STRUCTURAL PROPERTIES OF A CLASS OF ROBUST INVENTORY AND QUEUEING CONTROL PROBLEMS

Zeynep Turgay, Fikri Karaesmen*, and E.Lerzan Örmeci

Department of Industrial Engineering
Koç University
34450, Sarıyer, İstanbul, TURKEY
zturgay@ku.edu.tr, fkaraesmen@ku.edu.tr, lormeci@ku.edu.tr
* Corresponding author

February 2015
Structural Properties of a Class of Robust Inventory and Queueing Control Problems

July 22, 2015

Abstract

In standard stochastic dynamic programming, the transition probability distributions of the underlying Markov Chains are assumed to be known with certainty. We focus on the case where the transition probabilities or other input data are uncertain. Robust dynamic programming addresses this problem by defining a min-max game between nature and the controller. Considering examples from inventory and queueing control, we examine the structure of such robust dynamic programs on the structure of the optimal policy when event probabilities are uncertain. We identify the cases where certain monotonicity results still hold and the form of the optimal policy is determined by a threshold. We also investigate the marginal value of time and the case of uncertain rewards.

Keywords: stochastic dynamic programming, robust optimization, inventory control, queueing control

1 Introduction

In many practical optimization problems, the input parameters to the problem are not known with certainty; rather they are either estimated from the existing data or the possible values that they can take can be specified by expert opinions. This uncertainty, if ignored, may cause significant suboptimality or infeasibility for the solution considered. Robust optimization is a specific methodology that addresses this problem and has received much attention lately (see Ben Tal, El Ghaoui and Nemirovskii (2009) for a comprehensive presentation).
Our focus in this paper is on a set of stochastic dynamic problems from inventory and queueing theory where the transition probability distributions of the underlying Markov Chains may be uncertain. The models considered include discrete-time versions of some well-established cases such as service rate control and admission control problems of Lippman (1975), the stock rationing problem of Ha (1997) and the dynamic revenue management problem of Lautenbacher and Stidham (1999).

To address the uncertainty on the probability distributions in such problems we formulate a robust stochastic dynamic program with a maximin approach. This approach defines a game between the controller (system manager) and Nature. For instance, in the context of demand admission control, the controller’s aim is to maximize the expected profit by choosing the allowable actions (for example by admitting a given class of demand or not), whereas Nature tries to minimize the controller’s expected profit by choosing the worst-possible parameters (for example arrival probabilities for different classes) and acts upon observing the controller’s choice. This formulation is known as the robust counterpart of the standard problem. The robust optimal policy designates the policy which yields the highest expected profit result after minimization by Nature. In this formulation, the controller acts upon the worst-case scenario, which does not always happen. Consequently, his actions can be labeled as the most conservative with respect to maximizing the expected revenue. To overcome this problem, semi-robust MDPs are developed, which allows the controller to have varying degrees of conservatism.

The robust formulation of a Markov Decision Process (MDP) with an uncertain transition probability distribution goes back to Satia and Lave (1973) who propose a solution by a policy iteration approach. White and Eldeib (1994) present value iteration based numerical algorithms and bounds for polyhedral uncertainty sets on transition probabilities. Bagnell, Ng and Schneider (2001) propose a robust value iteration and discuss the related computational complexity issues. Nilim and El Ghaoui (2005) and Iyengar (2005) simultaneously study robust stochastic dynamic programs and establish the existence of a robust Bellman recursion whose solution yields the robust value function and the corresponding optimal policy. In addition, both papers emphasize that under appropriate choice of uncertainty sets, the additional complexity brought by the robust formulation is reasonable when the standard formulation has a tractable solution. Paschalidis and Kang (2008) explore the effect of using stronger uncertainty set formulations and investigate a specific queueing control problem. Delage and Mannor (2010) consider chance-constrained MDPs with uncertain parameters and show that some important instances of this class of problems is computationally
tractable. Kardeş, Ordóñez, and Hall (2011) investigate infinite horizon discounted stochastic games with uncertain transition probabilities, and establish the existence of equilibria and propose a method for computing the equilibria. Finally, Xu and Mannor (2012) propose a more sophisticated model of parameter uncertainty which allows multiple nested uncertainty sets.

In this paper, we mainly focus on investigating the structural properties of optimal policies in a class of robust and semi-robust MDPs. The queueing and inventory control literature has a strong tradition in exploring the structure of optimal policies. This is in part due to the computational efficiency of structured policies. However, the main motivation for looking for structured policies is that they are usually expressed in a few parameters and tend to be easy to understand and implement. There are known effective techniques to investigate the structure of the solution of a stochastic dynamic program. Event-based dynamic programming (EBDP) proposed by Koole (1998) and further extended in Koole (2006) streamlines this procedure for a class of queueing control problems. Recently, Çil, Örmeci and Karaesmen (2009) employ EBDP to explore structural properties in a class of production/inventory problems and propose an extension to study the effects of perturbations of input parameters. In this paper, we employ event-based dynamic programming to formulate the robust MDPs and investigate the structure of their optimal policies.

There are also a number of recent papers that investigate specific robust MDP problems in queueing or inventory control. Birbil, Frenk, Gromicho and Zhang (2009) consider a dynamic revenue management problem and show that there is a very efficient solution under an ellipsoidal models of uncertainty. Turgay, Karaesmen and Örmeci (2014) investigate a similar problem with interval and polyhedral uncertainty sets and establish monotonicity results for nested uncertainty sets. Lim and Shanthikumar (2007), and Jain, Lim and Shanthikumar (2010) consider robust versions of dynamic pricing problems for queueing systems using entropy-based models of uncertainty. Rusmevichientong and Topaloglu (2012) explore a robust assortment optimization problem and show that optimal assortment policy can be computed efficiently by giving a complete characterization of the optimal policy. All of these papers obtain results on the structure of optimal policies by exploiting certain properties of the uncertainty set. In contrast, we consider general models in the EBDP framework with general uncertainty sets to establish structural results that hold for a class of problems.

As the literature shows, the robust dynamic programming problems have received a great interest recently. Our contributions in this area can be summarized as follows: (1) Our results on robust MDPs apply to the EBDP framework with general uncertainty sets on the transition prob-
abilities, (2) Our results extend to the semi-robust MDP models with general uncertainty sets on
the transition probabilities, (3) A discussion of the robust MDPs with general uncertainty sets on
other parameters is included, where the structure of optimal policy in a specific system is derived.
By (1), our results are valid for all models that can be generated within the EBDP framework in
addition to the models considered in this paper, whereas (2) allows us to compare the performances
of controllers with different levels of conservatism in a computationally efficient way. By (3), we
observe the difficulties for the robust MDP models when the other parameters are uncertain.

The organization of the paper is as follows; in section 4 we present the event-based dynamic
programming (EBDP) framework for the nominal MDPs, while section 3 defines the robust coun-
terparts of the nominal MDPs. Section 4 establishes the structural properties of both nominal and
robust MDPs under certain conditions on the underlying operators that form the EBDP framework.
In section 1, we introduce the operators commonly used in literature within EBDP framework, and
show that they all have the desired properties that guarantee certain structural properties of the
underlying value functions. Section 6 presents models analyzed in literature that fall into the EBDP
framework. The purpose here is two-fold: (1) to illustrate the intriguing effects of our results, and
(2) to show the generality of our results. Section 7 extends our results when the uncertainty on
the parameters are represented by two or more uncertainty sets, rather than one as in the regular
robust definition. However, we also show that we cannot show monotonicity of the value functions
with respect to nested uncertainty sets. Finally, we conclude in the last section.

