \neq 0$;
— $T_{\text{MMS}} f(x)$ is minimized by $i_1, \ldots, i_s$ such that $\sum_{n=1}^{i_k - 1} x_n < k \leq \sum_{n=1}^{i_k} x_n$, $1 \leq k \leq s$, for $|x| > s$.

The result follows immediately from the condition wUI. In other words, it says that if $f \in w$UI, then the task (tasks) with the lowest index (indices) should be served.

8.3 Transfer Operators

Theorem 8.5. For the minimizer $\mu^*(x)$ in $T_{\text{CTD}(i)} f(x)$ holds:

— If $f \in \text{Super} C(i, (i+1) \bmod m)$ then $\mu^*(x)$ is increasing in x_i;
— If $f \in \text{Super} C((i+1) \bmod m, i)$ then $\mu^*(x)$ is decreasing in x_{i+1}.

The same remarks as for Theorem 8.3 apply.

Theorem 8.6. If $f \in g$UI, then $T_{\text{MTS}} f(x)$ is minimized by $\min_j \{x_j > 0\}$, for $x \neq 0$.

This result follows immediately from the form of gUI. Again the job with the lowest index should be served.

9

Models

In this chapter we consider different sets of functions, consisting of intersections of the sets defined in Chapter 6. For each set we consider all operators that propagate through this function set. These operators together can be used to build models.

We also give references to the literature, showing which papers are covered by the results.

9.1 First-Order Conditions

In this section, we consider sets of functions that are characterized by one of the properties wUI or gUI. The central operators are those that assign the server(s) to the available tasks. A characteristic result is the optimality of the μc rule.

Theorem 9.1. The following hold, for $1 \leq i \leq m$:

— $T_{A(i)}$, $T_{\mathrm{CA}(i)}$ (with $\mu(1) \leq \cdots \leq \mu(m)$), T_{MS}, T_{disc}, T_{env}, $T_{\min}$ (with $\mu(1) \leq \cdots \leq \mu(m)$), $T_{\max}$ (with $\mu(1) \geq \cdots \geq \mu(m)$): $w\mathrm{UI} \to w\mathrm{UI}$;

— $T_{A(i)}, T_{CA(i)}$ (with $\mu(1) \leq \cdots \leq \mu(m)$), T_{MS}, T_{MMS} (with $\mu(1) \leq$ $\cdots \leq \mu(m)$), T_{disc}, T_{env}, T_{min} (with $\mu(1) \leq \cdots \leq \mu(m)$), T_{max} (with $\mu(1) \geq \cdots \geq \mu(m)$): $I \cap wUI \to I \cap wUI$;
— $T_{A(i)}, T_{MTS}, T_{disc}, T_{env}$: $gUI \to gUI$,

where the μs in the inequalities are equal to the μs in the movable server operators.

The proof is a direct consequence of results in Chapter 7.

From combining operators T_{disc}, $T_{A(i)}$, and T_{MS} with Equation (6.1) and Theorem 8.4 the optimality of the μc rule follows. This was first shown in Buyukkoc *et al.* [18] and Baras *et al.* [12].

In our definition of T_{MS} we did not allow for idleness unless $x = 0$. If we assume that $f \in I$ then it is easy to show that idleness is suboptimal.

Adding T_{MMS} gives the model of Chang *et al.* [19]. The extra condition $\mu(1) \leq \cdots \leq \mu(m)$ makes that the optimal policy starts with the task with the longest expected processing times first. This policy is commonly known as LEPT. An interesting cost function in this context is $\mathbb{I}\{|x| > 0\}$. It can be verified that $\mathbb{I}\{|x| > 0\} \in I \cap wUI$. Thus one of the consequences of Theorem 9.1 is that LEPT minimizes the expected time to empty a system without arrivals. In Hordijk and Koole [33] other cost functions are considered, and also the use of the operator T_{min}.

Now consider the models with feedback using T_{MTS}. Then we find the model of Nain [59, Sec. 3], Koole [38] (extended in [39], see also [42]), and Weishaupt, [84, Sec. 3].

An important application of the results for T_{MTS} is the special case where the direct costs in T_{disc} is equal to $|x|$ or $-|x|$, and $\mu(i,j)$ is only positive for $j = i - 1$ and $i + 1$, and increasing or decreasing in i.

Let all customers arrive in queue 1 (or m), and let the queue index represent the *phase* of the time that a customer spends in the system. Then, using limiting results of phase-type distributions, we can model G/G/1 queues with general service disciplines. In addition to that, if the service time distribution has an increasing (decreasing) hazard rate, then the approximating phase-type distributions are such that the consecutive phases have increasing (decreasing) service rates (Koole [38]).

Two service disciplines are of interest to us: the LAST (the policy that serves the customer(s) with the least attained service time(s)) and the MAST (the policy that serves the customer(s) with the most attained service time(s)). The following corollary is a consequence of Theorem 9.1 and the phase-type results of [38].

Corollary 9.1. The average number of customers in a G/G/1 queue is:

— minimized (maximized) by LAST in case of service times with a decreasing (increasing) hazard rate;
— minimized (maximized) by MAST in case of service times an increasing (decreasing) hazard rate.

As an example, take $\mu(i, i - 1) > 0$, increasing in i and arrivals in queue m. This way we model services with a decreasing hazard rate (DHR). For cost function $-|x|$ it follows from Theorem 9.1 that it is optimal to serve the lowest numbered task available, which is the one with the most service phases yet. It follows thus that MAST minimizes $-|x|$ and therefore maximizes $|x|$ in the case of DHR service times.

The G/G/1 queue with LAST discipline is also called Foreground–background queue; see Nuyens and Wierman [62] for an overview of results on this queue.

The results of Corollary 9.1 can also be found in Righter and Shanthikumar [67].

We summarize this section by giving a list of function sets and the references that deal with these sets using dynamic programming:

- wUI: [12], [18], [33], [19];
- gUI: [38], [42], [59], [84].

An interesting generalization of the models of this section (especially the ones without feedback) is to a finite source of m customers each having their own parameters. Thus $x_i \in \{0, 1\}$ denotes whether customer i is present or not. For this new operator the results for $I \cap w$UI propagates if $\lambda(1) \leq \cdots \leq \lambda(m)$, with $\lambda(i)$ the arrival rate of customer i if $x_i = 0$. For more details see Koole and Vrijenhoek [47] and Koole [42].

9.2   Schur Convexity

In this section we consider sets of functions that are characterized by Schur convexity. The central operator is the routing operator T_R. A characteristic result is the optimality of the *join-the-shortest-queue* policy, a result first obtained, using a completely different method, by Winston [85].

Theorem 9.2. The following hold, for $1 \le i,j \le m$:

— T_disc, T_env, T_min, T_max, T_R, T_PD with $\mu(i) = \mu(j)$: $I \cap \mathrm{SC} \to I \cap \mathrm{SC}$;

— T_disc, T_env, T_min, T_max, T_R, T_PD with $\mu(i) \ge \mu(j)$ if $i < j$: $I \cap \mathrm{ASC} \to I \cap \mathrm{ASC}$.

The proof follows directly from the results in Chapter 7.

The first result of Theorem 9.2 shows that routing to the shortest queue is optimal in the case of symmetric servers. The second concerns asymmetric servers and shows that, while choosing between two servers, it is better to assign to the faster one if there are also less customers at that queue. No results are given in case the faster server has more waiting customers; it only partially characterizes the optimal policy. Note also that the first result is a special case of the second, by reordering the queues.

Several generalizations of these results are possible, including finite buffers and dedicated arrivals to the queues, as long as the arrival rates are equal (in the SC case) or lower for queues with lower indices (in the ASC case). Finite buffers can be included by adding high costs for every customer that exceeds the buffer constraint, of the form $K \sum_{k=1}^{m} (x_k - B)^+$, which B the size of the buffers. Note that this function is SC. Of course some form of admission control should be used to avoid that the number of customers might exceed mB. This can be modeled using $T\mathrm{min}$, for example by including $T\mathrm{min}(I, T_\mathrm{R})$ in the value function. For $B = 1$ the result for ASC reduces to sending to the fastest available server. T_R can also be extended to include finite buffers of different sizes. For more details, see Hordijk and Koole [32].

Further generalizations include multi-server queues, server rates depending on other queue lengths, etc. See Johri [35], Menich and Serfozo [55], and Koole [39] for more details.

