On submodular value functions and complex dynamic programming

Eitan ALTMAN
INRIA
2004 Route des Lucioles
06902 Sophia Antipolis
France

Ger KOOLE
Vrije Universiteit
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands

Published in *Stochastic Models* **14**: 1051–1072, 1998

Abstract: We investigate in this paper submodular value functions using *complex dynamic programming.* In complex dynamic programming (dp) we consider concatenations and linear combinations of standard dp operators, as well as combinations of maximizations and minimizations. These value functions have many applications and interpretations, both in stochastic control (and stochastic zero-sum games) as well as in the analysis of (non-controlled) discrete-event dynamic systems. The submodularity implies the monotonicity of the selectors appearing in the dp equations, which translates, in the context of stochastic control and stochastic games, to monotone optimal policies. Our work is based on the score-space approach of Glasserman and Yao.

Keywords: Dynamic programming, submodularity, admission control, stochastic games.

1 INTRODUCTION

Dynamic programming (dp) has been a key tool for the study of control of stochastic systems in different areas of applications. Quite often however, the direct use of dp for computing performance measures or optimal control policies is impossible due to the "curse of dimensionality". By exploiting some structure of the problem, dp can sometimes be used to establish some properties of the optimal control, and in particular, to show that optimal controls belong to a class of policies that can be characterized by some parameters. The original control problem can then be transformed to a simpler optimization problem over this parameter space. In admission control and flow control problems into queueing networks it has been shown already in the beginning of the seventies ([25, 29]) that, under some conditions, one may restrict to threshold policies, or more generally, to monotone policies. Computation of an optimal policy within these classes of policies goes back to Naor [20].

Since the above and other results were obtained there was a growing interest in identifying general properties of problems involving dp, that produce structural properties of optimal policies, and several abstract approaches have been proposed. Results on the control of generalized semi-Markov processes (GSMP) were obtained in [9, 10], that are based on submodularity properties of the value functions. Their methodology is related to Weber and Stidham [27] who consider a continuous time model of a cycle of queues with Poisson arrivals, and show that submodularity for every pair of service events propagates through the dp recursion. They also indicate how their results can be applied to other models.

In the applications as well as in the abstract framework noted in the previous paragraph, focus is on standard dp equations having a form, in the simplest cases, of a minimization (or maximization) over some sets of "actions" of the sum of an immediate cost plus some "future value" after one step. More generally, they had the form of an immediate cost, plus the sum of minimizations of different immediate costs plus a future value after one step. There exist however more complex forms of dp, which we shall therefore call "complex dp". Here are some examples:

(i) Value functions with both maximization and minimization. They (or the *val* operator of matrix games) have already been used a long time as a tool for solving zero-sum dynamic games, see e.g. [24]. This may reflect dynamic control problems where two controllers have opposite objectives, or also a situation of dynamic control by a single controller with a worst case type of a criterion (e.g. [3]). Recently, this type of dp was used in the context of (non-controlled) discrete event systems, for the performance evaluation of parallel processing with both "or" and "and" constraints, see [17].

(ii) Convex combinations of dp operations. This situation arises in team problems with special information structure (see [16, 22]), and in control problems with a random environment [4, 14, 15].

(iii) Coupling of the dp equation with other linear equations. This occurred in equations describing the time evolution of free choice Petri-nets (which are useful for the modeling of parallel computing), see [7].

(iv) Composition of different dp operators. This situation arises when several (possibly controlled) events happen immediately after each other.

In many cases, the above phenomena appear together (e.g. [4]).

The starting point of this paper is the framework of Glasserman and Yao [9, 10] for analyzing monotone structural properties in dp, who themselves built on work by Topkis [26]. The analysis is based on the submodularity of value functions that implies the monotonicity properties of optimal strategies. An essential contribution of Glasserman and Yao [9, 10] was to identify a natural context of analysis of monotonicity. Indeed, they showed that under fairly general conditions, one may obtain monotonicity in some augmented state space of the "scores" (these are counters of different types of events). One may then obtain monotonicity with respect to the original state space if some further (more restrictive) conditions on the structure of the problem are satisfied. The purpose of this paper is to extend the abstract framework of [9, 10] to complex dp. In particular, we shall consider dp involving compositions and convex combinations of dp operators, and combination of both minimization and maximization operators. We further investigate monotonicity in min-max types of dp. In the context of control, the monotonicity results mean that a certain parameter should be controlled monotonically as function of some state parameter.

The current model is very much queueing oriented. For a general approach to monotonicity results for some other types of problems we refer to Hinderer [12] and Hinderer and Stieglitz [13].

In Section 2 we start by formulating the value function in terms of the scores, using complex dp. Given a mapping from scores into states, we derive the value function as formulated in terms of the states. The advantages of using complex dp are explained. Then we present our three main examples: the control of a single queue, the cycle and the tandem of queues studied in [27], and a fork-join queue. For all three it is shown how complex dp allows us to study at the same time different variants of the models, such as both discrete and continuous time models.

Section 3 contains the main result. First we study an unbounded score space, and it is shown how submodularity, which is the crucial property, propagates through the dp recursion. Bounds to the score space (which prevent for example a negative queue length in the state space model) are introduced by taking costs outside the bounded score space infinite, and it is shown how submodularity leads to the desired monotonicity properties. The results are illustrated with the examples mentioned above. Results found in the literature on the control of the single queue and the cycle of queues are generalized to general arrival streams and discrete-time versions of the models. This section also contains new results for a single server queue with batch arrivals and for the fork-join queue.

We study games in Section 4. We have to confine ourselves to two events, assuming that one of them has a maximization instead of a minimization operator. Again monotonicity results hold, which is illustrated with the control of a single server queue.

2 SCORES AND COMPLEX DP

Our approach in this section is to start by defining a value function, based on a set of dp operators. We shall present two variants of the dp. The first corresponds to working in the so called score space (or model state), and the second corresponds to the so called system state. Quantities that appear in the dp equations may have different interpretations, which will be presented later.

2.1 The value function: scores and environment states

We introduce two sets that, together, have the role of “states” of a system described by dp equations:

- The countable set of *environment states*, denoted by V,
- the set of scores $D = (D_v, v \in V)$, where $D_v \subset \mathbb{N}^m$ are the scores available in environment state v (with $\mathbb{N} = \{0, 1, 2, \ldots\}$, and m some fixed number).

An element (v, d), with $v \in V$ and $d \in D_v$ is called a (model) state. Abusing notation a little we denote the set of all states with $V \times D$.

The advantage of adding the environment states is that in the context of stochastic control, it allows us to relax the independence and exponential assumptions on the time between transitions, which was previously considered (see [9, 10]), and is quite unnatural in many applications, such as telecommunication networks.

The role and interpretation of scores and environment states will become clear in many examples later on. In [9, 10] the scores are defined as counters of events in the context of GSMPs.