2 Nominal Event-Based Dynamic Programming Framework

In this section, we describe the event-based dynamic programming framework introduced by Koole
(1998), which provides the basis of our analysis. The problems we consider in the scope of this paper
are represented in discrete time. In our setting, the stages (time) are denoted by \( t = 0, 1, ..., T \),
where \( T \) is the last stage in the horizon. Let \( i = 0, 1, 2, ..., n \) denote the event indices, where events
1, 2, ..., \( n \) are events which may lead to a state transition (if the appropriate action is selected)
while event 0 corresponds to a fictitious event where no observable real event (and therefore no
state transition) occurs. The system state \( x \) can take values in set \( X \) (i.e., \( x \in X \)) at any stage \( t \),
where \( X \) is a subset of integers. We let \( a_i \) denote the controller action regarding event \( i \), so that an
action can be defined by \( a_i \in \{0, 1\} \) as in the case admission/rejection or by \( a_i \in \mathbb{R} \) as in the case
of pricing, where \( \mathbb{R} \) is the set of real numbers. Note that the controller is allowed to choose her/his
actions independently for all \((x, t)\) pairs, and there is no restriction among the actions regarding different events. The action vector \(a = (a_1, a_2, ..., a_n)\) denotes the actions for all possible events. At each stage, depending on the controller action, \(R_i(a, x)\) \((C_i(a, x))\) is gained (incurred) as an immediate reward (cost). The randomness is characterized by a transition probability distribution at each stage which is assumed to be independent of prior uncertainties. The probability that event \(i\) occurs at stage \(t\) is given by \(p_{i,t}\), so \(\sum_{i=0}^{n} p_{i,t} = 1\) for all \(t\). When event \(i\) occurs in state \(x\) at stage \(t\), the conditional probability that the next state is \(y\) if controller selects action \(a\) is denoted by \(q_{t}(a, x, y|i)\), hence \(\sum_{y} q_{t}(a, x, y|i) = 1\) for all \(t, a, x\) and \(i\). We express the optimal value function \(v_t(x)\) according to the following equation when the time left until the end of the horizon is \(T - t\):

\[
v_t(x) = \max_a \sum_{i=0}^{n} p_{i,t} \sum_{y \in X} q_{t}(a_i, x, y|i)(v_{t+1}(y) + R_i(a_i, x)).
\]

Note that the action vector \(a\) is a function of \(x\) and \(t\), but we suppress this for notational simplicity.

The event-based approach allows us to define the value function \(v_t(x)\) as a convex combination of operators, \(T_i\), where \(T_i\) is defined as follows:

\[
T_i v_{t+1}(x) = \max_a \sum_{y \in X} q_{t}(a, x, y|i)(v_{t+1}(y) + R_i(a, x)),
\]

hence \(v_t(x)\) can be written as:

\[
v_t(x) = \sum_{i=0}^{n} p_{i,t} \max_a \sum_{y \in X} q_{t}(a, x, y|i)(v_{t+1}(y) + R_i(a, x)) = \sum_{i=0}^{n} p_{i,t} T_i v_{t+1}(x).
\]

### 3 Robust Counterpart of Nominal Framework

This section introduces a robust version of the dynamic program (which will be referred to as the nominal problem) described in Section 2. The robust formulation assumes that a subset of the problem parameters is uncertain. In this section, we assume that the transition probability distributions are uncertain. Typically, the system controller decides on his actions before observing the uncertain parameters. Once his decisions are taken, Nature selects these parameters from an uncertainty set in order to minimize the expected profit of the system. Hence, the controller has to consider the worst case scenario in terms of the transition probability distribution when choosing his actions. Our robust formulations are based on the maximin approach suggested by Nilim and El Ghaoui (1995) and Iyengar (1995).

In the robust formulation, the transition probabilities belong to an uncertainty set, rather than being fixed values as in the nominal problem. In the EBDP approach, the transition probabilities
consist of two components as observed in Equation (1): $q_t(a, x, y|i)$ and $p_{i,t}$. We assume that $q_t(a, x, y|i)$ is known with certainty and $p_{i,t}$ is uncertain and can depend on the chosen action and the system state. This may be a limitation for certain models but is appropriate for queueing and inventory control problems represented by EBDP. In such models, generally $q_t(a, x, y|i)$ is either 0 or 1, and the probabilistic structure is only reflected through the event probability $p_{i,t}$. Hence, $q_t(a, x, y|i)$ can be omitted from the definition of the operator. One exception is dynamic pricing operator where $0 \leq q_t(a, x, y|i) \leq 1$.

We let $p_{i,t}(x, a)$ be the probability of observing event $i$ when the system is in state $x$ and action vector $a$ is chosen at stage $t$. Note that $a = (a_1, \ldots, a_n)$, so $a$ specifies the actions for all possible events. Naturally, $a$ depends on state $x$ and stage $t$. However, as in the previous section, we suppress this dependency for notational simplicity. We assume that $p_{i,t}(x, a)$ belongs to an uncertainty set $P_t$, where $P_t$ represents the available information on the event probability distribution. Our uncertainty model is based on the model proposed by Nilim and El Ghaoui (2005), where the so-called rectangularity property is the main condition for obtaining a recursive solution. When the rectangularity property holds, nature can independently select its action for every stage, state and the controller action. This assumption essentially implies that Nature’s choice of a particular distribution at time $t$ or in state $x$ does not limit the choices of Nature in the future or in other states. In addition to this basic assumption, we also assume that the uncertainty set at each stage $t$, $P_t$, is independent of the controller’s action vector $a$ as well as the state $x$. Note that this additional assumption is not very restrictive when handling queueing/inventory problems: The state-dependent event probability distribution is difficult to estimate from limited statistical data. Therefore, it is natural to assume that uncertainty sets depend on common global estimates rather than on state-dependent estimates. Moreover, in typical examples, these probabilities represent demand or processing rates which do not depend on the state of the system or the actions taken.

Now, we let $w_t(x)$ be the robust counterpart of the value function given in (2), and consider the following robust DP equation:

$$w_t(x) = \max_a \left\{ \min_{p_t(x,a)} \sum_{i=0}^n p_{i,t}(x, a) \sum_{y \in X} q_t(a, x, y|i)(w_{t+1}(y) + R_i(a, x)) \right\},$$

where $p_t(x, a) = (p_{1,t}(x, a), p_{2,t}(x, a), \ldots, p_{n,t}(x, a))$. This problem can be solved recursively due to the above assumptions, as shown by Nilim and El Ghaoui (2005) and Iyengar (2005).

Our next result, Theorem 1, shows that the optimal action at any stage is independent of the choice of event probability distribution at that stage. This allows to change the order of
maximization and minimization in equation (3). Then the robust value functions can be presented more similarly to the nominal value functions, as shown in the second part of Theorem 1. In Section 4.2, this result will play a key role in extending the structural properties of the nominal MDPs to their corresponding robust MDPs.