A list of function sets and papers, all using dynamic programming, is as follows:

- $I \cap \text{SC}$: [31, 32], [39], [55];
- $I \cap \text{ASC}$: [32], [39].

Generalizing the optimality of shortest-queue routing to non-exponential service times appeared to be quite complicated. See Koole *et al.* [46] and references therein.

An interesting related result using SC but for a continuous state space is the one in which we consider the *workload* in parallel queues. We formulate the complete value function. With u_n we denote the nth interarrival time, numbered backwards, and with F the distribution function of the service time. Then the DP equation for the workloads reads:

$$V_{n+1}(w) = \min_{1 \leq j \leq m} \left\{ \int_0^\infty V_n((w + se_j - u_n e)^+) dF(s) \right\}.$$

Similar equations as for the case with queue lengths hold for the workload case:

$$\int V_n(w + se_i) dF(s) \leq \int V_n(w + se_j) dF(s) \quad \text{for } w_i \leq w_j;$$

$$V_n(w) = V_n(w') \quad \text{for } w' \text{ a permutation of } w; \quad (9.1)$$

$$V_n(w) \leq V_n(w + se_1) \quad \text{for } s \geq 0.$$

Equation (9.1) shows that routing to the queue with the shortest workload is optimal. This equation should not be confused with $V_n(w + se_i) \leq V_n(w + se_j)$ for some or all s. Having the integral inside the minimization means that the routing decision is made *not knowing the service time of the arriving customer*, which is crucial to obtain the optimality result. The proof is straightforward (and can be found in [39]).

9.3 Convexity in One Dimension

In this section we deal with convexity results for $m = 1$.

Theorem 9.3. The following hold, for $m = 1$:

— T_{disc}, T_{env}, $T_{\max}$, $T_{\text{A}(1)}$, $T_{\text{CA}(1)}$, $T_{\text{FS}(1)}$, $T_{\text{CD}(1)}$: $Cx \to Cx$;
— T_{disc}, T_{env}, $T_{\max}$, $T_{\text{A}(1)}$, $T_{\text{CA}(1)}$, $T_{\text{FS}(1)}$, $T_{\text{D}(1)}$ with $\mu(x) \in I \cap Cv$, $T_{\text{D}1(1)}$, $T_{\text{CD}(1)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap Cx \to I \cap Cx$.

This result is a direct consequence of Theorems 7.2 and 7.3. For $T_{\text{CA}(1)}$ and $T_{\text{CD}(1)}$ the optimality of threshold policies follow from Theorem 8.1.

Note the differences between the two expressions in Theorem 9.3. If we require that the value function, and therefore also the direct costs, are convex *and* increasing then we can allow for uncontrolled departures. One-dimensional models with convex cost functions have been studied extensively, mainly to obtain the optimality of threshold policies. The results go back to Lippman [51]. See also Stidham [73] and Ramjee *et al.* [66]. All these results fall within the current framework.

There is no intuitive reason that convexity is necessary to obtain the optimality of threshold policies, but for the current line of proof convexity is necessary. In Stidham and Weber [76] another method is used to obtain threshold optimality, requiring only increasing direct costs.

Our results obtained so far are only valid if the direct costs $C \in Cx$ or $C \in I \cap Cx$. Often used cost functions are x and $(x - s)^+$, the latter in the case that T_{D} models an s-server system. These functions are both increasing and convex. Through Little's law that is equivalent to minimizing the average system or waiting time.

An example of a non-increasing cost function is $C(x) = -I\{x_1 > 0\}r + cx$, where r the reward rate for service to customers and c is the holding cost rate. This is the second model of Lippman [51].

An important question is whether the value function is also convex when the service rates in the operator T_{D} are not concave. This is relevant to call centers, where waiting customers abandon, usually at a higher rate than the service rate. This leads to a service rate of $\mu(x) = \min\{s, x\}\mu + (x - s)^+\gamma$, with γ the abandonment rate. This function is

concave iff $\mu \geq \gamma$. When $\gamma > \mu$ then it is possible to construct numerical counterexamples for the convexity using a value function of the form $T_{\text{costs}} T_{\text{unif}}(T_{\text{A}(1)}, T_{\text{D}(1)})$.

We give a list of function sets and papers in which the models of this section are used. Note that when costs are replaced by rewards then Cx becomes Cv:

- Cx: [40], [51];
- $I \cap Cx$: [40], [73, Sec. 1], [74, Sec. III.B].

9.4 Convexity in Multiple Dimensions

In this section we deal with convexity results that hold for arbitrary m.

Theorem 9.4. The following relations hold, for $1 \leq i \leq m$:

— T_{disc}, T_{env}, $T_{\text{A}(i)}$, $T_{\text{CA}(i)}$, $T_{\text{CD}(m)}$, $T_{\text{CTD}(i)}$: MM $\to$ MM;

— T_{disc}, T_{env}, $T_{\text{A}(i)}$, $T_{\text{CA}(i)}$, $T_{\text{D1}(m)}$, $T_{\text{CD}(m)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$, $T_{\text{TD1}(i)}$ for $i < m$: $I \cap \text{UI} \cap \text{MM} \to I \cap \text{UI} \cap$ MM;

— T_{disc}, T_{env}, $T_{\text{A}(i)}$, $T_{\text{D}(m)}$ with $\mu(x) \in I \cap Cv$, $T_{\text{TD1}(i)}$ for $i < m$, $T_{\text{TD}(i)}$ for $i < m$: $I \cap \text{UI} \cap Cx \cap \text{Super} \to I \cap \text{UI} \cap Cx \cap$ Super;

— T_{disc}, T_{env}, $T_{\text{A}(i)}$, $T_{\text{CA}(i)}$, $T_{\text{FS}(i)}$, $T_{\text{D}(i)}$, $T_{\text{D1}(i)}$, $T_{\text{CD}(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: Sub $\to$ Sub.

Observe that $T_{\text{TD1}(i)}$ and $T_{\text{D1}(m)}$ are special cases of $T_{\text{TD}(i)}$ and $T_{\text{D}(m)}$. Thus controlled operators are possible on the strong condition (see Equation (6.5)) that the costs are multimodular. If we add UI then also uncontrolled single-server queues can be added to the model. Uncontrolled multi-server queues can be studied using the weaker property of directional convexity (which is equal to $Cx \cap \text{Super}$) combined with UI. Note that we have no results for $T_{\min}$ and $T_{\max}$.

The results for MM can be found in Weber and Stidham [83]. The results involving UI can be found in Koole [44].

The models propagating Sub at first sight do not seem particularly interesting: all allowable operators only depend on a single dimension,

and therefore there seems to be no correlation between the queues, other then through the control. However, by interpreting the state as counting events rather than being the underlying state this model can lead to interesting results for relevant models, using only $T_{\mathrm{CA}(i)}$. The underlying state is a function of the event counts, the *scores*.

As an example, consider a model with arrivals in k queues at the same time and k single-server queues. The objective is to minimize the maximum queue length, arrivals and departures are controlled. We take $m = k + 1$, dimension i counts departures from queue i, $1 \leq i \leq k$, and dimension $k + 1$ counts the arrivals. The underlying state is $(x_{k+1} - x_1, \ldots, x_{k+1} - x_k)$. We take as direct costs $x_{k+1} - \min_{1 \leq i \leq k} x_i$ plus a component to avoid negative queue lengths, $\sum_{i=1}^{k} (x_i - x_{k+1})^+ K$, with $K \gg 0$. Note that both are in Sub. This result shows that the optimal action of any of the events is increasing in any other event (and that an event is decreasing in its own event).

Another example is a system with controlled arrivals to k queues and a controlled departure operator that takes an item from each of the queues. Now the underlying state is $(x_1 - x_{k+1}, \ldots, x_k - x_{k+1})$. To avoid negative state components we now take cost component $\sum_{i=1}^{k} (x_{k+1} - x_i)^+ K$, which is again Sub. This model has been studied in Benjaafar and ElHafsi [14] (they work in the underlying state space). The interpretation is that each queue contains inventory of parts of a certain type, and satisfying demand requires an item of each type.

The score state approach was introduced in Glasserman and Yao [26, 27], see also Altman and Koole [6].

Weber and Stidham [83] and Veatch and Wein [81] work in the state space, but they consider submodularity in the transitions (called *transition-monotonicity*). For the controlled tandem model of [83], Sub in the direction of the transitions is equivalent to MM in the state space. A three-dimensional model that combines aspects of a tandem system and arrivals in multiple queues is Miyoshi *et al.* [56].