For every $1 \le i \le m$ and every function f (defined on all (v, d) with $v \in V$ and $d \in D_v$) we define the dynamic programming (dp) operator T_i by

$$T_i f(v, d) = h_i(v, d) + \min_{\underline{\mu}_i \le \mu \le \overline{\mu}_i} \{c_i(\mu) + \mu f(v, d + e_i) + (1 - \mu) f(v, d)\} \qquad (1)$$

if $d + e_i \in D_v$ and

$$T_i f(v, d) = h_i(v, d) + f(v, d)$$

if $d + e_i \notin D_v$. In the first case we call event i active, in the latter case inactive. Both h_i and c_i are cost functions, c_i is assumed to be continuous in μ, $0 \le \underline{\mu}_i \le \overline{\mu}_i \le 1$, and $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 in ith position.

Now we define the value function. Let S_r for $1 \le r \le R$ represent a sequence of dp operators, i.e., S_r is of the form $S_r = T_{i_1} \ldots T_{i_a}$. The value function is recursively defined by

$$J_{n+1}(v, d) = \sum_{w \in V} \lambda_{vw} \sum_{1 \le r \le R} q^r_{vw} S_r J_n(w, d). \qquad (2)$$

Here we call $\lambda_{vw} \ge 0$ the transition probabilities of the environment, $q^r_{vw} \ge 0$ is the probability that S_r is taken if a transition from v to w occurs. We assume that $\sum_w \lambda_{vw} \le 1$ for each v, that $\sum_r q^r_{vw} \le 1$ for each v and w, and that J_0 is given. Furthermore, we assume that $D_v \subset D_w$ if $\lambda_{vw} > 0$. See Subsection 2.5 for some example models. For the infinite horizon case we shall take $\sum_w \lambda_{vw} = \beta$, where $\beta < 1$ is the discount factor.

The function J_n is our main subject of investigation. In Section 3 we formulate conditions on D, f_i, c_i and J_0 under which certain properties of J_n propagate. From this we will be able to derive monotonicity properties of the optimal policy. But let us first consider which systems can be analyzed using this type of value function.

2.2 Model states versus system states

Quite often, there may exist a state space X that describes the system and is simpler and "smaller" (in some sense) than $V \times D$. E.g., consider an M/M/1 queue; its state can be described by two counters, one counting the arrivals and the other the departures (this would correspond to the D component in the previous state description); alternatively, the queue can be described by a single system variable describing its length.

As in [9, 10], it is in the first formulation (i.e., of the enlarged state space $V \times D$) that monotonicity will be obtained in a natural way, and under some further assumptions, these properties can be translated to monotonicity properties with respect to X.

We next obtain a dynamic programming equation for the new state space X that corresponds to (2). We assume that there exists a function ϕ which maps the model states (v, d) into the system states such that $X = \{\phi(v, d) \mid v \in V, d \in D_v\}$. We will construct a value function $\tilde{J}_n$ on X such that $\tilde{J}_n(\phi(v, d)) = J_n(v, d)$, and such that minimizing actions correspond. The main problem is the fact that a system state can be represented by several model states, i.e., ϕ is not injective.

We define $\tilde{J}_0$, $\tilde{h}_i : X \to \mathbb{R}$, for which we assume the following.

Assumption 2.1 *For all (v^1, d^1) and (v^2, d^2), $d^i \in D_{v^i}$ such that $\phi(v^1, d^1) = \phi(v^2, d^2)$, we assume that $\tilde{J}_0(\phi(v^1, d^1)) = J_0(v^1, d^1) = J_0(v^2, d^2)$ and that $\tilde{h}_i(\phi(v^1, d^1)) = h_i(v^1, d^1) = h_i(v^2, d^2)$.*

As a result all direct costs in corresponding states are equal.

Now we define the dp operators for the system. To assure that the same events are active in corresponding model states, we formulate the following assumption.

Assumption 2.2 *For all* (v^1, d^1) *and* (v^2, d^2), $d^i \in D_{v^i}$ *such that* $\phi(v^1, d^1) = \phi(v^2, d^2)$, *we assume that an event is either active or inactive in both* (v^1, d^1) *and* (v^2, d^2).

To every event in the model space, resulting in a transition say from (v, d) to $(v, d+e_i)$, corresponds a transition in the system state, say from $x = \phi(v, d)$ to $A_i x = \phi(v, d + e_i)$. ($A_i x$ are defined only for $x = \phi(v, d)$ satisfying $d + e_i \in D_v$; for other x's we say (with some abuse of language) that $A_i x \notin X$.) Thus define the dp operators $\tilde{T}_i$ by

$$\tilde{T}_i \tilde{f}(x) = \tilde{h}_i(x) + \min_{\underline{\mu}_i \le \mu \le \overline{\mu}_i} \{c_i(\mu) + \mu \tilde{f}(A_i x) + (1 - \mu)\tilde{f}(x)\}$$

if $A_i x \in X$ and

$$\tilde{T}_i \tilde{f}(x) = \tilde{h}_i(x) + \tilde{f}(x)$$

if $A_i x \notin X$. From this it follows that $\tilde{T}_i \tilde{f}(x) = T_i f(v, d)$.

Finally, we have to consider the changes in the environment. Here the conditions are somewhat complicated; the reasons for this choice are well illustrated in the admission control example, later in this section. We assume the following.

Assumption 2.3 *For all* (v^1, d^1) *and* (v^2, d^2), $d^i \in D_{v^i}$ *such that* $\phi(v^1, d^1) = \phi(v^2, d^2)$, *and* w^1, w^2 *such that* $\phi(w^1, d^1) = \phi(w^2, d^2)$, *we assume that* $\lambda_{v^1 w^1} = \lambda_{v^2 w^2}$ *and* $q^r_{v^1 w^1} = q^r_{v^2 w^2}$.

Now we can unambiguously define $\tilde{\lambda}$ and $\tilde{q}$ by $\tilde{\lambda}_{\phi(v,d)\phi(w,d)} = \lambda_{vw}$ and $\tilde{q}^r_{\phi(v,d)\phi(w,d)} = q^r_{vw}$.

Finally, if $S_r = T_{i_1} \dots T_{i_a}$, let $\tilde{S}_r = \tilde{T}_{i_1} \dots \tilde{T}_{i_a}$. The value function in the system state is defined by

$$\tilde{J}_{n+1}(x) = \sum_y \tilde{\lambda}_{xy} \sum_{1 \le r \le R} \tilde{q}^r_{xy} \tilde{S}_r \tilde{J}_n(y).$$

The following theorem can now be shown inductively:

Theorem 2.4 *Under Assumptions 2.1, 2.2 and 2.3 we have* $\tilde{J}_n(\phi(v^1, d^1)) = J_n(v^1, d^1) = J_n(v^2, d^2)$ *for all* (v^1, d^1) *and* (v^2, d^2), $d^i \in D_{v^i}$, *such that* $\phi(v^1, d^1) = \phi(v^2, d^2)$.