**Theorem 1**

1. The optimal policy of the controller is independent of the Nature’s posteriori decision.

2. The robust value function $w_t(x)$ can be expressed as:

$$w_t(x) = \min_{p_t(x)} \left\{ \sum_{i=0}^{n} p_{i,t}(x) T_i w_{t+1}(x) \right\}.$$  \hspace{1cm} (4)

**Proof:**

1. Let $p_t = (p_{1,t}, \ldots, p_{n,t})$ be any transition probability distribution in uncertainty set $\mathcal{P}_t$. If Nature decides to use $p$ before observing the controller’s action, the controller’s optimal policy is determined by the solution of the following problem:

$$\max_{a} \sum_{i=0}^{n} p_{i,t} \sum_{y \in X} q_t(a, x, y|i)(w_{t+1}(y) + R_i(a, x)),$$  \hspace{1cm} (5)

by equation (3). We first observe that the optimal action corresponding to an event does not have any effect on the optimal actions of other events. Then, it is clear that the controller’s optimal decision is unaffected by the Nature’s choice of $p_t$. This allows us to change the order of maximization and summation in (5):

$$\sum_{i=0}^{n} p_{i,t} \max_{a} \sum_{y \in X} q_t(a, x, y|i)(w_{t+1}(y) + R_i(a, x)),$$  \hspace{1cm} (6)

which can then be written in terms of operators $T_i$’s as follows:

$$\sum_{i=0}^{n} p_{i,t} T_i w_{t+1}(x).$$  \hspace{1cm} (7)

The optimal actions of the controller are the outputs of the operators, which are the same for each possible choice of $p_t = (p_{1,t}, \ldots, p_{n,t})$. Hence, we can conclude that the optimal policy of the controller is independent of the Nature’s posteriori decision.

2. Let $a_t^*(x) = a^* = (a_1^*, \ldots, a_n^*)$ be the optimal action of the controller found as a result of the operators. From part (1), we know that $a_t^*(x)$ determines the controller’s optimal policy, independent of the Nature’s posteriori decision. Then, in equation (3), it is enough for Nature to find the action which minimizes the expected revenue of the controller for $a_t^*(x)$ only. Hence,
equation (3) can be written as:

\[ w_t(x) = \min_{p_t(x,a^*_t)} \sum_{i=0}^{n} p_{i,t}(x,a^*_t) \sum_{y \in X} q_t(a^*_i, x, y|i)(w_{t+1}(y) + R_i(a^*_i, x)), \]  

(8)

But then we have

\[ T_i w_{t+1}(x) = \sum_{y \in X} q_t(a^*_i, x, y|i)(w_{t+1}(y) + R_i(a^*_i, x)) \] by definition. Moreover, \( a^*_i(x) \) is a function of \( x \) and \( t \), so that the original action-dependent probability distribution, \( p_{i,t}(x,a) \) in (3) can be replaced by \( p_{i,t}(x) \). Accordingly, equation (8) can be expressed as follows:

\[ w_t(x) = \min_{p_t(x)} \sum_{i=0}^{n} p_{i,t}(x) T_i w_{t+1}(x). \]  

(9)

By Theorem 1, we establish the independence of the controller’s action from the Nature’s posteriori action for all problems that can be constructed within EBDP framework, which enables us to define the optimality equation as in (4). Consequently, it is enough to solve the inner problem of nature only for one action of the controller, instead of solving it for all possible actions of the controller, at each stage and state. This improves the solution time significantly as the controller’s action set gets larger.

4 Structural Properties for Nominal and Robust MDPs

In this section we first present the structural properties of the nominal MDPs, then extend these to the robust MDPs within the EBDP framework. For this purpose, we first define the properties of the value functions that can guarantee the existence of monotone optimal policies. In an EBDP framework, if the operators preserve these properties, then the value functions will also have them due to the construction of EBDP and the value iteration algorithm.

4.1 Structural Properties for Nominal MDPs

We are interested in the following properties of the value function: (1) Non-increasingness (non-decreasingness) in \( x \) refers to \( v_t(x) \leq (\geq) v_t(x - 1) \) for all \( x \geq 1 \) and all \( t \), and (2) concavity (convexity) in \( x \) refers to: \( v_t(x + 1) - v_t(x) \leq (\geq) v_t(x) - v_t(x - 1) \) for all \( x \geq 1 \) and all \( t \). Concavity and monotonicity properties of the value functions determine the structure of the optimal policies as well, e.g., concavity of the value function leads to the optimality of threshold policies for admission control and optimal base stock policies for inventory control.

Next, we consider the supermodularity/submodularity of the value functions in \( x \) and \( t \), which has not been studied in this perspective to our knowledge: Supermodularity (submodularity) (in
(x, t)) refers to \( v_t(x) - v_t(x - 1) \geq (\leq) v_{t+1}(x) - v_{t+1}(x - 1) \) for all \( x \geq 1 \) and all \( t \). These properties ensure that the parameters of the structured optimal policies are also monotone in time, e.g., the optimal base stock levels and optimal thresholds are monotone in \( t \). The preservation of supermodularity is related to the marginal benefit (MB) of the operator, which is first defined by Çil et al. (2009). Marginal benefit of an operator indicates the difference of the value functions between the two systems, where one system observes the event that corresponds to the operator and the other system remains in the same state. Hence, the MB of the operator, \( B_i \), is given by:

\[
B_i v(x) = T_i v(x) - v(x). 
\] (10)

Then, equation (2) can be written in terms of \( B_i \) as follows:

\[
v_t(x) = \sum_{i=0}^{n} p_{i,t} B_i v_{t+1}(x) + v_{t+1}(x), \quad (11)
\]

since \( \sum_{i=0}^{n} p_{i,t} = 1 \) for all \( t \). This representation of the value function sets a direct relationship between the MB function and the supermodularity/submodularity properties in the context of these systems. If the MB function of an operator \( T_i, B_i v_t(x) \), is non-increasing (non-decreasing) in \( x \), then the submodularity (supermodularity) in \( x \) and \( t \) is preserved by the corresponding operator, \( T_i v_t(x) \).

To summarize, we are interested in the operators which preserve the following properties: non-increasingness (NI), non-decreasingness (ND), concavity (C). Moreover, the MB functions of the operators should be either non-increasing (MB-NI) or non-decreasing (MB-ND). Then, we can use the event-based dynamic programming framework, to constitute a model with certain structural properties:

**Theorem 2**  
(i) If a nominal value function, \( v_t(x) \), can be represented as a convex combination of operators with properties NI (ND) and C, then \( v_t(x) \) is NI (ND) and C.

(ii) Furthermore, if all the operators that constitute the nominal value function have property MB-ND (MB-NI), then \( v_t(x) \) is supermodular (submodular) in \( x, t \).

**Proof:** The proof of Part (i) follows from the definition of NI (ND) and C by induction in the value iteration algorithm to solve the MDPs. Here, we only provide the proof of part (ii) when all the operators have property MB-ND, i.e., \( B_i v_t(x) \) is ND in \( x \) for all \( t \), since the other case is similar. Then, we have:

\[
v_t(x) - v_{t+1}(x) = \sum_{i=0}^{n} p_{i,t} B_i v_{t+1}(x) \geq \sum_{i=0}^{n} p_{i,t} B_i v_{t+1}(x - 1) = v_t(x - 1) - v_{t+1}(x - 1), \quad (12)
\]
where the first and last equalities are due to equation (11) and the inequality follows from the assumption that MB functions, \( B_i v_1(x) \), are ND in \( x \) for all \( t \). This completes the proof.

**Remark:** Theorem 2 establishes concavity under fairly general conditions, so that threshold policies are optimal for all models that can be represented by a convex combination of the operators above. It also ensures the supermodularity (submodularity) property for a general class of models under the limitation that all the operators of a given model should have ND (NI) MB functions. However, this limitation turns out to be more restrictive and problem-dependent. For instance, it is not natural to conceive a queueing system that consists of operators whose MB functions are all NI (or ND). By nature, queueing systems do not possess this property (see the queueing operators in Section 1). There are, however, plausible inventory systems that support the property.