Although elegant, proving transition-monotonicity (with or without working in the score space) strongly limits the number of models that can be dealt with. The results can always obtained by working directly with the states; the reverse is not true. But the conditions in the state space can be quite complicated. In conclusion: when transition-monotonicity holds for a particular model then it is an

elegant way to derive monotonicity results. In other cases one should work directly in the state space.

We conclude the section with a list of function sets and papers in which they are used:

- MM: [44], [83];
- $I \cap UI \cap MM$: [44];
- $I \cap UI \cap DC$: [44] (in which DC is called vector convexity);
- Sub (in the transitions/scores): [6], [14], [26, 27], [56], [81], [83].

9.5 Convexity in Two Dimensions

The results of Section 9.4 hold also for $m = 2$. However, additional results can be obtained for two dimensions. Remember that $MM = \text{Super} \cap \text{Super}C$ for $m = 2$, and that $DC = Cx \cap \text{Super}$. We focus on results that include I. The main consequences of not having I is that uncontrolled departures or transfers are not allowed, but direct costs need not necessarily be increasing. In [40], both function sets with and without I are studied. We neither consider UI.

Theorem 9.5. The following relations hold, for $1 \leq i \leq 2$:

- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{FS(i)}, T_{D(i)}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap \text{Super} \rightarrow I \cap \text{Super}$;
- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{FS(i)}, T_{D(i)}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap Cx \cap \text{Super} \rightarrow I \cap Cx \cap \text{Super}$;
- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{R}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$, $T_{CTD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap \text{Super} \cap \text{Super}C \rightarrow I \cap \text{Super} \cap \text{Super}C$;
- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{FS(i)}, T_{CAF}, T_{D(i)}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap \text{Sub} \rightarrow I \cap \text{Sub}$;
- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{FS(i)}, T_{D(i)}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap Cx \cap \text{Sub} \rightarrow I \cap Cx \cap \text{Sub}$;
- $T_{\text{disc}}, T_{\text{env}}, T_{A(i)}, T_{CA(i)}, T_{CAF}, T_{D1(i)}, T_{CD(i)}$ with $c(0) = \min_{\mu \in [0,1]} c(\mu)$: $I \cap \text{Sub} \cap \text{Sub}C \rightarrow I \cap \text{Sub} \cap \text{Sub}C$.

The most interesting set of models is the one with function set $I \cap$ Super $\cap$ Super$C$. The models include the routing model of Hajek [28] (essentially consisting of T_R and $T_{D1(i)}$) and the server assignment model of Lin and Kumar [50] (consisting of $T_{A(1)}$, $T_{CTD(1)}$ and $T_{D1(i)}$ for $i = 1, 2$).

Many researchers have attempted to generalize [28] and [50] to more than two servers in total, either by taking more than two single-server queues or by taking multi-server queues. In de Véricourt and Zhou [22] the errors in the papers who claimed to do this are shown. That an approach using some of our inequalities does not work can be seen as follows. We start the work of Hajek [28], in which characterizing operators are T_R and $T_{D1(i)}$. According to Theorem 7.2 we need Super $\cap$ Super$C$ and $m = 2$ to obtain convexity results for T_R. However, as can be seen in Theorem 7.3, we were not able to propagate $T_{D(i)}$ for Super$C$. This explains why we cannot combine T_R and $T_{D(i)}$ in our model. The situation for Lin and Kumar [50] is similar but a little more complicated. Because $T_{D1(1)}$ is one of the operators, we cannot use UI. Thus taking $I \cap UI \cap DC$ as in Theorem 9.4 is not an option, thus we have to rely on one of the cases of Theorem 9.5. However, neither of these allow for $T_{CTD(1)}$ and $T_{D(i)}$ at the same time.

A third paper using $I \cap$ Super $\cap$ Super$C$ is Weber and Stidham [83]. They consider a cycle of queues described by $T_{A(1)}$, $T_{A(2)}$, $T_{CTD(1)}$, and $T_{CTD(1)}$. They also consider a tandem system, replacing $T_{CTD(2)}$ by $T_{D1(2)}$.

For models with multi-server queues we have only a limited number of operators to choose from. The trunk reservation model of Altman *et al.* [2] and Örmeci *et al.* [64] contains only $T_{CA(i)}$ and $T_{D(i)}$ with direct costs $(b_1 x_1 + b_2 x_2 - B)^+ K$ to model a buffer that limits the overall number of jobs. According to Theorem 9.5 $I \cap Cx \cap$ Super is the appropriate model (extending the results of [2]), showing the existence of decreasing switching curves for each type of calls (which was already obtained in Ross and Tsang [68], using a different method, and for a restricted set of policies). Ni *et al.* [61] addresses the interesting problem of determining efficiently the switching curves.

Next is a list with function sets discussed in this section and papers in which they are used:

- $I \cap Cx \cap \text{Super}$: [2], [63], [64];
- $I(1) \cap \text{Super}C(1,2)$: [49];
- $\text{Super} \cap \text{Super}C$: [72] (on $\mathbb{Z}^2$), [17];
- $I \cap \text{Super} \cap \text{Super}C$: [1, Sec. 10.6.2], [25], [28], [30], [40], [50], [83].

Suk and Cassandras [78] study a two-dimensional model with $T_{\text{A}(i)}$ and T_{MS} and finite buffers. This is equivalent to $T_{\text{CA}(i)}$ and T_{MS} and non-linear costs C. As $C \notin w\text{UI}$ Theorem 9.1 cannot be used. [78, Lemma 2] propagates a functional property that is equal to $\text{Super}C$ if $\mu(1) = \mu(2)$ (actually, Super is needed as well, but this is not mentioned in the paper).

10

Other State Spaces

In this overview we focused on the state space $\mathbb{N}_0^m$, although we could introduce finite buffers in certain models by adding functions of the form $\sum_i (x_i - B_i)^+ K$ or $(\sum_i x_i - B)^+ K$ to the direct costs. Then any controlled operator in which costs are minimized will avoid going to states with $x_i > B_i$ or $\sum_i x_i > B$ as long as K is big enough.

In this chapter we discuss briefly models with other state spaces.

Our first model has state space $\{0,1\}^m$. We use operator T_{PD} and a new controlled routing operator of the form $\min\{b + f(x), \min_{j:x_j=0}\{f(x + e_j)\}\}$. Thus it is possible to reject arrivals, at cost b; otherwise we have to assign the arrival to one of the empty queues. If we assume that $\mu(1) \geq \cdots \geq \mu(m)$ then $I \cap \mathrm{ASC} \cap \mathrm{Super}$ propagates, now defined on the new state space $\{0,1\}^m$. From ASC it follows that if assignment is optimal then the customer should be assigned to the fastest available server. Super shows that assignment is less likely to be optimal if more servers are occupied; this is relevant if there are multiple arrival operators having different rejection costs. This result was first derived in [39], and is a generalization of results in Derman *et al.* [23].

Another interesting model with state space $\{0,1\}^m$ and function set $I \cap w$UI has been discussed at the end of Section 9.1.

In some situations it is also useful to have negative queue lengths, for example in situations where the components model inventory of parts and where backorders are possible. When $x_i > 0$, item i is on stock; $x_i < 0$ denotes the number of backorders. In the papers dealing with these models threshold policies are often called *hedging* policies, terminology introduced in Kimemia and Gershwin [36], in the context of a flexible manufacturing system with unreliable machines. An example of a paper containing a monotonicity result is Song and Sun [72], in which Super $\cap$ Super$C$ is propagated on $\mathbb{Z}^2$ and direct convex costs taking its minimum at 0. Other examples are Benjaafar and ElHafsi [14, Section 3.3] (see Section 9.4) and Benjaafar *et al.* [13].

An interesting extension of Markov decision processes is to models where the state is not observed. In this situation actions cannot be based on the actual state, but on the information that is available on the state, the so-called *information state*. In general these are distributions on the underlying states. However, special cases exist for which the information states have some nice structure. Several monotonicity results for these models have been obtained in the literature. We mention some of these papers, without going into details.

One example is Kuri and Kumar [48] which considers the symmetric routing model of Section 9.2, which was characterized by SC, but with a delay in the queue length information. They show that for a delay of 1 routing to the shortest queue is still optimal; for delay 2 or longer counterexamples exist. Both Altman and Nain [7] and Altman and Koole [5] consider a queue with some form of admission control and delayed information. Beutler and Teneketzis [15] generalize some models characterized by $I \cap$ Super $\cap$ Super$C$ to partial information. Artiges [10] extends the routing model of Hajek [28] to a delay of 1. A general framework for partially ordered state spaces and two actions is given in Altman and Stidham [8].