Often the events have a physical interpretation in the system state, like an arrival to a queue. By defining the system state through the model, and the way J_n is defined, the system state cannot depend on the order of events. This property is called strong permutability in [9]. There the analysis is done in the other direction: starting from a permutable system a value function in terms of the scores is derived.

2.3 Interpretation of the dynamic programming

The value function (2) can be studied at two different levels. First it can be seen as a mapping from J_n to J_{n+1} with states of the form (v, d). We can also look at a more detailed level, for the dp operators separately. This leads to the following definitions.

- *The macro stages:* There are some basic time periods (called macro-stages) parametrized by n, and $J_n(v, d)$ is the value corresponding to n more steps to go, given that the initial state is (v, d).
- *The state:* The state space of the MDP is a tuple (v, d, r, k). r denotes a micro-state (or phase) within the period n, and k will denotes a micro-stage within that phase. k can be taken to be 0 at the beginning of a macro-stage.
- *Actions, transitions and costs:* At the beginning of a macro-stage we are at some state $(v, d, 0)$. With probability $\lambda_{vw} q^r_{vw}$ we move instantaneously to environment state w, and to a type r micro-state (phase), $1 \le r \le R$. We may denote the new state as $(w, d, (r, 0))$. If the transition was to some phase r then a specific r-type sequence of a_r instantaneous controlled transitions occur, and the state evolution has the form

  $$(w, d, (r, 0)) \to (w, d^1, (r, 1)) \to ... \to (w, d^{a_r - 1}, (r, a_r - 1)) \to (w, d^{a_r}, 0).$$

  We may thus consider a second time counter that counts these *micro stages* within the phase r. The state after the kth instantaneous transition within a r-type sequence (r phase) has the form $(w, d^k, (r, k))$. The transition probabilities within consecutive micro-stages are functions of the current state and action. At the kth micro-stage, an action $\mu \in [\underline{\mu}_{j(k,r)}, \overline{\mu}_{j(k,r)}]$ is chosen, the state moves to $(w, d + e_{j(k,r)}, (r, k + 1))$

with probability μ, and to $(w, d, (r, k+1))$ with probability $1-\mu$. (Here, $j(k, r)$ defines which dp operator is used at the kth micro-stage in the rth macro-stage.) Moreover, an immediate cost of $h_{j(k,r)}(w, d)+c_{j(k,r)}(\mu)$ is charged. After a_r micro transitions we arrive to a new macro-stage with a state of the form $(w, d^{a_r}, 0)$.

In the above description we have in fact two time parameters, one counting the time period n (the macro-stage), and a second, within each time period, which counts the micro-stage. This distinction may become important when one is interested in monotonicity properties in time.

Standard objectives are expected costs over a finite interval, the expected average costs and discounted costs. Note that discounting can be introduced by taking $\sum_w \lambda_{vw} = \beta$, where $\beta < 1$ is the discount factor. There is a large literature on the existence of optimal policies and the convergence of J_n to the discounted or average costs. Although general conditions can be given, we refrain from doing so and deal with these issues for each model separately.

In applications, a macro-stage often corresponds to models with either:
(i) a fixed time period (slotted time); these models are called *discrete-time* models. Often (2) consists of a single concatenation of operators.
(ii) sampled versions of continuous time models. In that case, the duration of a macro-stage is exponentially distributed. This model is the one obtained by the uniformization techniques due to Lippman [19] (see also Serfozo [23]). Often (2) consists of a convex combination of operators. (This is the type of model that was studied in [9, 10].) We call these the *time-discretized models*.

A basic restrictive feature of the time-discretized models, as studied in [9, 10], is that only *one single transition* may occur at a time (i.e., at a macro-stage). But in discrete time models typically several transitions may occur in each time-period (macro-stage). This makes discrete time models usually harder to analyze. However, if we assume that in a discrete time model events happen one after each other, possibly instantaneously, we can model it with consecutive dp operators, each representing a single event. Thus our dp formulation (2) enables to handle these situations as well.

2.4 Infinite horizon

The dynamic programming equation (2) may appear in the form of a fixed point equation:

$$J(v,d) = \sum_{w\in V} \lambda_{vw} \sum_{1\le r\le R} q^r_{vw} S_r J(w,d). \tag{3}$$

Under fairly general conditions, there exists a solution to (3) and it is unique within some class of functions (see e.g. [21] p. 236).

Moreover, in the context of control, $J(v,d)$ equals the optimal cost (the minimum cost over all policies) for a control problem with an infinite horizon (infinitely many macro-stages) and with initial state (v,d). Consider a stationary policy that chooses when the micro-state (phase) and micro-stage are (k,r), a minimizer appearing if $T_{i_k(r)} J'_n(v,d,k,r)$, $k = 0,1,...,a(r)-1$, where $J'(v,d,k,r) = T_{i_{k+1}}(r)...T_{i_a(r)} J(v,d)$. It is well known that this policy is optimal under fairly general conditions (see e.g. [21] p. 236). This happens to be true for both continuous time models that are time-discretized (where discretization is done at time instants that are exponentially distributed, [9, 10]) or for real continuous time control (see e.g. Fleming and Soner [8] Ch. 3).

Finally, $J(v,d)$ can be computed using value iteration, i.e., by calculating J_n inductively in (2) and taking the limit as $n \to \infty$. We present in the next sections conditions for the submodularity of J_n. This will imply the submodularity of the limit J, and hence the existence of monotone policies for both the continuous time control problem as well as its discretized version.

2.5 Examples

Admission control model This example consists of a single queue for which the arrivals can be controlled. To model this we take two events, the first corresponding to arrivals, the second to departures. For the moment we assume that the environment consists of a single state (which we omit from the notation). Thus the model state is $d = (d_1, d_2)$, and $D = \{d \mid d_1 \ge d_2\}$. The two dynamic programming operators can for example be defined by

$$T_1 f(d) = h(d) + \min_{0\le\mu\le\overline{\mu}_1} \{(1-\mu)K + \mu f(d+e_1) + (1-\mu) f(d)\}$$

and

$$T_2 f(d) = h(d) + \min_{\underline{\mu}_2\le\mu\le\overline{\mu}_2} \{-\mu R + \mu f(d+e_2) + (1-\mu) f(d)\}$$

if $d_1 > d_2$ and

$$T_2 f(d) = h(d) + f(d)$$

if $d_1 = d_2$. Here K can be interpreted as blocking costs, R as a reward for serving a customer, and h as holding costs. If $\underline{\mu}_2 = \overline{\mu}_2$ the departures are uncontrolled (the service rate is constant).

