

Computing Protection Level Policies for Dynamic Capacity Allocation Problems by Using Stochastic Approximation Methods

Alexander Erdelyi
School of Operations Research and Information Engineering,
Cornell University, Ithaca, NY 14853, USA
alex@orie.cornell.edu

Huseyin Topaloglu
(Corresponding Author)
School of Operations Research and Information Engineering,
Cornell University, Ithaca, NY 14853, USA
topaloglu@orie.cornell.edu
Tel: 1-607-255-0698 Fax: 1-607-255-9129

May 30, 2008

Abstract

In this paper, we consider a dynamic capacity allocation problem. We have a fixed amount of daily processing capacity. Jobs of different priorities arrive randomly over time and we need to decide which jobs should be scheduled on which days. The jobs that are waiting to be processed incur a holding cost depending on their priority levels. We want to minimize the total expected cost over a planning horizon. In this paper, we focus on a class of policies that are characterized by a set of protection levels. The role of the protection levels is to “protect” a portion of the capacity from the lower priority jobs so as to make it available for the future higher priority jobs. We develop a stochastic approximation method to find a good set of protection levels and prove its convergence. Computational experiments indicate that protection level policies perform especially well when the coefficient of variation for the job arrivals is high.

The problem of allocating limited capacity among competing jobs in a dynamic fashion occurs in many settings. Clinics allocate appointment slots to patients with different needs, production plants allocate production capacity to orders with different priorities, hotels allocate room inventory to customers with different willingness to pay amounts. Controlling such systems requires careful planning for several reasons. To begin with, the decisions for the current day are made with limited information about the future job arrivals. The problem is dynamic in the sense that the future capacity that is not committed to the jobs that arrive today can be committed to the jobs that arrive tomorrow. Finally, it is crucial to keep a balance between committing the capacity to a lower priority job that is available today and reserving the capacity for a potential higher priority job that may arrive tomorrow.

In this paper, we consider a capacity allocation problem that captures the tradeoffs that we describe above. We have a fixed amount of daily processing capacity. Jobs of different priorities arrive randomly over time and we need to decide which jobs should be scheduled on which days. The jobs that are waiting to be processed incur a holding cost depending on their priority levels. The goal is to minimize the total expected cost over a planning horizon. It is possible to formulate this problem as a dynamic program, but this formulation quickly gets intractable for practical problem instances. Instead, we focus on a class of suboptimal policies that are characterized by a set of protection levels and develop a stochastic approximation method to find a good set of protection levels. Throughout the paper, we use the term “good” to refer to protection levels that yield near optimal total expected costs.

A protection level policy behaves similar to a prioritized first come first serve policy. As the jobs arrive over time, it commits the capacity starting from the highest priority jobs, but the protection levels stipulate how much capacity should be left untouched for the future job arrivals. The key idea behind our approach is to write down the total expected cost over the planning horizon as a function of the protection levels and to search for a good set of protection levels by using sample path-based derivatives of the total expected cost function. However, this approach brings unique challenges in the context of the capacity allocation problem. First, jobs are indivisible and if we change a particular protection level by an infinitesimal amount, then the number of jobs scheduled on a particular day either does not change or changes by an integer amount. This implies that the sample path-based derivatives of the total expected cost function are either equal to zero or do not exist. We deal with this difficulty by computing the sample path-based derivatives of the total expected cost function under the assumption that the jobs are divisible and different portions of a job can be scheduled on different days. Naturally, it is not realistic to assume that the jobs are divisible and we drop this assumption when implementing a protection level policy in practice. Second, even if we assume that the jobs are divisible, the sample path-based derivatives of the total expected cost function may not exist when the number of jobs of a particular priority level is exactly equal to the amount of capacity that the protection levels make available. We deal with this difficulty by perturbing the capacities by small random amounts. The role of these two modifications is to smooth the problem so that the sample path-based derivatives of the total expected cost function exist with probability one (w.p.1). In this case, we can develop a stochastic approximation method to find a good set of protection levels.