Koole [41] considers a piece of equipment (e.g., a fire detector) which has an exponentially distributed lifetime. As long as we do not check its condition our knowledge about it is summarized by the probability it is up (or, equivalently, its lifetime). This probability is the state of

the process, thus $\mathcal{X} = [0,1]$. An action might be to check whether it is functioning and replace it when necessary. After the check, the state is 1. It is shown in [41] that below a certain threshold probability we replace the piece of equipment.

11

Comparing Systems

Structural properties of value functions can be used to compare different systems. In this section we focus on systems that differ in one or more operators, but for which the form of the value function is the same. Typical examples are comparing systems with batch arrivals versus arrivals one at a time, and comparing systems which differ in one or more parameters (such as the arrival or departure rate). We will see that for comparing systems or to prove montonicity in a parameter we often need convexity in the state.

We will focus on systems in which only dimension 1 is treated differently. For this reason we omit the subscript "(1)" from most of the notation. e.g., we write T_A instead of $T_{A(1)}$.

11.1 Comparing Different Arrival Processes

Next to the value function V_n, defined by the operators T_i, we now have a second value function V_n' that differs from V_n only in the fact that T_i is replaced everywhere by T_i', for all i. We would like to show $V_n(x) \leq V_n'(x)$ for all x and n, for well-chosen T_i and T_i'. To do this it suffices to prove, for functions f and f', that $T_i f \leq T_i'(f')$ follows from

$f \leq f'$. To show this, we usually need an extra condition of the form f or $f' \in \mathcal{F}$, with $\mathcal{F}$ typically Cx or $I \cap Cx$.

The prime application of this idea is to compare the regular arrival operator $T_A f(x) = f(x + e_1)$ with a batch arrival operator $T'_A f(x) = \sum_{k=0}^{\infty} p_k f(x + k e_1)$, with $\sum_{k=0}^{\infty} p_k = 1$. Define $\bar{p} = \sum_{k=0}^{\infty} k p_k$.

The basis of the comparison results is the following lemma. Note that all comparisons of vectors and functions are componentwise.

Lemma 11.1. For functions f and f' it holds that:

— $f \leq f' \Rightarrow Tf \leq Tf'$ for $T \in \{T_{\text{env}}, T_{\text{max}}, T_{\text{min}}, T_A, T_{\text{FS}}, T_{\text{CA}}, T_{\text{disc}}, T_D, T_{\text{CD}}\}$;

— $f \leq f'$, $f \in Cx$, $\bar{p} = 1 \Rightarrow T_A f \leq T'_A f'$.

Proof. The first statement is trivial. The second follows from Jensen's inequality and the first statement:

$$T_A f(x) \leq T'_A f(x) \leq T'_A f'(x). \qquad \square$$

In case $\bar{p} > 1$, then the proof continues under the additional condition that f or $f' \in I$.

A simple direct consequence of the first statement of Lemma 11.1 is in comparison with time, i.e., in the subscript n of V_n.

Lemma 11.2. Let V_n be a value function defined using operators for which Lemma 11.1 holds. Then

$$V_0 \leq V_1 \Rightarrow V_n \leq V_{n+1}.$$

Often we take $V_0 = 0$. Then $V_0 \leq V_1$ is equivalent to requiring non-negative direct costs.

An important implication of Lemma 11.1 is that it allows us to compare queueing models with their fluid limits. Indeed, if we interpret x as a measure of the load in the system, and a batch as the workload of an arrival in the original queueing system, then the results

compare approximations of the fluid limit and the original queue. For more details, see Altman *et al.* [3].

The intuition behind Lemma 11.1 is that systems behave "better" when the arrival streams are less peaked. Another situation where this occurs is when there are multiple peaked streams (i.e., on–off processes): the superposition of processes smoothes out the peaks. To study this, consider two value functions that differ in one operator that models two uniformized on–off processes (there are possibly other operators modeling other on–off processes). Abusing notation a little, we assume that the environment variable takes values in $\{0,1\}^2$ (of course, this can simply be mapped into a one-dimensional variable), representing the states of both on–off processes. These is one operator $T_{0/1T}$ representing the transitions of the two on–off processes, possibly resulting from uniformization. For simplicity we assume both on–off processes have the same parameters: with probability γ_1 (γ_0) each process goes to (stays in) state 1 (0), and thus $2(\gamma_0 + \gamma_1) = 1$:

$$
\begin{aligned}
T_{0/1T}f((a,b),i) \;=\; & \gamma_0 f((0,b),i) + \gamma_0 f((a,0),i) \\
& + \gamma_1 f((1,b),i) + \gamma_1 f((a,1),i).
\end{aligned}
$$

Define another operator $T_{0/1A}$ modeling the arrivals generated by the on–off processes: λ_k is the probability that an arrival is generated by on–off process k, if it is in state 1. With $\lambda_1 + \lambda_2 = 1$ this leads to

$$
\begin{aligned}
T_{0/1\mathrm{A}}f((a,b),i) \;=\; & \lambda_1 f((a,b),i + \mathrm{I\!I}\{a = 1\}) \\
& + \lambda_2 f((a,b),i + \mathrm{I\!I}\{b = 1\}).
\end{aligned}
$$

We assume that $\lambda_1 \leq \lambda_2$.

Consider also a second operator V'_n, which differs from V_n only in the arrival probabilities λ'_i: $\lambda'_1 < \lambda_1$. Because also $\lambda'_1 + \lambda'_2 = 1$, $\lambda'_2 > \lambda_2$. Intuitively speaking, the V_n process is more balanced than the V'_n process, because the arrival probabilities are closer.

Let $\pi(0)$ ($\pi(1)$) be the stationary probability that any of the on–off processes is in state 0 (1). We define the ordering $\prec_{0/1}$ as follows:

$f \prec_{0/1} f'$ if

$$
\begin{aligned}
\pi(0)^2 &f((0,0),i) + \pi(1)^2 f((1,1),i+j+k+l) \\
&+ \pi(0)\pi(1)[f((1,0),i+j) + f((0,1),i+j+k)] \\
\leq \pi(0)^2 &f'((0,0),i) + \pi(1)^2 f'((1,1),i+j+k+l) \\
&+ \pi(0)\pi(1)[f'((1,0),i+j) + f'((0,1),i+j+k)]
\end{aligned}
$$

for all $i,j,k,l \in \mathbb{N}_0$.

Lemma 11.3. For functions f and f' it holds that:

— $f \prec_{0/1} f'$, $f \in Cx \Rightarrow Tf \prec_{0/1} T'f'$ for $T \in \{T_{0/1T}, T_{0/1A}\}$;
— $f \prec_{0/1} f' \Rightarrow Tf \prec_{0/1} T'f'$ for $T \in \{T_{\mathrm{disc}}, T_{\mathrm{env}}, T_{\mathrm{A}}, T_{\mathrm{D}}, T_{\mathrm{CD}}\}$.

Proof. The proof of the first expression is similar to the proof of Lemma 4.3 of [45]. The second expression is trivial to show. □

Lemma 11.3 can be used as follows. By taking $j = k = l = 0$ in the ordering, we find $f((X,Y),i) \leq f((X,Y),i)$ for X and Y, the states of on–off process 1 and 2, stationary.

Consider now two systems with K on–off processes, with arrival probabilities the vectors λ and λ' with $\sum \lambda(k) = \sum \lambda'(k) = 1$. Let λ and λ' differ in only two entries, with λ more "balanced" than λ'. Now Lemma 11.3 can be applied. By repeating this we find an ordering between pairs of λ and λ': $\lambda \prec \lambda'$ if we can construct one from the other by repeatedly changing two entries. This ordering is exactly the Schur convex ordering of Marshall and Olkin [54], see Section 6.2. This leads to the following:

Corollary 11.1. Consider two value functions V_n and V_n' consisting of $T_{\mathrm{disc}}, T_{\mathrm{env}}, T_{0/1T}, T_{0/1A}, T_{\mathrm{A}}, T_{\mathrm{D}}$ with $\lambda \prec \lambda'$ and direct costs $C \in I \cap Cx$ then $V_n(i) \prec_{0/1} V_n'(i)$.

Note that we need $C \in I$ as T_{D} is part of the value function. This corollary is a generalization of results in Koole and Liu [45].