### 4.2 Structural Properties for Robust MDPs

In this section, we show that the robust value functions have similar properties as nominal value functions for the problems of interest, regardless of the shape of the uncertainty set. Our main result extends Theorem 2 to the robust counterparts of the nominal problems under consideration. More explicitly, for any problem that can be expressed in the form of Equation (2), all the structural results given in Theorem 2 extend to the value function of the robust counterpart, \( w_t(x) \).

**Theorem 3**  
(i) If a robust value function, \( w_t(x) \), can be represented as a convex combination of operators with properties NI (ND) and C, then \( v_t(x) \) is NI (ND) and C.

(ii) Furthermore, if all the operators that constitute the robust value function have property MB-ND (MB-NI), then \( w_t(x) \) is supermodular (submodular) in \( x, t \).

**Proof:** Let \( p^*_t(x-1), p^*_t(x), p^*_t(x+1) \) denote the optimal choices of Nature in states \( x-1, x \) and \( x+1 \), respectively, at stage \( t \), in the rest of the proof.

We first establish concavity. Assume that \( w_{t+1}(x) \) is concave in \( x \) at stage \( t+1 \). By assumption, all operators that constitute the value function preserve concavity in \( x \). Then the convex combination of these operators also preserve this property. Hence, we can write the following inequality:

\[
\sum_i p^*_t(x)T_i w_{t+1}(x) + \sum_i p^*_t(x)T_i w_{t+1}(x+1) \leq 2 \left[ \sum_i p^*_t(x)T_i w_{t+1}(x) \right].
\]  

(13)
We know that:

\[ \sum_i p_{i,t}^*(x-1)T_iw_{t+1}(x-1) \leq \sum_i p_{i,t}^*(x)T_iw_{t+1}(x-1), \]

by the optimality of \( p_{i,t}^*(x) \) in state \( x-1 \), and:

\[ \sum_i p_{i,t}^*(x+1)T_iw_{t+1}(x+1) \leq \sum_i p_{i,t}^*(x)T_iw_{t+1}(x+1), \]

by the optimality of \( p_{i,t}^*(x+1) \) in state \( x+1 \). Then, the sum of the left-hand-sides (LHSs) of these two inequalities is less than the LHS of inequality (13), which proves that \( w_t(x) \) is concave in \( x \) at stage \( t \).

Next, we prove that the value function is NI for queueing operators. The result for the inventory operators can be proven similarly. We suppose that \( w_{t+1}(x) \) is NI in \( x \). Then \( T_iw_{t+1}(x) \) is NI in \( x \), since \( T_i \) is one of the operators defined in Table 1. Hence, we have:

\[ \sum_i p_{i,t}^*(x)T_iw_{t+1}(x) \leq \sum_i p_{i,t}^*(x-1)T_iw_{t+1}(x) \leq \sum_i p_{i,t}^*(x-1)T_iw_{t+1}(x-1), \]

where the first inequality is due to the optimality of \( p_{i,t}^*(x) \) in state \( x \), and the second inequality follows since \( T_iw_{t+1}(x) \) is NI in \( x \). This completes the proof of Part 1.

We prove part (2) only for the case when MB functions of all the operators are ND, since the proof of the other case is similar. First, let us note that by equations (11) and (9), we have:

\[ w_t(x) - w_{t+1}(x) = \sum_i p_{i,t}^*(x)B_iw_{t+1}(x). \]

On the other hand:

\[ \sum_i p_{i,t}^*(x)B_iw_{t+1}(x) \geq \sum_i p_{i,t}^*(x-1)B_iw_{t+1}(x-1) \geq \sum_i p_{i,t}^*(x-1)B_iw_{t+1}(x-1), \]

where the first inequality follows since \( B_iw_{t+1}(x) \) is ND in \( x \), and the second inequality is due to the optimality of \( p_{i,t}^*(x-1) \) in state \( x-1 \), and. Therefore:

\[ w_t(x) - w_{t+1}(x) = \sum_i p_{i,t}^*(x)B_iw_{t+1}(x) \geq \sum_i p_{i,t}^*(x-1)B_iw_{t+1}(x-1) = w_t(x-1) - w_{t+1}(x-1). \]

This completes the proof. \( \square \)

Theorem 3 establishes a general result: the robust value function \( w_t(x) \) has the same monotonicity properties with the nominal value function \( v_t(x) \), regardless of the particular uncertainty
set. Particularly, this result implies that the structure of the optimal policy of robust problem is the same with the nominal problem.

Finally, let us briefly discuss the infinite horizon extension. Iyengar (2005) and Nilim and El Ghaoui (2005) establish that the respective controller and nature policies are stationary for the infinite horizon problem. Moreover, Nilim and El Ghaoui (2005) show that the optimal value function of the infinite horizon problem with a discounted cost function can be obtained as the unique limit of the finite horizon problem. This implies that the optimal policy structure can be extended to the infinite horizon case.

5 Structural Properties of Some Queueing and Inventory Control Problems

In this section, we introduce a number of commonly-used operators for queueing and inventory problems as in Koole (1998), (2006) and Çil et al. (2009). Table 1 presents the type (“I” for inventory and “Q” for queueing), names, notations and definitions of these operators. Their detailed definitions as well as the structural properties they preserve are summarized in Table 3 in Appendix A.

From Koole (1998) and Çil et al. (2009), it is known that if an initial value function \(v(x)\) is concave in \(x\), then the operators in Table 1 preserve concavity in \(x\). In addition, the queueing operators preserve the non-increasingness (NI) property whereas the inventory operators preserve the non-decreasingness (ND) property in \(x\). Note that additional conditions are necessary for general forms of \(T_{ARR}\) and \(T_{DEP}\), and for \(T_{COST}\) to preserve these properties as given in Çil et al. (2009). Çil et al. (2009) showed the following properties of the MB function:

Property 1 (Properties of the MB function):

1. The MB functions of the below operators are nonincreasing (NI) in \(x\).

   (a) The queueing operators, admission control \(B_{A,v}(x)\), batch admission \(B_{BA,v}(x)\), queue pricing \(B_{QP,v}(x)\) and uncontrolled arrival \(B_{UA,v}(x)\),

   (b) The inventory operators, production control \(B_{P,v}(x)\) and production rate control \(B_{PR,v}(x)\).