Several papers study variations of our capacity allocation problem. The most related one is Patrick, Puterman and Queyranne (2008), where the authors consider the allocation of appointment slots in a

clinic. The authors begin by formulating their capacity allocation problem as a dynamic program. Since this formulation quickly gets intractable for practical problem instances, they develop an approximate dynamic programming method that uses linear approximations to the value functions. To choose the slope parameters of the linear approximations, they plug the linear approximations into a linear program that characterizes the dynamic programming formulation of the capacity allocation problem. We note that both our approach and the approach followed by Patrick et al. (2008) can incur approximation errors. In particular, we apriori focus on a class of policies that are characterized by a set of protection levels, whereas Patrick et al. (2008) apriori focus on a class of value function approximations that are linear. If the optimal policy cannot be approximated well with a set of protection levels, then the approximation error suffered by our approach can potentially get large. On the other hand, if the value functions cannot be approximated well with linear functions, then the approximation error suffered by the approach followed by Patrick et al. (2008) can potentially get large. It is also worthwhile to emphasize that an important distinction between the two approaches arises when we use them to make the capacity allocation decisions. Making the capacity allocation decisions by using a set of protection levels is simple. We commit the capacity starting from the highest priority jobs, but the protection levels stipulate how much capacity should be left untouched for the future job arrivals. In contrast, if we have linear approximations to the value functions, then making the capacity allocation decisions involves solving an integer program that minimizes the sum of the immediate cost and the value function approximation at the next time period. Under certain assumptions on the cost parameters, Patrick et al. (2008) show that one can find a closed-form solution to this integer program. When these assumptions do not hold, however, one needs to solve a small integer program to make the capacity allocation decisions. Despite this slight potential disadvantage, an important advantage of the approach followed by Patrick et al. (2008) is that it naturally provides lower bounds on the optimal total expected cost, whereas we work with the perfect hindsight problem to obtain lower bounds in this paper.

A second paper that is related to our work is Gerchak, Gupta and Henig (1996), where the authors consider a variant of our capacity allocation problem with two priority levels. The main contribution of their paper is to characterize the structure of the optimal policy and to show that protection level policies are not necessarily optimal. Although protection level policies are not necessarily optimal, extensive computational experiments indicate that the performances of protection level policies differ from the optimal total expected costs by at most 1.56%. Gerchak et al. (1996) do not provide an algorithmic method to compute good protection levels for large problems, but their work motivates us to pursue protection level policies for problems with arbitrary number of priority levels. Finally, a third paper by Gupta and Wang (2007) presents a model that considers the customer choice behavior and again demonstrates that the optimal policy may have a complicated structure. Their model focuses on scheduling appointments on a fixed day and does not capture the overflow between different days.

We borrow the idea of protection levels from the revenue management literature. Littlewood (1972), Wollmer (1992) and Brumelle and McGill (1993) show that it is possible to use protection levels to characterize the optimal policies for single flight leg revenue management problems. On the other hand, protection level policies are not necessarily optimal for network revenue management problems, but their intuitive appeal and simplicity make them a popular choice in practice. Talluri and van Ryzin (2004)

and Phillips (2005) describe methods to compute protection levels by decomposing the network revenue management problem into a sequence of single flight leg problems. Bertsimas and de Boer (2005), van Ryzin and Vulcano (2006) and van Ryzin and Vulcano (2004) compute protection levels by using stochastic approximation methods. It is interesting to note that the capacity allocation problem that we consider in this paper can be viewed as a revenue management problem as it involves making the most use out of limited and perishable capacity. However, traditional revenue management problems involve only the decision of whether to accept or reject a job, whereas our capacity allocation problem involves the decision of when to schedule a job along with whether the job should be accepted or rejected.

The use of stochastic approximation methods for solving stochastic optimization problems is well-known. Kushner and Clark (1978) and Bertsekas and Tsitsiklis (1996) provide excellent coverage of the theory of stochastic approximation methods. As far as the application areas are concerned, in addition to the papers mentioned in the paragraph above, Karaesmen and van Ryzin (2004) provide applications in revenue management, L'Ecuyer and Glynn (1994) provide applications in queueing and Fu (1994), Glasserman and Tayur (1995), Bashyam and Fu (1998), Mahajan and van Ryzin (2001) and Kunnumkal and Topaloglu (2007) provide applications in inventory control.

We make the following research contributions in this paper. 1) We consider a capacity allocation problem that involves scheduling jobs of different priorities in a dynamic fashion. We focus on a class of policies characterized by protection levels and develop a stochastic approximation method to find a good set of protection levels. 2) We show that the iterates of our method converge to a stationary point of the total expected cost function. 3) Computational experiments demonstrate that the protection levels obtained by our method perform well. In all of our test problems, the performance gap between the protection level policy obtained by our method and the optimal policy is at most 4.08%. Furthermore, the performance gap with a standard benchmark method that uses a deterministic linear program becomes particularly noticeable when the coefficient of variation for the job arrivals is high.