The system state is defined by $\phi(d) = d_1 - d_2$. Thus if $\phi(d^1) = \phi(d^2)$, both d^1 and d^2 are either active or inactive, given the definition of D. This ensures that $J_n(d) = \tilde{J}_n(\phi(d))$ if the obvious conditions (on transition probabilities and cost functions) are satisfied.

The value function of a continuous time model (sampled at exponentially distributed times) could now be defined by

$$J_{n+1}(d) = p_1 T_1 J_n(d) + p_2 T_2 J_n(d).$$

(Formally, this means an environment with $V = \{v\}$, $\lambda_{vv} = 1$, $q^i_{vv} = p_i$ and $S_i = T_i$, for $i = 1, 2$ and $p_1 + p_2 = 1$.) This is indeed the value function of a queue with Poisson arrivals (with rate $\gamma p_1 \overline{\mu}_1$) and exponential service times (with rate $\gamma p_2 \overline{\mu}_2$), for an arbitrary constant γ.

A discrete time model could have the value function

$$J_{n+1}(d) = T_1 T_2 J_n(d).$$

Now $\overline{\mu}_1$ and $\overline{\mu}_2$ serve as arrival and departure probabilities.

This simple model can be generalized in various ways. A first way would be to allow batch arrivals. Assume that customers arrive in batches of size B. This can easily be modeled, just by taking $D = \{d \mid Bd_1 \geq d_2\}$ and $\phi(d) = Bd_1 - d_2$.

Another generalization would be a model in which there are both controlled and uncontrolled arrivals. (For example, there could be a minimum rate of arrivals, and the control would then determine the excess beyond this level. In continuous time this could model the superposition of several non-controlled arrival streams with an additional controlled one.) The simplest way to model this would be by taking $\underline{\mu}_1 > 0$. However, this does not always work, as in the case of a discrete time model in which we can have both types of arrivals at the same epoch. This can be solved by defining another operator, or by letting the uncontrolled arrivals be part of the environment, i.e., v counts the number of uncontrolled arrivals. Now $D_v = \{d \mid v + d_1 \geq d_2\}$, and let λ_{vv} and λ_{vv+1} be independent of v, with $\lambda_{vv} + \lambda_{vv+1} = 1$. Note that it follows that $D_v \subset D_{v+1}$, and that $J(v,d)$ depends only on $v + d_1 - d_2$ (which is equal to $\phi(v,d)$), as long as the cost functions do. This makes that all conditions given in this section are satisfied.

The last generalization we discuss is that to general arrival streams. Take $\overline{\mu}_i = 1$, let λ_{vw} be the transition probabilities of some Markov chain. If this Markov chain does not change state, a departure can occur, and thus $q^1_{vv} = 1$ with $S_1 = T_2$. If the Markov chain changes state, an arrival is generated at the transition with probability q^2_{vw} (for $v \neq w$). The corresponding dp operator is $S_2 = T_1$. With probability $q^3_{vw} = 1 - q^2_{vw}$ no arrival occurs, and thus S_3 is the null event. Let again $D_v = \{d \mid d_1 \geq d_2\}$. This results in

$$J_{n+1}(v,d) = \sum_w \lambda_{vw}(q^1_{vw} T_2 J_n(v,d) + q^2_{vw} T_1 J_n(v,d) + q^3_{vw} J_n(v,d)),$$

which is the value function of a continuous time model (sampled at exponentially distributed times) where the arrivals are generated by a Markov arrival process (MAP). Approximation results exist showing that this type of arrival process is dense in the set of all arrival processes (Asmussen & Koole [6]). Of course, the MAP could at the same time be used to govern the departure process, really giving it the interpretation of an environment.

Tandem model In Weber & Stidham [27] a cycle of m queues is considered. Thus customers leaving queue i join queue $i+1$ (with queue $m+1$ identified with queue 1). The service rate at each queue can be controlled. Furthermore, there are uncontrolled exogenous arrivals at each queue. We model this by taking $d = (d_1, \ldots, d_m)$ and $v = (v_1, \ldots, v_m)$, where d_i represents a customer leaving center i, and v_i represents an uncontrolled arrival to center i. Define $s_i = -e_i + e_{i+1}$, and $\phi(v,d) = \sum_i (v_i e_i + d_i s_i)$. The state space is defined by $D_v = \{d \mid \phi(v,d) \geq 0\}$.

In [27] the continuous time model is studied, having a value function of the form

$$J_{n+1}(v,d) = \sum_i \lambda_{vv+e_i} J_n(v+e_i, d) + \lambda_{vv} \sum_i q^i_{vv} T_i J_n(v,d).$$

It is readily seen that this choice satisfies the conditions formulated in this section.

In the next section we will see that our results generalize those in [27]. We show that the results proven there hold also for several related models, like the discrete time version, or the model with arrivals according to a MAP, as described in the admission control model.

Fork-join queue Our third example is that of a fork-join queue. A fork-join queue consists of two or more parallel queues. Jobs arrive at the system according to a Poisson process, and on arrival they place exactly one task in each queue (the fork primitive). Take $|V| = 1$. Let operator T_1 represent the arrival event, and operator T_i, $2 \le i \le m$, the departure event at queue $i-1$. Then $\phi(d) = (d_1 - d_2, \ldots, d_1 - d_m)$, where the ith component represents the number of customers in the ith queue.

The number of customers in state x is given by $\max_i x_i$, if we assume that the queues work in a FIFO manner. This, or a convex function thereof, are interesting cost functions. In the next section we will see how the optimal service rate at each of the queues changes as the state changes. Also the admission control can be studied.

This model can be generalized in various ways, similarly to the previous examples, e.g. to arrivals according to a MAP or to batch arrivals. Combinations with the previous examples are also possible, resulting for example in a cycle of fork-join queues.

Remark In the next section we show that certain properties of the value function J_n hold also for $T_{i_1} \ldots T_{i_a} J_n$. Then, after showing that the same properties are preserved while taking linear combinations, we conclude that the same properties hold for J_{n+1}. This approach however can also be applied to other models, falling outside the scope of this paper. For example, the optimality of the μc rule for discrete time models follows directly from the result for the continuous time version, as given in Hordijk & Koole [15]. Note that this discrete time result is proven directly in Weishaupt [28].

3 MONOTONICITY

3.1 The unconstrained model

We call a function f D-submodular if $f(v, d) + f(v, d + e_i + e_j) \le f(v, d + e_i) + f(v, d + e_j)$ for all $i \ne j$, all $v \in V$, and for all d such that d, $d + e_i + e_j$, $d + e_i$ and $d + e_j \in D_v$.