2. The MB functions of the below operators are nondecreasing (ND) in \(x\).
<table>
<thead>
<tr>
<th>Type</th>
<th>Operator Name</th>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Rationing</td>
<td>$T_R, v(x)$</td>
<td>Special case of $T_{BR_i}$ where $B = 1$</td>
</tr>
<tr>
<td>I</td>
<td>Batch Rationing</td>
<td>$T_{BR_i}, v(x)$</td>
<td>$\max_{\kappa_i \in \min(x, B)} {\kappa_i R_i + v(x - \kappa_i)}$</td>
</tr>
<tr>
<td>I</td>
<td>Production Rate</td>
<td>$T_{PR_i}, v(x)$</td>
<td>$\max_{\Pi \in [0,1]} {-C_{\Pi} + \Pi v(x + 1) + (1 - \Pi) v(x)}$</td>
</tr>
<tr>
<td>I</td>
<td>Production</td>
<td>$T_P, v(x)$</td>
<td>$\max{v(x + 1) - C_i, v(x)}$</td>
</tr>
<tr>
<td>I</td>
<td>Inventory Pricing</td>
<td>$T_{IP}, v(x)$</td>
<td>$\max_{\Pi \in [0,1]} {F_Z(R)[v(x + 1) + R] + F_Z(R) v(x)}$</td>
</tr>
<tr>
<td>Q</td>
<td>Admission</td>
<td>$T_{ADM_i}, v(x)$</td>
<td>Special case of $T_{BADM_i}$ where $B = 1$</td>
</tr>
<tr>
<td>Q</td>
<td>Batch Admission</td>
<td>$T_{BADM_i}, v(x)$</td>
<td>$\max_{\kappa_i} {\kappa_i R_i + v(x + \kappa_i)}$</td>
</tr>
<tr>
<td>Q</td>
<td>Controlled Departure</td>
<td>$T_{CD_i}, v(x)$</td>
<td>$\max{v(x - 1) - C_i, v(x)}$</td>
</tr>
<tr>
<td>Q</td>
<td>Departure Rate</td>
<td>$T_{DR_i}, v(x)$</td>
<td>$\max_{\Pi \in [0,1]} {-C_{\Pi} + \Pi v(x - 1) + (1 - \Pi) v(x)}$</td>
</tr>
<tr>
<td>Q</td>
<td>Queue Pricing</td>
<td>$T_{QP}, v(x)$</td>
<td>$\max_R {F_Z(R)[v(x + 1) + R] + F_Z(R) v(x)}$</td>
</tr>
<tr>
<td>Q</td>
<td>Uncontrolled Arrival</td>
<td>$T_{ARR}, v(x)$</td>
<td>$a(x) v(x + 1) + (1 - a(x)) v(x)$</td>
</tr>
<tr>
<td>Q</td>
<td>Uncontrolled Departure</td>
<td>$T_{DEP}, v(x)$</td>
<td>$b(x) v(x - 1) + (1 - b(x)) v(x)$</td>
</tr>
<tr>
<td></td>
<td>- Cost</td>
<td>$T_{COST}, v(x)$</td>
<td>$v(x) - h(x)$</td>
</tr>
<tr>
<td></td>
<td>- Fictitious</td>
<td>$T_{FIC}, v(x)$</td>
<td>$v(x)$</td>
</tr>
</tbody>
</table>

Table 1: Definitions for the operators (Q denotes a queueing operator and I denotes an inventory operator, see Appendix A for more details)

(a) The queueing operators, controlled departure $B_{CD_i}, v(x)$, departure rate $B_{DR_i}, v(x)$, uncontrolled departure $B_{UD_i}, v(x)$.

(b) The inventory operators, rationing $B_{R_i}, v(x)$, batch rationing $B_{BR_i}, v(x)$ and inventory pricing $B_{IP}, v(x)$.

Property 1 shows that these operators have all the properties that satisfy the assumptions of Theorem 2 and 3, so that all the nominal and robust MDP models that can be represented by a combination of these operators have monotone and concave value functions. Moreover, if all the operators in a certain model have MB-NI (MB-ND) property, then the corresponding value functions are also supermodular (submodular) in $x$ and $t$. However, note that, as remarked above, it is not possible to represent a meaningful queueing system with only the operators having MB-NI or MB-ND properties. Hence, queueing systems do not possess supermodularity or submodularity. There are, however, plausible inventory systems that support the property.
6 Illustration of the Framework

The aim of this section is to illustrate our results in well-known models. The illustrative example in Section 6.1 points out why our results are interesting even in a simple setting. Section 6.2, on the other hand, presents well-known problems in the literature that can be analyzed within our framework.

6.1 An Illustrative Example: Robust Revenue Management

To illustrate the type of results that we seek in the sequel, let us take the example of single-resource capacity control problem from revenue management Lautenbacher and Stidham (1999), where the objective is to admit or reject demands from customer classes with different rewards to maximize the expected reward until the end of the horizon. This problem is modeled using multiple inventory rationing operators. The nominal value function can be expressed as:

\[ v_t(x) = \sum_{i=1}^{n} p_{i,t} R_i v_{t+1}(x) + p_0, t v_{t+1}(x) \quad \text{for } x > 0 \]

where \( x \) denotes the number of available inventory (seats) and \( p_{i,t} \) is the probability of a class-\( i \) arrival with a corresponding reward of \( R_i \) in period \( t \) (\( p_{0,t} \) corresponds to the probability that there are no arrivals in period \( t \)).

Following the results in Cil et al. (2009) and Theorem 2, it is easily seen that \( v_t(x) \) is concave in \( x \) for all \( t \) and is supermodular in \((x, t)\). Using these properties, we conclude that a threshold (protection level) policy is optimal due to concavity and that the optimal thresholds are monotone over time due to supermodularity. This formulation naturally assumes that the event probabilities \( p_{i,t} \) do not depend on the state of the system.

To obtain numerical results, let us focus on a particular case of the above problem with two customer classes where the arriving batch size is equal to one (single arrivals) and \( p_{1,t} = p_1, p_{2,t} = p_2 \) for all \( t \). For each demand class \( a_i = 1 \) denotes the action that admits the arriving demand, and \( a_i = 0 \) corresponds to rejecting the customer. The nominal problem is then represented by:

\[ v_t(x) = p_1 \max \{ a_1 (v_{t+1}(x - 1) + R_1) + (1 - a_1) v_{t+1}(x) \} + p_2 \max \{ a_2 (v_{t+1}(x - 1) + R_2) + (1 - a_2) v_{t+1}(x) \} + p_3 v_{t+1}(x) \quad \text{for } x = 1, 2, \ldots, Q. \]

with \( v_T(x) = 0 \), for all \( x \), and \( v_t(0) = 0 \) for all \( t \).
We solve the resulting robust MDP numerically. Let  
\[ p(x, a) = \min_{p \in \mathcal{P}} \{ p_1(x, a) (a_1(w + 1(x - 1) + R_1) + (1 - a_1)w + 1(x)), \] 
\[ + p_2(x, a) (a_2(w + 1(x - 1) + R_1) + (1 - a_2)w + 1(x)) + p_3(x, a)w + 1(x) \} \] 
with  \[ w(0) = 0, \] for all  \( x \), and  \[ w(t) = 0 \] for all  \( t \).

Let us next consider the robust version where the uncertainty set  \( \mathcal{P} \) consists of two vectors:  
\[ p_1 = (p_1, p_2, p_3) = (0.48, 0.2, 0.32) \] and  \[ p_2 = (p_1, p_2, p_3) = (0.45, 0.55, 0) \]. There are clearly four
admissible actions as  \( a = (a_1, a_2) \). In summary, Nature is allowed to choose the probability distributions at each stage, state and action from  \( \mathcal{P} \):  
\[ (p_1, a, p_2, a, p_3, a) \in \mathcal{P} = \{ p_1, p_2 \} \].

Then the robust value function is given by the following recursion.
\[ w_t(x) = \max_a \left\{ \min_{p \in \mathcal{P}} \left\{ p_1(x, a) \left( a_1(w + 1(x - 1) + R_1) + (1 - a_1)w + 1(x) \right) \right. \right. \]
\[ + p_2(x, a) \left( a_2(w + 1(x - 1) + R_1) + (1 - a_2)w + 1(x) \right) + p_3(x, a)w + 1(x) \} \right\}, \]
with  \[ w_T(x) = 0, \] for all  \( x \), and  \[ w_t(0) = 0 \] for all  \( t \).