The rest of the paper is organized as follows. Section 1 describes the capacity allocation problem and gives a precise definition for the protection level policies. Section 2 formulates a basic optimization problem that minimizes the total expected cost by choosing the protection levels. Section 3 derives a number of recursions that can be used to compute the sample path-based derivatives of the total expected cost function. Section 4 gives a stochastic approximation method to find a good set of protection levels and establishes a convergence result for this method. Section 5 shows how to obtain lower bounds on the optimal total expected cost by using the perfect hindsight problem. Such lower bounds become useful when establishing the optimality gap of a suboptimal policy. Section 6 presents computational experiments. Section 7 gives concluding remarks.

1 CAPACITY ALLOCATION PROBLEM AND PROTECTION LEVEL POLICIES

We have a fixed amount of daily processing capacity. Jobs of different priorities arrive randomly over time and we have to decide which jobs should be scheduled on which days. The jobs that are waiting to be processed incur a holding cost depending on their priority levels. We are interested in minimizing the total expected cost over a finite planning horizon.

The problem takes place over the planning horizon $\mathcal{T} = \{1, \dots, T\}$. A job arriving on day t can be scheduled on any of the S future days. We refer to S as the length of the booking horizon and let $\mathcal{S} = \{1, \dots, S\}$ throughout the paper. The length of the booking horizon is a part of the problem data and it represents how far into the future we are willing to schedule the jobs. There are different priority levels for the jobs and we denote the priority levels by $\mathcal{N} = \{1, \dots, N\}$. We adopt the convention that the jobs with smaller priority indices have higher priorities. The priority of a job does not change. The daily available capacity is $c > 0$. For notational brevity, we assume that each job consumes one unit of capacity, but it is possible to extend our approach to multiple units of capacity consumption. The number of priority i jobs that arrive into the system on day t is given by the random variable d_{it} . If we schedule a priority i job on the s th day of the booking horizon, then we incur a holding cost of \tilde{h}_{is} . We also have the option of rejecting a job and the penalty cost associated with rejecting a priority i job is \tilde{b}_i . Rejected jobs leave the system and do not come back. The jobs with smaller priority indices have higher priorities in the sense that we have $\tilde{h}_{is} \geq \tilde{h}_{i+1,s}$ and $\tilde{b}_i \geq \tilde{b}_{i+1}$ for all $i \in \mathcal{N} \setminus \{N\}$, $s \in \mathcal{S}$. We also assume that we have $\tilde{h}_{is} \leq \tilde{h}_{i,s+1}$ and $\tilde{h}_{iS} \leq \tilde{b}_i$ for all $i \in \mathcal{N}$, $s \in \mathcal{S} \setminus \{S\}$ so that longer delays mean higher holding costs and rejecting a job is never the trivially optimal decision. Timeline of events in a given day is straightforward. At the beginning of day t , we observe the job arrivals $\{d_{it} : i \in \mathcal{N}\}$ and decide which jobs should be scheduled on which days of the booking horizon $\{t, \dots, t + S - 1\}$. At the end of day t , we process the jobs that are scheduled on this day.

We use x_{st} to denote the remaining capacity on the s th day of the booking horizon when we are on day t so that the vector $x_t = \{x_{st} : s \in \mathcal{S}\}$ gives the remaining capacities over the booking horizon on day t . In this case, using x_t as the state variable and assuming that the job arrivals $d_t = \{d_{it} : i \in \mathcal{N}\}$ on day t are independent of the job arrivals on the other days, it is possible to formulate the problem as a dynamic program. However, since the state variable x_t is a high dimensional vector, this dynamic programming formulation quickly gets intractable for practical problem instances. Instead, we focus on a class of suboptimal policies that are characterized by a set of protection levels. In particular, the policy characterized by protection levels $\lambda = \{\lambda_{is} : i \in \mathcal{N}, s \in \mathcal{S}\}$ fills the capacity starting from the jobs with the highest priority, but it leaves at least λ_{is} units of capacity untouched when scheduling the priority i jobs on the s th day of the booking horizon. We expect the protection level policies to perform well since they tend to schedule the higher priority jobs early in the booking horizon and they allow us to protect the capacity from the lower priority jobs so that we can use the protected capacity for the potential future higher priority jobs.