Let us first assume that $D_v = \mathbb{N}^m$ for all v. This means that each event is always active. We call this the unconstrained case. For $S_r = T_{i_1} \dots T_{i_a}$ define $S_r^{(b)} = T_{i_b} \dots T_{i_a}$. Thus $S_r^{(b)}$ consists of the $a-b$ last operators of S_r. With $S_r^{(a+1)}$ we denote the identity operator. Then we have the following result.

Lemma 3.1 *If h_i for all i and J_0 are D-submodular, then $S_r^{(b)} J_n$ (and in particular J_n) are also D-submodular for all n, r and b.*

Proof We show that if a function f is submodular in (v, d) for some i and j $(i \neq j)$, it propagates through T_k, for an arbitrary k. Assuming that J_n is D-submodular, it then follows that $S_r^{(b)} J_n$ is D-submodular. By noting that convex combinations of submodular functions are again submodular, it is then easily shown that J_{n+1} is also D-submodular.

Thus, we assume that (dropping v from the notation) $f(d)+f(d+e_i+e_j) \leq f(d+e_i)+f(d+e_j)$ $(i \neq j)$. We have to show that $T_k f(d)+T_k f(d+e_i+e_j) \leq T_k f(d+e_i)+T_k f(d+e_j)$, while event k is active in all 4 states. Note first that h_k is submodular. Denote with μ_1 (μ_2) the minimizing μ for score $d+e_i$ $(d+e_j)$ (a minimizing action exists since the minimization is performed on a compact set, and the minimized function can be shown inductively to be continuous). Assume that $\mu_1 \leq \mu_2$ (the other case is similar by symmetry).

First consider the case $i \neq k$. Note that

$$T_k f(d) + T_k f(d+e_i+e_j) \leq h_k(d) + c_k(\mu_1) + \mu_1 f(d+e_k) + (1-\mu_1) f(d) +$$
$$h_k(d+e_i+e_j) + c_k(\mu_2) + \mu_2 f(d+e_i+e_j+e_k) + (1-\mu_2) f(d+e_i+e_j).$$

Summing

$$h_k(d) + h_k(d+e_i+e_j) \leq h_k(d+e_i) + h_k(d+e_j),$$

$$c_k(\mu_1) + c_k(\mu_2) \leq c_k(\mu_1) + c_k(\mu_2),$$

$$\mu_1 f(d+e_k) + \mu_1 f(d+e_i+e_j+e_k) \leq \mu_1 f(d+e_i+e_k) + \mu_1 f(d+e_j+e_k), \quad (4)$$

$$(\mu_2-\mu_1) f(d+e_j) + (\mu_2-\mu_1) f(d+e_i+e_j+e_k) \leq$$
$$(\mu_2-\mu_1) f(d+e_i+e_j) + (\mu_2-\mu_1) f(d+e_j+e_k), \quad (5)$$

and

$$(1-\mu_1) f(d) + (1-\mu_1) f(d+e_i+e_j) \leq (1-\mu_1) f(d+e_i) + (1-\mu_1) f(d+e_j) \quad (6)$$

gives

$$T_k f(d) + T_k f(d+e_i+e_j) \leq T_k f(d+e_i) + T_k f(d+e_j).$$

Note that (5) does not hold if $i = k$.

If $i = k$, we start with

$$T_i f(d) + T_i f(d + e_i + e_j) \leq h_i(d) + c_i(\mu_2) + \mu_2 f(d + e_i) + (1 - \mu_2) f(d) +$$
$$h_i(d + e_i + e_j) + c_i(\mu_1) + \mu_1 f(d + 2e_i + e_j) + (1 - \mu_1) f(d + e_i + e_j).$$

The inequality follows as the previous one if we replace (4)–(6) by

$$\mu_1 f(d + e_i) + \mu_1 f(d + 2e_i + e_j) \leq \mu_1 f(d + 2e_i) + \mu_1 f(d + e_i + e_j)$$

and

$$(1 - \mu_2) f(d) + (1 - \mu_2) f(d + e_i + e_j) \leq (1 - \mu_2) f(d + e_i) + (1 - \mu_2) f(d + e_j). \quad \square$$

Further generality can be obtained by letting $\underline{\mu}_i$ and c_i depend on the environment. As this adds merely to the notation, we will not do this.

Remark In Glasserman & Yao [9, 10] sub and supermodularity is combined, see Lemma 4.2 in [9] which corresponds to Lemma 3.1. Unfortunately there is a gap in its proof, see also [11]. In our opinion Lemma 3.1 is as general as possible.

From Lemma 3.1 the monotonicity of the optimal policy can be derived. (Note that we use increasing and decreasing in the non-strict sense throughout the paper.)

Theorem 3.2 *For the unconstrained case, if h_i for all i and J_0 are D-submodular, then the optimal control of event i is increasing in d_k, for all $i \neq k$.*

Proof $T_i f(v, d)$ can be rewritten as

$$h_i(v, d) + \min_{\underline{\mu}_i \leq \mu \leq \overline{\mu}_i} \{c_i(\mu) + \mu(f(v, d + e_i) - f(v, d))\} + f(v, d).$$

If f is D-submodular, then $f(v, d+e_i+e_j) - f(v, d+e_j) \leq f(v, d+e_i) - f(v, d)$. The result follows now from the fact that optimal selectors of submodular functions can be chosen to be monotone (e.g. [21] Section 4.7). $\square$

The above theorem should be understood as follows: consider some macro-stage r and micro-stage k. There exists an optimal policy according to which the action $\mu_{j(k,r)}(v, d)$ (see the definitions in Subsection 2.3) is used at state (v, d) and the action $\mu_{j(k,r)}(v, d+e_i)$ is used at state $(v, d+e_i)$ where $i \neq j(k, r)$, and $\mu_{j(k,r)}(v, d + e_i) \geq \mu_{j(k,r)}(v, d)$.

3.2 Infinite costs

As it stands, the result is not very useful. For example, in the admission control model it would allow for the departure event always to be active, which could result in negative queue lengths. To prevent this, we need the D_v to be proper subsets of $\mathbb{N}^m$, and to show that D-submodularity still propagates.

A simple way to deal with this problem is extending D_v to $\mathbb{N}^m$ by taking $h(v,d) = \infty$ if $d \notin D_v$, and then applying Lemma 3.1. For this to yield finite costs we have to assume that non-permanent events (i.e., events for which there are (v,d) and i such that $d \in D_v$ and $d + e_i \notin D_v$) are controllable to 0 (i.e., $\underline{\mu}_i = 0$). (These terms come from [9] and [27], respectively.) This ensures that the control can be such that infinite costs states can be avoided, i.e., the optimal action for the dp operator T_i in (v,d) with $d + e_i \notin D_v$ will be $\mu = 0$ as $J_n(v, d + e_i) = \infty$. (We assume that $0 \cdot \infty = 0$.) Note that controlling the active event i at 0 is equivalent to i being nonactive, if we assume that $c_i(0) = 0$. The condition that $D_v \subset D_w$ if $\lambda_{vw} > 0$ ensures that infinite costs are avoided due to a change of state in the environment.