11.2 Monotonicity in Parameters

Increasingness in a parameter value is often easy to show. Take an M/M/1 queue as example. That the average costs are increasing in the arrival rate can simply be shown by induction, under the condition that the direct costs are increasing. Showing convexity is more challenging.

In what follows we define the different operators that we consider and we prove first the increasingness of the value function. This will be an excellent preparation of the main objective of this paragraph: proving convexity of the value function in the parameter value(s).

We will compare systems with different arrival rates and different number of servers. We extend T_A and T_D as follows as to make this possible. Let S be the upper bound of the number of servers.

Definition 11.1. We define:

— $T_{A(\lambda)} = \lambda f(x + e_1) + (1 - \lambda)f(x)$ for $0 \le \lambda \le 1$;
— $T_{D(\mu)} = \mu f((x - e_1)^+) + (1 - \mu)f(x)$ for $0 \le \mu \le 1$;
— $T_{D(s)} = [\min\{x_1, s\}f(x - e_1) + (S - \min\{x_1, s\})f(x)]/S$.

The equivalent of Lemma 11.1 for the operators just defined is as follows.

Lemma 11.4. For functions f and f' it holds that:

— $f \le f'$, f or $f' \in I \Rightarrow T_{A(\lambda)}f \le T_{A(\lambda + \delta)}f'$;
— $f \le f'$, f or $f' \in I \Rightarrow T_{D(\mu + \delta)}f \le T_{D(\mu)}f'$;
— $f \le f'$, f or $f' \in I \Rightarrow T_{D(s+1)}f \le T_{D(s)}f'$.

Proof. Using $f \le f'$ and then $f' \in I$ we find:

$$
\begin{aligned}
T_{A(\lambda)}f(x) &= \lambda f(x + e_1) + (1 - \lambda)f(x) \\
&\le \lambda f'(x + e_1) + (1 - \lambda)f'(x) \\
&\le (\lambda + \delta)f'(x + e_1) + (1 - \lambda - \delta)f'(x) \\
&= T_{A(\lambda + \delta)}f'(x).
\end{aligned}
$$

By starting with $\lambda f(x + e_1) + (1 - \lambda)f(x) \le (\lambda + \delta)f(x + e_1) + (1 - \lambda - \delta)f(x)$ and then using $f \le f'$ we need $f' \in I$ instead of $f' \in I$. The other two statement are hardly more complicated to prove. $\square$

Next we move to convexity. We cannot prove it directly, we need another condition. For this, we introduce the following notation: $\Delta_1 f(x) = f(x + e_1) - f(x)$. In what follows we will regularly use conditions of the form $\Delta_1 f \le \Delta_1 f'$ which are to be considered componentwise. Thus $\Delta_1 f \le \Delta_1 f'$ is equivalent to

$$f'(x) + f(x + e_1) \le f(x) + f'(x + e_1)$$

for all x. Note the equivalence with supermodularity as it is introduced in Section 6.3: when we take $f'(x) = f(x + e_2)$ then both definitions are equivalent.

The following lemma is crucial in proving convexity.

Lemma 11.5. For functions f and f' it holds that:

- $\Delta_1 f \le \Delta_1 f' \Rightarrow \Delta_1 T f \le \Delta_1 T f'$ for $T \in \{T_{\mathrm{disc}}, T_{\mathrm{env}}, T_{\mathrm{A}}, T_{\mathrm{FS}}, T_{\mathrm{D}}\}$;
- $\Delta_1 f \le \Delta_1 f'$, f or $f' \in Cx \Rightarrow \Delta_1 T_{\mathrm{A}(\lambda)} f \le \Delta_1 T_{\mathrm{A}(\lambda + \delta)} f'$;
- $\Delta_1 f \le \Delta_1 f'$, f or $f' \in Cx$, $f' \in I \Rightarrow \Delta_1 T_{\mathrm{D}(\mu + \delta)} f \le \Delta_1 T_{\mathrm{D}(\mu)} f'$;
- $\Delta_1 f \le \Delta_1 f'$, f or $f' \in Cx$, $f' \in I \Rightarrow \Delta_1 T_{\mathrm{D}(s+1)} f \le \Delta_1 T_{\mathrm{D}(s)} f'$.

Proof. The first statement follows directly because the operators in $\{T_{\mathrm{disc}}, T_{\mathrm{env}}, T_{\mathrm{A}}, T_{\mathrm{FS}}, T_{\mathrm{D}}\}$ all have the same transitions, independent of the values of the function to which they are applied (which is for example not the case with T_{max}: the actual transition depends on the values in the states that are reachable).

The second statement, written out, is equivalent to a convex combination of $\Delta_1 f(x) \le \Delta_1 f'(x)$, $\Delta_1 f(x + e_1) \le \Delta_1 f'(x + e_1)$ and

$$f'(x + e_1) + f(x + e_1) \le f(x) + f'(x + 2e_1), \qquad (11.1)$$

which is the sum of the following two inequalities:

$$2f'(x + e_1) \le f'(x) + f'(x + 2e_1)$$

and

$$f'(x) + f(x + e_1) \leq f(x) + f'(x + e_1),$$

which are exactly the conditions. Instead of the convexity of f' the convexity of f can be used together with $f'(x + e_1) + f(x + 2e_1) \leq f(x + e_1) + f'(x + 2e_1)$.

Proving the third statement is a little bit more involved. We make a distinction between $x_1 = 0$ and $x_1 > 0$. When $x_1 = 0$, $\Delta_1 T_{D(\mu+\delta)} f(x) \leq \Delta_1 T_{D(\mu)} f'(x)$ reduces to

$$(1 - \mu - \delta)f(x + e_1) + (1 - \mu)f'(x)$$
$$\leq (1 - \mu - \delta)f(x) + (1 - \mu)f'(x + e_1),$$

which is the sum of

$$(1 - \mu - \delta)f(x + e_1) + (1 - \mu - \delta)f'(x)$$
$$\leq (1 - \mu - \delta)f(x) + (1 - \mu - \delta)f'(x + e_1)$$

and

$$\delta f'(x) \leq \delta f'(x + e_1).$$

The first is equivalent to $\Delta_1 T_{D(\mu+\delta)} f(x) \leq \Delta_1 T_{D(\mu)} f'(x)$, the second is $f' \in I$. Now consider $x_1 > 0$. Writing out the inequality leads to a convex combination of $\Delta_1 f(x) \leq \Delta_1 f'(x)$, $\Delta_1 f(x - e_1) \leq \Delta_1 f'(x - e_1)$ and

$$f(x) + f'(x) \leq f'(x + e_1) + f(x - e_1).$$

The last inequality follows in the same way as (11.1).

Proving the fourth statement is again more involved. Consider first the case $0 < x_1 < s$ (the case $x_1 = 0$ is trivial). Then $\Delta_1 T_{D(s+1)} f(x) \leq \Delta_1 T_{D(s)} f'(x)$ is equivalent to a convex combination of $\Delta_1 f(x) \leq \Delta_1 f'(x)$ and

$$f'(x - e_1) + f(x) \leq f(x - e_1) + f'(x). \tag{11.2}$$

The latter inequalities hold by induction. Now consider $x_1 = s$. Then we have a combination of $\Delta_1 f(x) \leq \Delta_1 f'(x)$, (11.2), and $f'(x - e_1) \leq$

$f'(x)$, which is where we need $f \in I$. Finally, consider $x_1 > s$. Then we get a combination of $\Delta_1 f(x) \leq \Delta_1 f'(x)$, (11.2), and

$$f'(x) + f(x) \leq f(x - e_1) + f'(x + e_1).$$

The proof of the latter inequality is equivalent to that of (11.1).   □

Now we are in a position to prove convexity in the arrival rate and the number of servers.

Lemma 11.6. For functions f, f', and f'' it holds that:

— $2f' \leq f + f'' \Rightarrow 2Tf' \leq Tf + Tf''$ for $T \in \{T_{\text{env}}, T_{\text{max}}, T_{\text{A}}, T_{\text{FS}}, T_{\text{disc}}, T_{\text{D}}\}$;

— $2f' \leq f + f'', f, f', f'' \in Cx, \Delta_1 f \leq \Delta_1 f' \leq \Delta_1 f'' \Rightarrow 2T_{\text{A}(\lambda+\delta)}f' \leq T_{\text{A}(\lambda)}f + T_{\text{A}(\lambda+2\delta)}f''$;

— $2f' \leq f + f'', \ f, f', f'' \in Cx, \ \Delta_1 f \leq \Delta_1 f' \leq \Delta_1 f'', \ f, f' \in I \Rightarrow 2T_{\text{D}(\mu+\delta)}f' \leq T_{\text{D}(\mu)}f + T_{\text{D}(\mu+2\delta)}f''$;

— $2f' \leq f + f'', \ f, f', f'' \in Cx, \ \Delta_1 f \leq \Delta_1 f' \leq \Delta_1 f'', \ f, f' \in I \Rightarrow 2T_{\text{D}(s+1)}f' \leq T_{\text{D}(s)}f + T_{\text{D}(s+2)}f''$.