To be specific, we let \tilde{u}_{is} be the number of priority i jobs that we schedule on the s th day of the booking horizon. In this case, given that the remaining capacities over the booking horizon are x_t and the job arrivals are d_t , we make the following decisions on day t under the policy characterized by protection levels λ . Leaving λ_{11} units of capacity untouched and letting $[\cdot]^+ = \max\{\cdot, 0\}$, we schedule $\tilde{u}_{11} = \min\{d_{1t}, [x_{1t} - \lambda_{11}]^+\}$ priority 1 jobs on day 1 of the booking horizon. Since there are now $x_{1t} - \tilde{u}_{11}$ units of remaining capacity on day 1 of the booking horizon, we schedule $\tilde{u}_{21} = \min\{d_{2t}, [x_{1t} - \tilde{u}_{11} - \lambda_{21}]^+\}$ priority 2 jobs on day 1 of the booking horizon. The decisions for the lower priority jobs for day 1 of the booking horizon are made in a similar fashion. After making the decisions for day 1 of the booking horizon, there are $d_{1t} - \tilde{u}_{11}$ priority 1 jobs remaining and we schedule $\tilde{u}_{12} = \min\{d_{1t} - \tilde{u}_{11}, [x_{2t} - \lambda_{12}]^+\}$

priority 1 jobs on day 2 of the booking horizon. The decisions for the subsequent days in the booking horizon are made in a similar fashion. We note that as long as the protection levels take integer values, the numbers of jobs that we schedule on each day of the booking horizon also take integer values.

Generalizing the discussion above, as a function of the remaining capacities over the booking horizon and the job arrivals on day t , the policy characterized by protection levels λ schedules

$$\tilde{u}_{is}(x_t, d_t, \lambda) = \min \left\{ d_{it} - \tilde{u}_{i,s-1}(x_t, d_t, \lambda) - \dots - \tilde{u}_{i1}(x_t, d_t, \lambda), \right. \\ \left. [x_{st} - \tilde{u}_{i-1,s}(x_t, d_t, \lambda) - \dots - \tilde{u}_{1s}(x_t, d, \lambda) - \lambda_{is}]^+ \right\} \quad (1)$$

priority i jobs on the s th day of the booking horizon. We refer to $\tilde{u}_{is}(\cdot, \cdot, \lambda)$ as the decision function for the policy characterized by protection levels λ . In the expression above, $d_{it} - \tilde{u}_{i,s-1}(x_t, d_t, \lambda) - \dots - \tilde{u}_{i1}(x_t, d_t, \lambda)$ gives the number of remaining priority i jobs after making the decisions for the earlier days in the booking horizon, whereas $x_{st} - \tilde{u}_{i-1,s}(x_t, d_t, \lambda) - \dots - \tilde{u}_{1s}(x_t, d, \lambda)$ gives the remaining capacity on the s th day of the booking horizon after making the decisions for the higher priority jobs. Therefore, leaving λ_{is} units of capacity untouched, we have $[x_{st} - \tilde{u}_{i-1,s}(x_t, d_t, \lambda) - \dots - \tilde{u}_{1s}(x_t, d, \lambda) - \lambda_{is}]^+$ units of capacity available for priority i jobs on the s th day of the booking horizon.

The difficulty associated with the decision function in (1) is that $\tilde{u}_{is}(x_t, d_t, \lambda)$ is not necessarily differentiable with respect to λ when the two arguments of $\min\{\cdot, \cdot\}$ operator are equal to each other or when the argument of $[\cdot]^+$ operator is equal to zero. This implies that the total cost incurred by the policy that uses the decision function in (1) is not necessarily differentiable with respect to the protection levels. Since the total cost is not necessarily differentiable, we cannot find a good set of protection levels by using the sample path-based derivatives of the total expected cost function. In the next section, we develop a smooth decision function that is differentiable w.p.1.