Furthermore, we assume that if, for some v, $d+e_i$ and $d+e_j \in D_v$, then so are d and $d + e_i + e_j$. (This is called compatible in [27], p. 208.) This ensures that if h and J_0 are submodular on D_v, then so they are on $\mathbb{N}^m$. Of course, this assumes that $c_i(0) = 0$ for all non-permanent i (this condition is missing in [9]). In fact, the above condition can be simplified, as argued in Theorem 5.2 of [9]: for all (v,d) such that $d \in D_v$, we assume that if $d+e_i, d+e_j \in D_v$, then $d + e_i + e_j \in D_v$. This is equivalent to saying that D_v is closed under maximization. In [9] it is shown that this max-closure is equivalent to both permutability and non-interruption. An event i is called non-interruptive if in case events i and j are active in $x \in D_v$, this implies that i is also active in $x + e_j \in D_v$, for all $j \neq i$.

Using Theorem 3.2, we arrive at the following.

Corollary 3.3 *If*
(i) h_i for all i and J_0 are D-submodular,
(ii) non-permanent events are controllable to 0 and have $c_i(0) = 0$,
(iii) D_v is closed under maximization,
then the optimal control of event i is increasing in d_j, for all $i \neq j$.

Often c_i is concave (in particular, linear) for some event i. It is readily seen that the optimal control in this case is either $\underline{\mu}_i$ or $\overline{\mu}_i$. This type of control, where only the extremal values of the control interval are used, is sometimes called bang-bang control. For many different control models bang-bang control is optimal, see e.g. Theorem 4 of Hinderer [12].

Example (admission control model (continued)) We apply the results obtained so far to the admission control model. To do this, we take $\underline{\mu}_2 = 0$ to assure that this non-permanent event is controllable to 0. It is readily seen that both events are non-interruptive. Therefore all that is needed to prove the submodularity of J_n is the submodularity of h and J_0. We have $\phi(d) = \phi(d + e_1 + e_2)$, this means that having the direct system costs $\tilde{h}$ convex is a sufficient condition for the existence of an optimal threshold policy. (Note that the direct costs need not be increasing.) The monotonicity results translated to the system state give that the admission probabilities decrease and the service probabilities increase as the queue length increases. As the cost for controlling the events is linear in μ, this leads to bang-bang control, i.e., at a certain threshold level the control switches from the low to the high rate or vice versa.

Let us also consider the model with batch arrivals of size B. As $\phi(d) = Bd_1 - d_2$, submodularity of h results in the following sufficient condition for $\tilde{h}$:

$$\tilde{h}(x) + \tilde{h}(x + B - 1) \leq \tilde{h}(x - 1) + \tilde{h}(x + B),\ x > 0$$

(in this example, $\phi(d + e_1) = \phi(d) + B$). Note that this condition is weaker than convexity of $\tilde{h}$.

Another generalization discussed in the previous section are arrivals modeled by a MAP (Markov arrival process, see e.g. [6]). Now $h(v, \cdot)$ needs to be submodular for every v. Note that there are no restrictions on the costs between different states of the environment. This allows for many types of different costs functions.

A generalization we did not yet discuss is a finite buffer. This generalization is direct as $\underline{\mu}_1 = 0$.

Example (tandem model (continued)) Let us apply Corollary 3.3 to the cycle of queues as studied in Weber & Stidham [27]. The service events at the queues are non-permanent, and they can indeed be controlled to 0. In [27] $c_i(0)$ need not be 0 if i is a departure event; instead of this an event is always active and the control 0 is selected if the corresponding queue is empty.

This is obviously equivalent. The max-closure is easily verified. Rewriting the submodularity in terms of the system states gives the inequalities (2) in [27]. They take as cost functions additive functions, which are convex in each component of the state space. This is a sufficient condition for the corresponding h_i to be submodular. Indeed, some algebra shows that if $\tilde{h}_i(x) = \sum_j g_j(x_j)$, then $h(d + e_i + e_j) + h(d) = h(d + e_i) + h(d + e_j)$ if $|i - j| > 1$, and $h(d + e_i + e_{i+1}) + h(d) - h(d + e_i) - h(d + e_{i+1}) = 2g_{i+1}(x_{i+1}) - g_{i+1}(x_{i+1} - 1) - g_{i+1}(x_{i+1} + 1) \leq 0$ by convexity, for $x = \phi(d)$.

Corollary 3.3 now gives us the monotonicity of the optimal control for each n. Together with results on the existence of average cost optimal policies (on which, for this specific model, is elaborated in Section 3 of [27]), we arrive at Weber & Stidham's main result on the existence of optimal monotone average costs policies ([27], p. 206).

We assumed that exogenous arrivals were part of the environment. The reason for this is that, for a reasonable type of cost function as in [27], submodularity does not hold: we expect that the control in queue j is decreasing in the number of arrivals at queue $j + 1$.

However, Weber & Stidham [27] study also a model where customers departing from queue m leave the system. Then it makes sense to include the arrivals to the first queue as an event, giving that the optimal control of each server is increasing in the number of arrivals at the first queue.

Example (fork-join queue (continued)) The main interest for the fork-join queue is the control of the servers. We can apply Corollary 3.3 if we assume that they are controllable to 0 with $c_i(0) = 0$. We take $h_i(d) = \max_{2\leq j\leq m} d_j - d_1$ (for some or all i), which corresponds to taking $\tilde{h}(x) = \max_{1\leq j\leq m-1} x_j$. It is easily checked that h_i is D-submodular and that D is closed under maximization. Thus the optimal control of a departure event is increasing in the other events. This means that an arrival increases the optimal rate, but also a departure at another queue does.

3.3 Projection

Extending the cost function from D to $\mathbb{N}^m$ has several drawbacks, one being the necessity of the assumption that non-permanent events are controllable to 0. For example, if we were in the admission control model just to control arrivals, i.e., if we would take $T_2 f(d) = h(d) + f(d + e_2)$ if $d + e_2 \in D$

and $T_2 f(d) = h(d) + f(d)$ if $d + e_2 \notin D$, then the second event is non-permanent but not controllable to 0. This problem would not exist if, in the case where the second event is controllable to zero (and $R = 0$), the optimal control would always be equal to 1. This appears to be the case if $h(d) \geq h(d + e_2)$. In the admission control model this means that we should assume that $\tilde{h}(x) \leq \tilde{h}(x+1)$, i.e., the direct costs are increasing in the queue length.

This method is called projection in [9], and it has also been applied to the control of a single queue with delayed noisy information in Altman & Koole [5]. We formalize the ideas, by first considering a model in which the non-permanent events are controllable to 0. After that we show that this model is equivalent to the one we are interested in.