Proof. The proof of the first statement, for all but T_{max}, follows from the fact that $2f' \leq f + f''$ is preserved under linear combinations. It is readily seen that also maximization is preserved. This does not hold for minimization, the reason why T_{min}, T_{CD}, and T_{CA} are excluded.

Using $2f' \leq f + f''$ it follows that $2T_{\text{A}(\lambda+\delta)}f' \leq T_{\text{A}(\lambda+\delta)}f + T_{\text{A}(\lambda+\delta)}f''$. Thus it suffices to show

$$T_{\text{A}(\lambda+\delta)}f + T_{\text{A}(\lambda+\delta)}f'' \leq T_{\text{A}(\lambda)}f + T_{\text{A}(\lambda+2\delta)}f''.$$

Writing out this inequality shows that it is equivalent to $\Delta_1 f \leq \Delta_1 f''$, the second statement.

For the third statement we have to show

$$T_{\text{D}(\mu+\delta)}f + T_{\text{D}(\mu+\delta)}f'' \leq T_{\text{D}(\mu+2\delta)}f + T_{\text{D}(\mu)}f''. \qquad (11.3)$$

When $x_1 = 0$, then (11.3) reduces to an equality. When $x_1 > 0$, then it gives out

$$f(x) + f''(x - e_1) \leq f(x - e_1) + f''(x),$$

which follows from $\Delta_1 f \leq \Delta_1 f''$.

To prove the last statement, we have to show that

$$T_{\mathrm{D}(s+1)}f + T_{\mathrm{D}(s+1)}f'' \leq T_{\mathrm{D}(s+2)}f + T_{\mathrm{D}(s)}f''. \qquad (11.4)$$

Consider first $x_1 \leq s$. Then (11.4) is an equality. Next consider $x_1 = s + 1$. Then (11.4) reduces to $f(x - e_1) \leq f(x)$. Finally consider $x_1 > s + 1$. Then (11.4) reduces to $f(x) + f''(x - e_1) \leq f(x - e_1) + f''(x)$, as in the proof of the third statement. $\qquad \square$

Theorem 11.7. Value functions consisting of operators from the set $\{T_{\mathrm{env}}, T_{\mathrm{A}(\lambda)}, T_{\mathrm{FS}}, T_{\mathrm{disc}}$ with $C \in I \cap Cx, T_{\mathrm{D}}$ with $\mu(x) \in I \cap Cv\}, T_{\mathrm{D}(\mu)}, T_{\mathrm{D}(s)}\}$ are increasing and convex in the arrival rate, departure rate, and/or number of servers, assuming $V_0 = 0$.

Proof. According to Theorem 9.3 the value functions are convex and increasing. Lemmas 11.1 (first statement) and 11.4 combined to show the increasingness in the arrival rate, departure rate, and/or number of servers. The convexity in these parameters follows from Lemmas 11.5 and 11.6. $\qquad \square$

Of course the initial condition $V_0 = 0$ can be weakened.

How can Theorem 11.7 be used? As an example, suppose we have a number of service centers, each of which can be modeled as an $\mathrm{M}/\mathrm{M}/s$ queue. The question is how to allocate the total pool of servers to the service centers as to minimize the total expected number of customers in the system. Note that this, thanks to Little's law, is equivalent to the expected time customers spend in the system. $C(x) = x$ is a linear increasing cost function, and thus we can apply Theorem 11.7. First we should assign as many servers to each queue as to make it stable, otherwise the objective is ∞. The remaining servers are assigned one by one in a greedy way, where the objective is minimized the most. Thanks to the convexity the return of adding servers to a queue is deminishing. Therefore the greedy assignment algorithm leads to a globally optimal assignment.

An interesting variant to this problem is considering expected waiting times instead of sojourn or system times. Now the direct costs become $C(x) = (x - s)^+$, depending on s. Denote this dependence by $C_s(x)$. Thus for this choice Lemmas 11.5 and 11.6 should be extended to different operators T_{disc}. It is readily seen that we need $\Delta_1 C_{s+1} \leq \Delta_1 C_s$ and $2C_{s+1} \leq C_s + C_{s+2}$ to make these lemmas work. Both inequalities hold, and therefore the average waiting times are also convex in the parameters.

In many applications, especially in call centers, it is of interest to look at functions of the waiting time distribution different from just looking at the expectation. The simplest and best known example is the probability that the waiting time exceeds a number t. Consider the case $t = 0$. Then the direct costs are as follows: $C(x) = \text{I\!I}\{x \geq s\}$. This function is clearly not convex in x, thus our methods cannot be applied to this system. The question that remains is whether some structural result holds for the stationary $M/M/s$ queue as a function of s. Using analytical techniques Jagers and van Doorn [34] have shown that for the stable stationary $M/M/s$ queue $\text{I\!P}(W_q \leq t)$ is increasing concave in s.

An interesting question is whether there is a functional of the waiting time which is convex in x and more sensitive than the expected waiting time. In Koole [43] such a cost function is proposed: $\text{I\!E}(W_q - t)^+$. Using a coupling argument it can indeed be seen that this cost function is convex in x.

Another interesting question is whether the results on $T_{\text{D}(s)}$ can be extended to abandonments. This can indeed be done, as long as the abandonment rate is lower than the service rate.

Finally, we pose ourselves the question whether some results hold for operators that are excluded from the comparison. For example: is the stationary queue length in the $M/M/1$ queue with admission control convex in the arrival rate? The answer to this intriguing question is no, see Varga [80] for a counterexample. The intuition behind the counterexample is as follows. The threshold is decreasing in λ (the arrival rate). For every fixed threshold level the value function is convex. The minimum of these functions is the value function of the queue with admission control. The minimum of a number of convex functions need

not be convex, and numerically it was shown in [80] that this is indeed not the case.

11.3   More-Dimensional Models

Up to now we dealt elaborately with the comparison of different systems for one-dimensional systems. These results were mainly based on the convexity of the value function in a certain parameter. These results can be cast in a multi-dimensional setting for those cases where the convexity holds, i.e., where we have MM or DC, or Sub and Sub$C$ in the case of two-dimensions.

11.4   Bibliographical Notes

Many monotonicity results for one-dimensional queueing systems can be found in Liyanage and Shanthikumar [53]. See Armony *et al.* [9] for recent results on M/M/s queues with impatient customers. For the Erlang blocking model with multiple customer classes we refer to Nain [60] and Ross and Yao [69]. These papers use stochastic orders, combined with sample path arguments.

Independently, Çil *et al.* [21] have developed a similar framework as the one described in this section to study convexity of the value function in several parameters of one-dimensional systems, notably the arrival rate, the service rate, and the buffer size. Fu *et al.* [24] already showed the submodularity of the queue length of the M/G/s queue in the state and the number of servers, using dynamic programming. They apply this to a transient staffing problem.

12

Conclusions and Future Research Directions

This chapter describes and categorizes the state of the art of dynamic programming monotonicity results for multi-dimensional discrete-event systems. It also makes the analysis of new models much simpler. Results for already analyzed operators can be used, and for each function class of Chapter 9 it can be seen if the model fits in. In the remainder of this final chapter, we discuss the limitations of the methods, alternative methods, and possible future research directions.

The best known alternative to DP is stochastic coupling. The central idea is that trajectories of the process, using different initial states and/or policies, are compared. An elegant proof using this method of the optimality or routing to the shortest queue, the symmetric routing model of Section 9.2, can be found in Walrand [82]. Trajectories under two different policies are coupled such that arrivals occur at the same time, as do departures from the longer queues and departures from the shorter queues. Using forward induction in time it can be shown that the trajactory under shortest-queue routing has at all times less customers in the system and a shorter longest queue (i.e., is smaller in the Schur convex sense).