Let us further assume that  \( T = 20, R_1 = 10, R_2 = 1 \) and that the starting inventory level is 12. We solve the resulting robust MDP numerically. Let  \( a_t^*(x) \) denote the optimal action selected by controller at time  \( t \) and state  \( x \) and  \( p_t^*(x) \) denote the optimal event probability distribution selected by Nature for that action. Table 2 reports  \( p_t^*(x) \) and  \( a_t^*(x) \) for  \( t = 5, 10, 15 \) and  \( x = 1, 2, \ldots, 12 \). The numerical solution reveals an interesting property for this example. It is observed from Table 2 that both the controller’s and Nature’s optimal policies are state and time dependent. In particular, for Nature, different probability distributions may be optimal at different states and at different times. Regardless, the optimal policy of the controller is of threshold type. For each  \( t \), there is a protection level for Class-2 demand arrivals below which they are rejected. In addition, these protection levels are non-increasing in  \( t \). A closer examination reveals that the robust value function is concave in  \( x \) for all  \( t \) and is supermodular in  \( x \) and  \( t \) just like the nominal value function. This is an interesting observation because the event probability distributions chosen by Nature do depend on  \( t \) and on  \( x \) for the robust problem.

<table>
<thead>
<tr>
<th>( t \downarrow \mid x \to )</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
</tr>
<tr>
<td>( t \downarrow \mid x \to )</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td>(1, 0), p_2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Controller’s and Nature’s Optimal Policies for the Example
6.2 Illustrations from the Literature

In this section, we present some results on the structure of optimal policies for robust versions of some well-known examples from the literature using the earlier results.

We first consider an extended version of the single-resource capacity control problem introduced in Section 6.1 by adding dynamic pricing. In this case, rationing operators and dynamic pricing operators are used together in order to model a special customer segment (class $n+1$ that is offered a spot price). A typical value function is then given by:

$$v_t(x) = \sum_{i=1}^{n} p_{i,t} T_{BR} v_{t+1}(x) + p_{n+1,t} T_{IP} v_{t+1}(x) + p_{0,t} v_{t+1}(x),$$

By Theorem 3, the value function of the robust version is concave, so that optimal admission policies are of threshold type, and the thresholds are monotone over time. Moreover, optimal prices to charge to class $n+1$ are non-increasing in $x$ and in $t$.

Apart from discrete-time models, our results apply for continuous-time models under certain assumptions. In particular, a class of continuous-time problems can be converted to equivalent discrete time problems using uniformization (Lippman, 1975). Uniformization converts the transition rates of a continuous-time model to the transition probabilities of an equivalent discrete-time model. The basic assumption needed in the uniformization technique is the existence of a bound on the total potential transition rates in all states and time periods of the continuous-time model under consideration. To be able to use uniformization in a robust setting, the total potential transition rate should continue to be bounded over all the uncertainty sets in all time periods. If the uncertainty in the model can be represented after the conversion in discrete time, the results continue to apply directly. As an example, let us consider the uniformized version of a typical admission control problem to a single-server Markovian queue (Koole, 1998):

$$v_t(x) = -h(x) + p_{1,t} T_{ADM} v_{t+1}(x) + p_{2,t} T_{DEP} v_{t+1}(x) + p_{0,t} v_{t+1}(x),$$

where $x$ denotes the number of customers in the system, $p_{1,t}$ is the probability of an arrival, $p_{2,t}$ is the probability of a service completion and $h(x)$ is a non-decreasing and convex holding cost function.

From Theorem 3, the optimal admission control policy of the robust version of the above problem is a threshold policy but optimal thresholds are not necessarily monotone over time because the marginal benefits of the two operators have monotonicity properties in opposing directions.
Now, let us consider a dynamic pricing and production control problem for a make-to-stock queue with lost sales (see Gayon et al. (2004) for example). After uniformization, the value function is expressed as:

\[ v_t(x) = -h(x) + p_{1,t}TIPv_{t+1}(x) + p_{2,t}TPv_{t+1}(x) + p_{0,t}v_{t+1}(x), \]

where \( x \) denotes the available inventory, \( p_{1,t} \) is the probability of a potential demand arrival, \( p_{2,t} \) is the probability of production completion and \( h(x) \) is a non-decreasing and convex holding cost function.

Using Theorem 3, we observe that the optimal production policy is determined by a threshold and that the optimal prices are non-decreasing in \( x \) for the robust version of this problem.

7 Nested Uncertainty Sets: Structure of Optimal Policies

In this section, we have two aims: (1) investigating the conditions under which the monotonicity results shown in the previous section can be extended. (2) exploring whether we can establish certain monotonicity properties with respect to the uncertainty sets.

It has been shown by several authors that less conservative robust policies are more adaptive to volatile conditions than both nominal and absolute robust policies. Examples can be found in Paschalidis and Kang (2008), Xu and Mannor (2012) and Turgay (2014). We focus on a special formulation -S-robust policy-, an extended robust formulation recently proposed by Xu and Mannor (2012), and explore the structure of the robust value function for nested uncertainty sets. In this formulation, the uncertainty is not represented by a single uncertainty set, but by a number of uncertainty sets that have a nested structure. Each uncertainty set corresponds to a probabilistic guarantee for a different confidence level. Then, the corresponding optimal policy, which is called S-robust, must take into account these different probabilistic guarantees. In this section, we show that, for the class of problems treated in this paper, the structural properties of the standard and robust problem are retained for the S-robust counterpart.

Similarly to Xu and Mannor (2012), we define the nested uncertainty sets as follows \( P^1_t \subseteq P^2_t \subseteq \cdots \subseteq P^K_t \) for all \( t \), so that the uncertain transition probability distribution belongs to set \( P^K_t \) with probability \( \lambda_k \), where \( \lambda_k \geq \lambda_{k-1} \) and \( \sum_{k=1}^{K}(\lambda_k - \lambda_{k-1}) = 1 \) by setting \( \lambda_0 = 0 \). In this context,
a policy is said to be S-robust if it satisfies the following condition:

\[
w_t(x) = \max_a \left\{ \sum_{k=1}^{K} (\lambda_k - \lambda_{k-1}) \left( \min_{p_i(x,a) \in P_k^t} \sum_{i=0}^{n} \sum_{y \in X} q_t(a, x, y | i)(w_{t+1}(y) + R_t(a, x)) \right) \right\}, \tag{15}
\]

Equation (15) is a convex combination of \( K \) expressions with the same structural properties. It is clear that Theorem 3 is also valid for this equation. The S-robust value function \( w_t(x) \) can then be represented as follows:

\[
w_t(x) = \sum_{k=1}^{K} (\lambda_k - \lambda_{k-1}) \left( \min_{p_i(x) \in P_k^t} \left\{ \sum_{i} p_{i,t}(x) T_i w_{t+1}(x) \right\} \right), \tag{16}
\]

where \( p_k^t(x) = (p_{1,k}^t(x), \ldots, p_{n,k}^t(x)) \) is the transition probability distribution that Nature will select from uncertainty set \( P_k^t \). Note that (16) can also be expressed as:

\[
w_t(x) = \sum_{k=1}^{K} (\lambda_k - \lambda_{k-1}) \left( \min_{p_i(x) \in P_k^t} \left\{ \sum_{i} p_{i,t}(x) B_i w_{t+1}(x) \right\} \right) + w_{t+1}(x).
\]

**Remark:** We note the extension of the structural results to S-robust policies has not used the assumption that the sets are nested. Consequently, the structural results will still be valid when there are \( K \) different uncertainty sets \( P_t \) for all \( t \), where \( P_t \)'s are completely arbitrary. We let \( \gamma_k \) be the probability that uncertainty set \( k \) is used in any period, where \( \sum_{k=1}^{K} \gamma_k = 1 \). Then replacing \( (\lambda_k - \lambda_{k-1}) \) with \( \gamma_k \) in equation (16) will give the desired result.