Theorem 3.4 *If*
(i) h_i *for all* i *and* J_0 *are* D*-submodular,*
(ii) $h_j(v, x + e_i) \leq h_j(v, x)$ *for all non-permanent* i, j *and* x *such that* x *and* $x + e_i \in D_v$,
(iii) non-permanent events are controllable to 0 *and have* $c_i(\mu) = 0$,
(iv) D_v *is closed under maximization,*
then the optimal control of event i *is* $\overline{\mu}_i$ *if* $x + e_i \in D_v$.

Proof The conditions are a superset of those in Corollary 3.3, and thus by taking infinite costs outside the score space it follows that $S_r^{(b)} J_n$ is D-submodular. We show inductively that $S_r^{(b)} J_n(v, d + e_i) \leq S_r^{(b)} J_n(v, d)$ for i non-permanent and for all $d, d + e_i \in D_v$. Thus assume that f satisfies all conditions. We show first that $T_k f(v, d + e_i) \leq T_k f(v, d)$ if $d, d + e_i \in D_v$. This follows easily if both $d + e_i + e_k$ and $d + e_k \in D_v$, or if $d + e_k \notin D_v$. Now assume that $d + e_i + e_k \notin D_v$. By (iv) also $d + e_k \notin D_v$, and the inequality still holds. Finally we show $T_i f(v, d + e_i) \leq T_i f(v, d)$. This follows easily. ▫

This Theorem states that if we have a non-permanent event which is not controllable to 0, and if $h_k(v, d + e_i) \leq h_k(v, d)$ for all k and appropriate d, then we might as well make the event controllable to 0 as the event will always be controlled at the highest rate possible. This leads to the following.

Corollary 3.5 *If*
(i) h_i *for all* i *and* J_0 *are* D*-submodular,*
(ii) $h_k(v, d + e_i) \leq h_k(v, d)$ *for all* k, d, $d + e_i \in D_v$, *and* $c_i(\mu) = 0$ *for all*

non-permanent i,
(iii) D_v is closed under maximization,
then the optimal control of event i is increasing in d_k, for all $i \neq k$.

Note that the condition on the c_i cannot be found in [9]. Of course the Corollaries 3.3 and 3.5 can be combined in a single model.

Example (admission control model (continued)) As is argued above, if the departure process is non-controllable, the condition that the direct costs are increasing makes that the monotonicity result still holds. This model is thus an example where the Corollaries 3.3 and 3.5 are combined: Corollary 3.3 is used for the arrival event, Corollary 3.5 is used for the departure event.

Remark Besides taking infinite costs outside the state space and projection other methods are possible to deal with the boundary. An example of such a model can be found in [18]. There a tandem model with finite buffers is studied. The costs for rejecting a customer are equal to 1, and these are the only costs in the system (however, this implies implicitly costs for queues that are full, since the controller is forced to reject an arriving customer when the queue is full). The boundary is dealt with by including, besides submodularity, inequalities (formulated in the system space) of the form $\tilde{f}(x+e_i) \leq 1 + \tilde{f}(x)$.

3.4 Admission control model with random batches

In the admission control model we considered a generalization to batch arrivals. This could be further analyzed by assuming that each batch has a random size. However, this would mean that ϕ becomes a random function, and D would depend on its realization. Thus we cannot apply the theory developed above directly.

An elegant solution is to assume that d_1 does not count the number of batches, but the number of customers that have arrived. If we assume that a batch of k customers has probability β_k, this would result in

$$T_1 f(d) = \min_{0 \leq \mu \leq \overline{\mu}_1} \{(1-\mu)K + \mu \sum_k \beta_k f(d+ke_1) + (1-\mu) f(d)\}.$$

Instead of submodularity, we show that the following inequality propagates:

$$f(d) + \sum_k \beta_k f(d + ke_1 + e_2) \le \sum_k \beta_k f(d + ke_1) + f(d + e_2).$$

Copying the proof of Lemma 3.1 for the current case is easily done. As the arrival event is permanent, no complications arise due to boundary issues: taking infinite costs outside $D = \{d_1 \ge d_2\}$ retains submodularity. With regard to the costs in the system state, the situation is similar to that in the case of fixed batch sizes. The sufficient condition on $\tilde{h}$ is as follows:

$$\tilde{h}(x) + \sum_k \beta_k \tilde{h}(x + k - 1) \le \tilde{h}(x - 1) + \sum_k \beta_k \tilde{h}(x + k), \ x > 0.$$

Again convexity of $\tilde{h}$ implies the above inequality.

Note that if the departures cannot be controlled we have to assume again that the costs are monotone.

3.5 Convexity and stationarity

Above we showed that the rate at which an event α should be controlled increases as other events have occurred more often. One would conjecture that (under certain conditions) the reverse holds for the event itself, i.e., that the optimal control for an event α is decreasing in d_α. This would mean proving convexity in d_α. However, we were not able to prove convexity of J_n in one or all components.

In [27] however an argument is given showing that, under stationary conditions, the optimal control of event i is decreasing in d_i for all i. As it applies also to our other examples, we give it here.

We consider problems with an infinite horizon cost. First note that there exist discounted and average optimal policies that are stationary, i.e., they depend only on the state (and on the micro-stage), and not on the time (i.e., on the macro-stage). This means that the rate at which event i should be controlled is increased if another event j occurs. However, in the cycle of queues, the same system state is reached if all transitions have fired once. This means that also the optimal control of event i should be the same as before the firing of all events. As the optimal rate increased with the firing of each event except event i itself, this means that the optimal control of event i should be decreasing in d_i.

The same applies to the admission control model and the fork-join queue. Note that in the case of batch arrivals of size B the other event(s) have to fire B times each to reach the same system state.

4 GAMES

In this section we restrict ourselves to two events, where in the second event the minimization is replaced by maximization (we have not been able to obtain a general theory for more than two controlled events). We thus consider the same value function as in (2), where the minimization in (1) is replaced by maximization for $i = 2$. Let T_1 be the minimizing operator, and T_2 the maximizing operator.

This type of dynamic programming equation is used in stochastic (or Markov) games (see [24]) having two players, each one controlling a different event, both having the same objective function, which is maximized by one and minimized by the other, i.e., a zero sum setting. Another application of such dynamic programming appears in the context of (non-controlled) discrete event systems, and is used for the performance evaluation of parallel processing with both "or" and "and" constraints [17].

Now we show that in this setting submodularity still propagates, first for the unconstrained case. The reason for considering only two events is that we cannot prove the Lemma for $m > 2$.