Coupling has several advantages and disadvantages. An advantage is that it is easier to deal with general service times. On the other hand,

convexity results are harder to derive and are different to come with a formal framework; interchange arguments remain partly intuitive and are therefore hard to check. For an overview of the method and an attempt to formalize it, see Liu *et al.* [52].

A few other methods are used as well. Stidham and Weber [76] use an intriguing argument to determine the average-cost optimal policy in a single queue with not necessarily convex costs. Their method involves the total costs until state 0 is reached. Blanc *et al.* [16] have a two-dimensional model with a concave cost function. Their method involves the partial construction of the solution of the dynamic programming equation.

Evidently there are many models that fall outside the techniques discussed so far, and even if a model falls within, then the optimal policy is often very complicated and only partially specified (for example, an increasing switching curve). For this reason many authors have studied optimal policies for some related simpler model, such as a fluid limit. Often this leads to gross oversimplications. The study by Chen *et al.* [20] is an example where this is not the case.

A major disadvantage of DP is the fact that the results we prove for V_n have to hold for all n. It is by taking limits that we obtain results for long-run average and thus also stationary situations. Thus any result that holds only in the stationary situation cannot be obtained directly by DP. A possible solution is to consider $\mathbb{E}V_n(X)$ for X stationary (for 1 or more components) instead of $V_n(x)$ for every x. That is the method that is used successfully in Section 11.1 to analyze systems with different on–off arrival processes. It is to be seen whether this method is useful for a wider class of problems. Problems that can potentially be solved by an extension of this method are Klimov's problem (an extension of the single-server general-routing model of Section 9.1, see [37]) and bandit problems.

An open problem that is central to the method is the extension of the results in [28, 50] to more than two servers. It is explained in Section 9.5 why this cannot be done directly. However, it is generally believed that the convexity and threshold results should still hold for more than two servers. A valid proof of this would be a major breakthrough.

Current reserach attention has shifted partially from obtaining theoretical monotonicity results to applications such as the comparison of systems, see Çil *et al.* [21]. Further comparison results can certainly be obtained. Also the relation between the partial characterization of optimal policies and adaptive control (e.g., the control of systems where certain parameters are unknown) needs further exploration.

References

[1] E. Altman, B. Gaujal, and A. Hordijk, *Discrete-Event Control of Stochastic Networks: Multimodularity and Regularity.* Lecture Notes in Mathematics 1829, Springer, 2003.

[2] E. Altman, T. Jiménez, and G. M. Koole, "On optimal call admission control in a resource-sharing system," *IEEE Transactions on Communications*, vol. 49, pp. 1659–1668, 2001.

[3] E. Altman, T. Jiménez, and G. M. Koole, "On the comparison of queueing systems with their fluid limits," *Probability in the Engineering and Informational Sciences*, vol. 15, pp. 165–178, 2001.

[4] E. Altman and G. M. Koole, "Stochastic scheduling games and Markov decision arrival processes," *Computers and Mathematics with Applications*, vol. 26, no. 6, pp. 141–148, 1993.

[5] E. Altman and G. M. Koole, "Control of a random walk with noisy delayed information," *Systems and Control Letters*, vol. 24, pp. 207–213, 1995.

[6] E. Altman and G. M. Koole, "On submodular value functions and complex dynamic programming," *Stochastic Models*, vol. 14, pp. 1051–1072, 1998.

[7] E. Altman and P. Nain, "Closed-loop control with delayed information," *Performance Evaluation Review*, vol. 20, pp. 193–204, 1992.

[8] E. Altman and S. Stidham Jr., "Optimality of monotonic policies for two-action markovian decision processes, with applications to control of queues with delayed information," *Queueing Systems*, vol. 21, pp. 267–291, 1995.

[9] M. Armony, E. Plambeck, and S. Seshadri, "Convexity properties and comparative statics for M/M/S queues with balking and reneging," Working paper, 2006.

[10] D. Artiges, "Optimal routing into two heterogeneous service stations with delayed information," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1234–1236, 1995.

[11] S. Asmussen and G. M. Koole, "Marked point processes as limits of Markovian arrival streams," *Journal of Applied Probability*, vol. 30, pp. 365–372, 1993.

[12] J. S. Baras, D.-J. Ma, and A. M. Makowski, "K competing queues with geometric service requirements and linear costs: The μc-rule is always optimal," *Systems and Control Letters*, vol. 6, pp. 173–180, 1985.

[13] S. Benjaafar, W. L. Cooper, and S. Mardan, *Production-inventory systems with imperfect advance demand information and due-date updates*. Technical report, University of Minnesota, 2005. Working Paper.

[14] S. Benjaafar and M. ElHafsi, "Production and inventory control of a single product assemble-to-order system with multiple customer classes," *Management Science*, vol. 52, pp. 1896–1912, 2006.

[15] F. J. Beutler and D. Teneketzis, "Routing in queueing networks under imperfect information: Stochastic dominance and thresholds," *Stochastics and Stochastics Reports*, vol. 26, pp. 81–100, 1989.

[16] J. P. C. Blanc, P. R. de Waal, P. Nain, and D. Towsley, "Optimal control of admission to a multiserver queue with two arrival streams," *IEEE Transactions on Automatic Control*, pp. 785–797, 1992.

[17] G. Brouns and J. van der Wal, "Optimal threshold policies in a two-class preemptive priority queue with admission and termination control," *Queueing Systems*, vol. 54, pp. 21–33, 2006.

[18] C. Buyukkoc, P. Varaiya, and J. Walrand, "The $c\mu$ rule revisited," *Advances in Applied Probability*, vol. 17, pp. 237–238, 1985.

[19] C. S. Chang, X. Chao, M. Pinedo, and R. R. Weber, "On the optimality of LEPT and $c\mu$-rules for machines in parallel," *Journal of Applied Probability*, vol. 29, pp. 667–681, 1992.

[20] M. Chen, C. Pandit, and S. Meyn, "In search of sensitivity in network optimization," *Queueing Systems*, vol. 44, pp. 313–363, 2003.

[21] E. B. Çil, E. L. Örmeci, and F. Karaesmen, "Effects of system parameters on the optimal policy structure in a class of queueing control problems," Working paper, 2007.

[22] F. de Véricourt and Y.-P. Zhou, "On the incomplete results for the heterogeneous server problem," *Queueing Systems*, vol. 52, pp. 189–191, 2006.

[23] C. Derman, G. J. Lieberman, and S. M. Ross, "On the optimal assignment of servers and a repairman," *Journal of Applied Probability*, vol. 17, pp. 577–581, 1980.

[24] M. C. Fu, S. I. Marcus, and I.-J. Wang, "Monotone optimal policies for a transient queueing staffing problem," *Operations Research*, vol. 48, pp. 327–331, 2000.

[25] H. A. Ghoneim and S. Stidham Jr., "Control of arrivals to two queues in series," *European Journal of Operational Research*, vol. 21, pp. 399–409, 1985.

[26] P. Glasserman and D. D. Yao, "Monotone optimal control of permutable GSMPs," *Mathematics of Operations Research*, vol. 19, pp. 449–476, 1994.

[27] P. Glasserman and D. D. Yao, *Monotone Structure in Discrete Event Systems*. Wiley, 1994.

[28] B. Hajek, "Optimal control of two interacting service stations," *IEEE Transactions on Automatic Control*, vol. 29, pp. 491–499, 1984.

[29] B. Hajek, "Extremal splitting of point processes," *Mathematics of Operations Research*, vol. 10, pp. 543–556, 1985.

[30] R. Hariharan, V. G. Kulkarni, and S. Stidham Jr., "Optimal control of two parallel infinite-server queues," in *Proceedings of the 29th CDC*, (IEEE, ed.), pp. 1329–1335, 1990.

[31] A. Hordijk and G. M. Koole, "On the optimality of the generalized shortest queue policy," *Probability in the Engineering and Informational Sciences*, vol. 4, pp. 477–487, 1990.

[32] A. Hordijk and G. M. Koole, "On the assignment of customers to parallel queues," *Probability in the Engineering and Informational Sciences*, vol. 6, pp. 495–511, 1992.

[33] A. Hordijk and G. M. Koole, "On the optimality of LEPT and μc rules for parallel processors and dependent arrival processes," *Advances in Applied Probability*, vol. 25, pp. 979–996, 1993.

[34] A. A. Jagers and E. A. van Doorn, "Convexity of functions which are generalizations of the Erlang loss function and the Erlang delay function: Problem 90-8," *SIAM Review*, vol. 33, pp. 281–282, 1991.