**Remark:** S-robust policies can be used to reflect varying degrees of conservatism. As a simple example, consider the case where the controller believes that a certain probability distribution (nominal parameter) will occur with probability \( \lambda_1 \), and with probability \( 1 - \lambda_1 \) the probability distribution will be chosen from an uncertainty set (which includes the nominal parameter) by Nature in order to minimize the controller’s expected revenue. Then, taking \( \lambda_1 = 0 \) corresponds to the robust problem, where the controller is the most conservative, while \( \lambda_1 = 1 \) corresponds to the nominal problem, where the controller is the least conservative. Hence, increasing values of \( \lambda_1 \) shows decreasing degrees of conservatism. By using more than one \( \lambda_1 \), we can model even more complicated conservatism types and degrees. With this approach, we can define a set of comparable robust strategies systematically and evaluate their performance with respect to other parameters such as variance, fill rates, service level etc.

Now we investigate the possibility of having some monotonicity results with respect to uncertainty sets. In this case, we consider two similar robust MDP models, identical except for their
uncertainty sets which are nested, i.e., $P^1_t \subseteq P^2_t$ for all $t$. Let $w^k_t(x)$ denote the robust value function corresponding to $P^k_t$. Intuitively, we expect that the value function of the system with a bigger uncertainty set should be greater, i.e., $w^1_t(x) \leq w^2_t(x)$ for all $x$ and $t$. This can be formally proven by induction as in Paschalidis and Kang (2008).

On the other hand, establishing monotonicity of optimal policies, which require second order comparisons turns out to be much more challenging. In order to have optimal policies that are monotone with respect to the uncertainty set, we need to consider the differences $\Delta w^k_t(x) = w^k_t(x) - w^k_t(x-1)$. More explicitly, it is required to establish that either $\Delta w^1_t(x) \leq \Delta w^2_t(x)$ or $\Delta w^1_t(x) \geq \Delta w^2_t(x)$ for all $t$ and $x$. However, neither of these two cases appears to be always true in general, as shown by the counterexamples presented in Appendix B. Here, we provide some insights for not having this kind of monotonicity. Assume, without loss of generality, that $\Delta w^1_t(x) \geq \Delta w^2_t(x)$, i.e., $w^1_t(x) - w^2_t(x) \geq w^1_t(x-1) - w^2_t(x-1)$. This means that, as the uncertainty set gets larger, the loss in $w^1_t(x)$ is greater than the loss in $w^1_t(x-1)$. Further, suppose that $P^1_t$ contains all decreasing feasible directions for $w^1_t(x-1)$ but not decreasing feasible directions for $w^1_t(x)$. Since $P^1_t \subseteq P^2_t$, Nature can choose the same action in both $P^1_t$ and $P^2_t$, so $w^1_t(x-1) - w^2_t(x-1) = 0$, whereas $w^1_t(x) - w^2_t(x) \geq 0$ since Nature cannot decrease $w^1_t(x)$ in state $x$. (A numerical example illustrating this intuition is given in Appendix B). Therefore, nested uncertainty sets do not imply monotone control policies in general. However, it is possible to show such monotonicity holds when there is interval uncertainty, see Turgay et al. (2014). They show the existence of optimal thresholds, and establish that these thresholds change monotonically with respect to the enlargements in the uncertainty set.

8 Conclusions

We considered robust versions of a class of event-based DPs frequently encountered in queueing and inventory control. Under event probability uncertainty, we were able to show that robust optimal policies have the same structure as nominal optimal policies. This is very appealing from a practical point of view because simple policies are easier to communicate, parameterize and adjust. Under other uncertain parameters than event probabilities, general results appear difficult to obtain and simple policies may no longer be optimal.

Turgay (2012) considers the cases when other parameters, such as the rewards, are uncertain. However, the formulation of these robust problems are more complex and the results are very
positive in the sense that even when the value functions can be shown to be concave, it does not imply any monotonicity in the optimal policy. Hence, although this line of research is attractive, the results may not be fruitful.

Multi-dimensional problems in queueing and inventory control bring significant additional challenges because establishing structural properties usually requires verifying multiple conditions that should hold simultaneously. Investigating the structure of robust optimal policies for multi-dimensional problems is an important avenue for further research.

References


Appendix (intended for on-line publication)

A Structural Properties of the Nominal Problem

In this section we give the following monotonicity properties of the operators introduced in Table 1.

<table>
<thead>
<tr>
<th>Operator Name</th>
<th>Supermodularity in $(x, t)$</th>
<th>$Bv(x)$</th>
<th>$Tv(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rationing</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>ND. in $x$</td>
</tr>
<tr>
<td>Batch Rationing</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>ND. in $x$</td>
</tr>
<tr>
<td>Production</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>ND. in $x$</td>
</tr>
<tr>
<td>Production Rate</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>ND. in $x$</td>
</tr>
<tr>
<td>Inventory Pricing</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>ND. in $x$</td>
</tr>
<tr>
<td>Admission</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Batch Admission</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Controlled Departure</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Departure Rate</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Queue Pricing</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Uncontrolled Arrival to a Queue</td>
<td>submodular</td>
<td>NI. in $x$</td>
<td>NI. in $x$</td>
</tr>
<tr>
<td>Uncontrolled Departure from a Queue</td>
<td>supermodular</td>
<td>ND. in $x$</td>
<td>NI. in $x$</td>
</tr>
</tbody>
</table>

A.1 Inventory Control Operators

Batch Rationing Operator

The batch rationing operator represents the choice of the number of customers to be admitted from an arriving batch of class-$i$ customers with batch size $B_i$ in inventory systems. Some of the customers in a batch can be admitted while the remaining ones are rejected, which is defined as partial acceptance. $\kappa_i$ is the number of class-$i$ customers admitted from this batch, and $R_i$ is the reward obtained by admitting one class-$i$ customer.

Definition of the Operator

\[ T_{BR_i}v(x) = \max_{\kappa_i \in \min(x, B_i)} \{\kappa_i R_i + v(x - \kappa_i)\}, \]
$T_{BR_i}v(x) = \max_{\kappa_i \in \min(x,B)} \{ v(x - \kappa_i) - v(x) + \kappa_i R_i \} + v(x)$

**Rationing Operator**

The rationing operator is a special case of the batch rationing operator where the batch size $B$ is exactly 1. However, we provide independent proofs in this section.

**Definition of the Operator**

$T_{R_i}v(x) = \max \{ R_i + v(x - 1), v(x) \}$ for $x > 0,$

$T_{R_i}v(x) = \{ v(x - 1) - v(x) + R_i \}^+ + v(x)$ for $x > 0,$ $T_{R_i}v(x) = v(x)$ for $x = 0$

**Production Rate Operator**

Production rate operator represents the choice of optimal service rate in production-inventory systems for production unit $i$. If the system uses $\Pi_i$ portion of the service rate, then a nonnegative cost of $C_{\Pi_i}$ is incurred.

**Definition of the Operator**

$T_{PR_i}v(x) = \max_{\Pi_i \in [0,1]} \{ -C_{\Pi_i} + \Pi_i v(x + 1) + (1 - \Pi_i) v(x) \}$,

$T_{PR_i}v(x) = \max_{\Pi_i \in [0,1]} \{ \Pi_i \{ v(x + 1) - v(x) \} - C_{\Pi_i} \} + v(x)$

**Production Operator**

The production operator is a special case of production rate operator where $\Pi_i = \{ 0, 1 \}$ and $C_{\Pi_0} = 0.$

**Definition of the Operator**

$T_{P_i}v(x) = \max \{ v(x + 1) - C_i, v(x) \}$,

$T_{P_i}v(x) = \{ v(x + 1) - v(x) - C_i \}^+ + v(x)$

**Inventory Pricing Operator**

The inventory pricing operator represents the optimal price to be charged for the arriving customers in inventory systems. $F_Z(.)$ is the cumulative distribution function of the reservation price of an arriving customer, where $R$ is the maximum price a customer is willing to pay.