Lemma 4.1 *If h_i for all i and J_0 are D-submodular, then $S_r^{(b)} J_n$ is also D-submodular for all n, r and b.*

Proof Propagating T_1 is similar to Lemma 3.1. Thus consider T_2. Denote with μ_1 (μ_2) the maximizing μ for score d ($d + e_i + e_j$). First assume that $\mu_1 \geq \mu_2$. Summing

$$h_2(d) + h_2(d + e_1 + e_2) \leq h_2(d + e_1) + h_2(d + e_2),$$

$$c_2(\mu_1) + c_2(\mu_2) \leq c_2(\mu_1) + c_2(\mu_2),$$

$$\mu_2 f(d + e_2) + \mu_2 f(d + e_1 + 2e_2) \leq \mu_2 f(d + e_1 + e_2) + \mu_2 f(d + 2e_2),$$

$$(\mu_1 - \mu_2) f(d + e_2) + (\mu_1 - \mu_2) f(d + e_1 + e_2) \leq$$
$$(\mu_1 - \mu_2) f(d + e_2) + (\mu_1 - \mu_2) f(d + e_1 + e_2),$$

and

$$(1-\mu_1)f(d)+(1-\mu_1)f(d+e_1+e_2) \le (1-\mu_1)f(d+e_1)+(1-\mu_1)f(d+e_2)$$

gives

$$T_2f(d)+T_2f(d+e_1+e_2) \le T_2f(d+e_1)+T_2f(d+e_2),$$

where we took μ_1 as (suboptimal) action in $d+e_1$ and μ_2 in $d+e_2$.

If $\mu_1 \le \mu_2$, we sum

$$\mu_1 f(d+e_2)+\mu_1 f(d+e_1+2e_2) \le \mu_1 f(d+e_1+e_2)+\mu_1 f(d+2e_2),$$

$$(\mu_2-\mu_1)f(d)+(\mu_2-\mu_1)f(d+e_1+e_2) \le (\mu_2-\mu_1)f(d+e_1)+(\mu_2-\mu_1)f(d+e_2),$$

$$(\mu_2-\mu_1)f(d+e_2)+(\mu_2-\mu_1)f(d+e_1+2e_2) \le \\ (\mu_2-\mu_1)f(d+e_1+e_2)+(\mu_2-\mu_1)f(d+2e_2),$$

and

$$(1-\mu_2)f(d)+(1-\mu_2)f(d+e_1+e_2) \le (1-\mu_2)f(d+e_1)+(1-\mu_2)f(d+e_2),$$

together with the inequalities for h_2 and c_2. □

From this Lemma the monotonicity of the optimal policy for the unbounded case follows. Because the maximizing actions are chosen in T_2, the optimal control for this operator is decreasing in d_1.

Theorem 4.2 *If h_i for all i and J_0 are D-submodular, then the optimal control of event* 1 *is increasing in d_2, and the optimal control of event* 2 *is decreasing in d_1.*

To deal with the boundary, we cannot immediately apply Corollary 3.3 or 3.5, as the maximization does not avoid ∞-cost states. This calls for costs $-\infty$ outside the state space. However, to maintain submodularity, we would have to replace the max-closure condition. We will not investigate this further, as in the example below the maximizing operator is permanent.

Although the setting in this section is that of a game, the optimal policies are non-randomized as the decisions do not occur simultaneously. If they would then the dynamic programming equation would have a "value" operator instead of a min and max operator (see [1, 2, 3]).

Example (admission control model (continued)) We consider here our admission control model, but with T_1 the departure event and T_2 the arrival event. Note that T_2, the maximizing event, is permanent. This model can be seen as a queue with controlled service which is operated under worst case conditions. Intuitively, under worse conditions, customers arrive if the queue is already full. This is indeed what follows from Theorem 4.2. Thus, if c_2 is linear, there is a threshold (possibly 0) such that arrivals are generated at the lowest rate below the threshold, and at the highest rate above it.

References

[1] E. Altman. Flow control using the theory of zero-sum Markov games. *IEEE Transactions on Automatic Control*, 39:814–818, 1994.

[2] E. Altman. Monotonicity of optimal policies in a zero sum game: A flow control model. *Advances of Dynamic Games and Applications*, pages 269–286, 1994.

[3] E. Altman and A. Hordijk. Zero-sum Markov games and worst-case optimal control of queueing systems. *QUESTA*, 21:415–447, 1995.

[4] E. Altman and G.M. Koole. Stochastic scheduling games and Markov decision arrival processes. *Computers and Mathematics with Applications*, 26(6):141–148, 1993.

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

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

[7] F. Baccelli and B. Gaujal. Stationary regime and stability of free-choice Petri nets. In *Proceedings of the 11th International Conference on Analysis and Optimization of Systems*, 1994.

[8] W.H. Fleming and H.M. Soner. *Controlled Markov Processes and Viscosity Solutions*. Springer-Verlag, 1993.

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

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

[11] P. Glasserman and D.D. Yao. Addendum to "Monotone optimal control of permutable GSMPs". In preparation, 1995.

[12] K.F. Hinderer. On the structure of solutions of stochastic dynamic programs. In *Proceedings of the 7th Conference on Probability Theory*, pages 173–182, Brasov, Romania, 1984.

[13] K.F. Hinderer and M. Stieglitz. On polychotomous search problems. *European Journal of Operations Research*, 73:279–294, 1994.

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

[15] 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*, 25:979–996, 1993.

[16] K. Hsu and S.I. Marcus. Decentralized control of finite state Markov processes. In *Proceedings of the 19th IEEE Conference on Decision and Control*, pages 143–148, 1980.

[17] A. Jean-Marie and G.J. Olsder. Analysis of stochastic min-max systems: Results and conjectures. Technical Report 93-94, Delft University of Technology, 1993.

[18] G.M. Koole and Z. Liu. Nonconservative service for minimizing cell loss in ATM networks. In *Proceedings of the 1st Workshop on ATM Traffic Management*, Paris, 1995.

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

[20] P. Naor. On the regulation of queueing size by levying tolls. *Econometrica*, 37:15–24, 1969.

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

[22] F.C. Schoute. Decentralized control in packet switched satellite communication. *IEEE Transactions on Automatic Control*, 23:362–371, 1978.

[23] R.F. Serfozo. An equivalence between continuous and discrete time Markov decision processes. *Operations Research*, 27:616–620, 1979.

[24] L.S. Shapley. Stochastic games. *Proceedings National Academy of Science USA*, 39:1095–1100, 1953.

[25] S. Stidham, Jr. Socially and individually optimal control of arrivals to a $GI|M|1$ queue. *Management Science*, 24:1598–1610, 1970.

[26] D.M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26:305–321, 1978.

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

[28] J. Weishaupt. Optimal myopic policies and index policies for stochastic scheduling problems. *ZOR - Mathematical Methods of Operations Research*, 40:75–89, 1994.

[29] U. Yechiali. On optimal balking rules and toll charges in a $GI|M|1$ queueing process. *Operations Research*, 19:349–370, 1971.