[35] P. K. Johri, "Optimality of the shortest line discipline with state-dependent service times," *European Journal of Operational Research*, vol. 41, pp. 157–161, 1989.

[36] J. Kimemia and S. B. Gershwin, "An algorithm for the computer control of a flexible manufacturing system," *Modeling and control of automated manufacturing systems table of contents*, pp. 119–128, 1989.

[37] G. P. Klimov, "Time-sharing service systems I," *Theory of Probability and its Applications*, vol. 19, pp. 532–551, 1974.

[38] G. M. Koole, "Optimal server assignment in the case of service times with monotone failure rates," *Systems and Control Letters*, vol. 20, pp. 233–238, 1993.

[39] G. M. Koole, *Stochastic Scheduling and Dynamic Programming*. CWI Tract 113, Amsterdam: CWI, 1995.

[40] G. M. Koole, "Structural results for the control of queueing systems using event-based dynamic programming," *Queueing Systems*, vol. 30, pp. 323–339, 1998.

[41] G. M. Koole, "A transformation method for stochastic control problems with partial observations," *Systems and Control Letters*, vol. 35, pp. 301–308, 1998.

[42] G. M. Koole, "Stochastic scheduling with event-based dynamic programming," *Mathematical Methods of Operations Research*, vol. 51, pp. 249–261, 2000.

[43] G. M. Koole, "Redefining the service level in call centers," Working paper, 2003.

[44] G. M. Koole, "Convexity in tandem queues," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 13–31, 2004.

[45] G. M. Koole and Z. Liu, "Stochastic bounds for queueing systems with multiple on-off sources," *Probability in the Engineering and Informational Sciences*, vol. 12, pp. 25–48, 1998.

[46] G. M. Koole, P. D. Sparaggis, and D. Towsley, "Minimizing response times and queue lengths in systems of parallel queues," *Journal of Applied Probability*, vol. 36, pp. 1185–1193, 1999.

[47] G. Koole and M. Vrijenhoek, "Scheduling a repairman in a finite source system," *Mathematical Methods of Operations Research*, vol. 44, pp. 333–344, 1996.

[48] J. Kuri and A. Kumar, "Optimal control of arrivals to queues with delayed queue length information," in *Proceedings of the 31th IEEE Conference on Decision and Control*, 1992.

[49] H. M. Liang and V. G. Kulkarni, "Optimal routing control in retrial queues," in *Applied Probability and Stochastic Processes*, (J. G. Shanthikumar and U. Sumita, eds.), Kluwer, 1999.

[50] W. Lin and P. R. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Transactions on Automatic Control*, vol. 29, pp. 696–703, 1984.

[51] S. A. Lippman, "Applying a new device in the optimization of exponential queueing systems," *Operations Research*, vol. 23, pp. 687–710, 1975.

[52] Z. Liu, P. Nain, and D. Towsley, "Sample path methods in the control of queues," *Queueing Systems*, vol. 21, pp. 293–335, 1995.

[53] L. Liyanage and J. G. Shanthikumar, "Second-order properties of single-stage queueing systems," in *Queueing and Related Models*, (U. N. Bhat and I. S. Basawa, eds.), pp. 129–160, Oxford, 1992.

[54] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*. Academic Press, 1979.

[55] R. Menich and R. F. Serfozo, "Monotonicity and optimality of symmetric parallel processing systems," *Queueing Systems*, vol. 9, pp. 403–418, 1991.

[56] N. Miyoshi, M. Ohnishi, and N. Okino, "Optimal service control of a station connected with two parallel substations," *IEEE Transactions on Automatic Control*, vol. 38, pp. 1567–1572, 1993.

[57] A. Müller, "On a conjecture of Koole," *Probability in the Engineering and Informational Sciences*, vol. 12, pp. 141–142, 1998.

[58] A. Müller and M. Scarsini, "Stochastic comparison of random vectors with a common copula," *Mathematics of Operations Research*, vol. 26, pp. 723–740, 2001.

[59] P. Nain, "Interchange arguments for classical scheduling problems in queues," *Systems and Control Letters*, vol. 12, pp. 177–184, 1989.

[60] P. Nain, "Qualitative properties of the Erlang blocking model with heterogeneous user requirements," *Queueing Systems*, vol. 6, pp. 189–206, 1990.

[61] J. Ni, H. K. Tsang, S. Tatikonda, and B. Bensaou, "Threshold and reservation based call admission control policies for multiservice resource-sharing systems," in *Proceedings of INFOCOM*, pp. 773–783, IEEE, 2005.

[62] M. Nuyens and A. Wierman, "The foregound-background processor-sharing queue: A survey," 2004.

[63] E. L. Örmeci and A. Burnetas, "Dynamic admission control for loss systems with batch arrivals," *Advances in Applied Probability*, vol. 37, pp. 915–937, 2005.

[64] E. L. Örmeci, A. Burnetas, and J. van der Wal, "Admission policies for a two class loss system," *Stochastic Models*, vol. 17, pp. 513–540, 2001.

[65] M. L. Puterman, *Markov Decision Processes*. Wiley, 1994.

[66] R. Ramjee, R. Nagarajan, and D. Towsley, "On optimal call admission control in cellular networks," in *Proceedings of IEEE Infocom '96*, pp. 43–50, 1996.

[67] R. Righter and J. G. Shanthikumar, "Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures," *Probability in the Engineering and Informational Sciences*, vol. 3, pp. 323–333, 1989.

[68] K. W. Ross and D. H. K. Tsang, "The stochastic knapsack problem," *IEEE Transactions on Communications*, vol. 37, pp. 740–747, 1989.

[69] K. W. Ross and D. D. Yao, "Monotonicity properties for the stochastic knapsack," *IEEE Transactions on Information Theory*, vol. 36, pp. 1173–1179, 1990.

[70] R. Serfozo, "An equivalence between continuous and discrete time Markov decision processes," *Operations Research*, vol. 27, pp. 616–620, 1979.

[71] M. Shaked and J. G. Shanthikumar, "Paramatric stochastic convexity and concavity of stochastic processes," *Annals of the Institute of Statistical Mathematics*, vol. 42, pp. 509–531, 1990.

[72] D. Song and Y. Sun, "Optimal control structure of an unreliable manufacturing system with random demands," in *Proceedings of the 35th CDC*, pp. 4552–4557, IEEE, 1996.

[73] S. Stidham Jr., "Socially and individually optimal control of arrivals to a $GI|M|1$ queue," *Management Science*, vol. 24, pp. 1598–1610, 1978.

[74] S. Stidham Jr., "Optimal control of admission to a queueing system," *IEEE Transactions on Automatic Control*, vol. 30, pp. 705–713, 1985.

[75] S. Stidham Jr., "Analysis, design, and control of queueing systems," *Operations Research*, vol. 30, pp. 705–713, 2002.

[76] S. Stidham Jr. and R. R. Weber, "Monotonic and insensitive optimal policies for control of queues with undiscounted costs," *Operations Research*, vol. 37, pp. 611–625, 1989.

[77] S. Stidham Jr. and R. R. Weber, "A survey of Markov decision models for control of networks of queues," *Queueing Systems*, vol. 13, pp. 291–314, 1993.

[78] J.-B. Suk and C. G. Cassandras, "optimal scheduling of two competing queues with blocking," *IEEE Transactions on Automatic Control*, vol. 36, pp. 1086–1091, 1991.

[79] N. M. van Dijk, "On the finite horizon Bellman equation for controlled Markov jump models with unbounded characteristics: Existence and approximations," *Stochastic Processes and their Applications*, vol. 28, pp. 141–157, 1988.

[80] Á. Varga, "Monotonicity properties of single server queues," Master's thesis, Vrije Universiteit Amsterdam, 2002.

[81] M. H. Veatch and L. Wein, "Monotone control of queueing networks," *Queueing Systems*, vol. 12, pp. 391–408, 1992.

[82] J. Walrand, *An Introduction to Queueing Networks*. Prentice-Hall, 1988.

[83] R. R. Weber and S. Stidham Jr., "Optimal control of service rates in networks of queues," *Advances in Applied Probability*, vol. 19, pp. 202–218, 1987.

[84] J. Weishaupt, "Optimal myopic policies and index policies for stochastic scheduling problems," *Mathematical Methods of Operations Research*, vol. 40, pp. 75–89, 1994.

[85] W. L. Winston, "Optimality of the shortest line discipline," *Journal of Applied Probability*, vol. 14, pp. 181–189, 1977.