**Definition of the Operator**

$T_{IP_i}v(x) = \max_{R} \{ F_Z(R) [v(x - 1) + R] + F_Z(R) v(x) \}$ for $x > 0,$

$T_{IP_i}v(x) = \max_{R} F_Z(R) \{ v(x - 1) - v(x) + R \} + v(x)$ for $x > 0,$

$T_{IP_i}v(x) = v(x)$ for $x = 0.$

**A.2 Queueing Operators**

The queueing operators we consider throughout the paper are given in the following. It is important to note that the waiting room is infinite.
Batch Admission

The batch admission operator represents the choice of the number of the customers to be admitted from an arriving batch of class-$i$ customers with batch size $B$ in queueing systems. Some of the customers in a batch can be admitted while the remaining ones are rejected, which is defined as partial acceptance $\kappa_i$ is the number of class-$i$ customers admitted from this batch, and $R_i$ is the reward obtained by admitting one class-$i$ customer.

**Definition of the Operator**

$T_{BADM_i}v(x) = \max_{\kappa_i \in \min(x, B)} \{\kappa_i R_i + v(x + \kappa_i)\}.$

$T_{BADM_i}v(x) = \max_{\kappa_i \in \min(x, B)} \{v(x + \kappa_i) - v(x) + \kappa_i R_i\} + v(x)$

**Admission**

The admission operator is a special case of the batch admission operator where the batch size $B$ is exactly 1.

**Definition of the Operator**

$T_{ADM_i}v_t(x) = \max \{R_i + v_{t+1}(x + 1), v_{t+1}(x)\},$

$T_{ADM_i}v_t(x) = \{v_{t+1}(x + 1) - v_{t+1}(x) + R_i\}^+ + v_{t+1}(x).$

**Departure Rate Operator**

The departure rate operator represent the choice of the best service rate in queueing systems. If the system uses $\Pi$ portion of the service rate, then a nonnegative cost of $C_{\Pi}$ is incurred.

**Definition of the Operator**

$T_{DR_i}v(x) = \max_{\Pi \in [0, 1]} \{-C_{\Pi} + \Pi v(x - 1) + (1 - \Pi)v(x)\}$ for $x > 0,$

$T_{DR_i}v(x) = \max_{\Pi \in [0, 1]} \{\Pi\{v(x - 1) - v(x)\} - C_{\Pi}\} + v(x)$ for $x > 0,$

$T_{DR_i}v(x) = v(x)$ for $x = 0$

**Controlled Departure Operator**

The controlled departure operator is a special case of the departure rate operator where $\Pi = \{0, 1\}$ and $C_0 = 0.$

**Definition of the Operator**

$T_{CD_i}v(x) = \max \{v(x - 1) - C_i, v(x)\}$ for $x > 0,$

$T_{CD_i}v(x) = \{v(x - 1) - v(x) - C_i\}^+ + v(x)$ for $x > 0,$

$T_{CD_i}v(x) = v(x)$ for $x = 0.$

**Queue Pricing Operator**

The queue pricing operator represent the optimal price to be charged for the arriving customers in queueing systems. $F_Z(.)$ is the cumulative distribution function of the reservation price of an
arriving customer, where $R$ is the maximum price a customer is willing to pay.

**Definition of the Operator**

\[ T_{QP}v(x) = \max_R \left\{ F_Z(R)[v(x+1) + R] + F_Z(R)v(x) \right\}, \]
\[ T_{QP}v(x) = \max_R F_Z(R)\{v(x+1) - v(x) + R\} + v(x) \]

**Uncontrolled Arrival to a Queue**

The uncontrolled arrival operator represents the arrival process to a queueing system. The function $a(x)$ is the probability that an arriving customer joins the system when there are $x$ customers, which we refer to as the joining probability. We assume that $a(x)$ is NI in $x$. When $a$ is constant, arrival operator models a system where customers enter the system with a fixed probability, independent of the state, or choose not to enter the system with probability $1-a$. We will call this type of arrivals as regular arrivals, since they do not depend on the state of the system.

**Definition of the Operator**

\[ T_{ARR}v(x) = a(x)v(x+1) + (1-a(x))v(x), \]
\[ T_{ARR}v(x) = a(x)\{v(x+1) - v(x)\} + v(x) \]

**Uncontrolled Departure to a Queue**

The uncontrolled departure operator represents the departure of an existing customer from the system, where the service rate may depend on the state of the system. The function $b(x)$ corresponds to the probability of a service completion when the system has $x$ customers. We assume that $b(x)$ is an ND function of $x$.

**Definition of the Operator**

\[ T_{DEP}v(x) = b(x)v(x-1) + (1-b(x))v(x) \text{ for } x > 0, \]
\[ T_{DEP}v(x) = b(x)\{v(x-1) - v(x)\} + v(x) \text{ for } x > 0, \]
\[ T_{DEP}v(x) = v(x) \text{ for } x > 0. \]

**B Counter Example for Monotonicity Of Thresholds**

We consider a single-resource revenue management system with 4 customer classes and compare two systems. The admission reward of each class is as follows: $R_1 = 40, R_2 = 35, R_3 = 30, R_4 = 20$ and is identical for the two systems.

The uncertainty set defining probabilities are denoted as $P$ and defined as follows:
\[ P = \{(p_1, p_2, p_3, p_4) : 4p_1 + 2p_2 + 3p_3 + p_4 \geq 1.65, \\
0.1 \leq p_i \leq 0.2 i = 1, ..., 4\}. \]

Now consider the second system. The uncertainty set \( P' \) of that system includes the uncertainty set of the first system \( P' \supseteq P \). The definition of \( P' \) is as follows:

\[ P' = \{(p_1, p_2, p_3, p_4) : 4p_1 + 2p_2 + 3p_3 + p_4 \geq 1.65, \\
0.1 \leq p_{i,t} \leq 0.2 i = 1, ..., 4, t \neq T - 5, \\
0.05 \leq p_{i,t} \leq 0.325 i = 1, 2, 0.1 \leq p_{i,t} \leq 0.2 i = 3, 4 \ t = T - 5\}. \]

If we denote the value function and probabilities of the second system with \( w'_t(x) \) and \( p'_{i,t}(x) \), it is apparent that \( w'_t(x) = w_t(x) \) and \( p'_{i,t}(x) = p_t(x) \) for \( t = T - 1, ..., T - 4 \).

We then numerically calculate the values for \( w_{T-5}(x) \) and \( w'_{T-5}(x) \) for \( x = 1, 2 \). The results are as follows where the greater values between the systems are shown with boldface letters:

<table>
<thead>
<tr>
<th>( x )</th>
<th>( w_{T-5}(x) )</th>
<th>( w'_{T-5}(x) )</th>
<th>( \Delta w_{T-5}(x) )</th>
<th>( \Delta w'_{T-5}(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33.68</td>
<td>33.54</td>
<td>33.68</td>
<td>33.54</td>
</tr>
<tr>
<td>2</td>
<td>61.53</td>
<td>61.40</td>
<td>27.85</td>
<td>27.86</td>
</tr>
</tbody>
</table>

Therefore, in the above example, neither of the \( \Delta w^P \geq (\leq) \Delta w^{P'} \) statements are true.