2 SMOOTHING THE DECISION FUNCTION

In this section, we develop a smooth version of the decision function in (1) that allows us to find a good set of protection levels by using the sample path-based derivatives of the total expected cost function. As mentioned before, if the two arguments of $\min\{\cdot, \cdot\}$ operator are equal to each other or the argument of $[\cdot]^+$ operator is equal to zero, then the decision function in (1) is not necessarily differentiable. We resolve this difficulty by using an approach proposed by van Ryzin and Vulcano (2006). In particular, we let $\{\alpha_{ist} : i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}\}$ be uniformly distributed random variables over a small interval $[0, \varepsilon]$ and perturb the remaining capacities over the booking horizon by $\{\alpha_{ist} : s \in \mathcal{S}\}$ when we make the decisions for priority i jobs on day t . This implies that we use the decision function

$$u_{is}(x_t, d_t, \alpha_t, \lambda) = \min \left\{ d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda), \right. \\ \left. [x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is}]^+ \right\}, \quad (2)$$

where we let $\alpha_t = \{\alpha_{ist} : i \in \mathcal{N}, s \in \mathcal{S}\}$. We note that perturbing the remaining capacities in this fashion is equivalent to assuming that very small but random extra capacity of α_{ist} units becomes available when we make the decisions for priority i jobs on the s th day of the booking horizon. When ε is small, we expect that perturbing the capacities in this fashion does not bring too much error.

Assuming that $\{\alpha_{ist} : i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}\}$ are independent of $\{d_{it} : i \in \mathcal{N}, t \in \mathcal{T}\}$ and $\{\alpha_{ist} : i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}\}$ are independent of each other, the event $x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is} = 0$ occurs with probability zero. In this case, it is possible to show that $u_{is}(x_t, d_t, \alpha_t, \lambda)$ is differentiable with respect to x_t and λ w.p.1. We formally show this fact in Lemma 2 in Appendix A. It is important to note that the numbers of jobs that the decision function in (2) schedules on each day of the booking horizon are not necessarily integers. Therefore, the decision function in (2) implicitly assumes that the jobs are divisible. We also emphasize that α_{ist} does not have to be uniformly distributed as long as it takes values over a set of nonzero Lebesgue measure.

Given the remaining capacities x_t over the booking horizon, job arrivals d_t and capacity perturbations α_t on day t , the remaining capacities over the booking horizon on day $t + 1$ are given by

$$R_s(x_t, d_t, \alpha_t, \lambda) = x_{s+1,t} + \sum_{i \in \mathcal{N}} \alpha_{i,s+1,t} - \sum_{i \in \mathcal{N}} u_{i,s+1}(x_t, d_t, \alpha_t, \lambda) \quad \text{for all } s \in \mathcal{S} \setminus \{S\} \quad (3)$$

$$R_S(x_t, d_t, \alpha_t, \lambda) = c. \quad (4)$$

The first expression above uses the observation that the s th day of the booking horizon on day $t + 1$ corresponds to the $(s + 1)$ st day of the booking horizon on day t . We refer to $R_s(\cdot, \cdot, \cdot, \lambda)$ as the transition function and if we follow the policy characterized by protection levels λ , then the remaining capacities over the booking horizon on the next day are given by $R(x_t, d_t, \alpha_t, \lambda) = \{R_s(x_t, d_t, \alpha_t, \lambda) : s \in \mathcal{S}\}$.

We use $d = \{d_{it} : i \in \mathcal{N}, t \in \mathcal{T}\}$ to denote the job arrivals and $\alpha = \{\alpha_{ist} : i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}\}$ to denote the capacity perturbations over the whole planning horizon. In this case, as a function of the remaining capacities x_t over the booking horizon, job arrivals d and capacity perturbations α , the cost function of the policy characterized by protection levels λ can be written recursively as

$$C_t(x_t, d, \alpha, \lambda) = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} \tilde{h}_{is} u_{is}(x_t, d_t, \alpha_t, \lambda) + \sum_{i \in \mathcal{N}} \tilde{b}_i \left[d_{it} - \sum_{s \in \mathcal{S}} u_{is}(x_t, d_t, \alpha_t, \lambda) \right] + C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda) \quad (5)$$

with the boundary condition that $C_{T+1}(\cdot, \cdot, \cdot, \lambda) = 0$. The boundary condition indicates that we do not incur costs once we reach the end of the planning horizon. Therefore, if we use the policy characterized by protection levels λ and the remaining capacities over the booking horizon on day t are x_t , then the total cost that we incur over days $\{t, \dots, T\}$ is given by the random variable $C_t(x_t, d, \alpha, \lambda)$. This implies that we can find a good set of protection levels by solving the problem

$$\min_{0 \leq \lambda \leq c} \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}. \quad (6)$$

The problem above finds protection levels λ so that the total expected cost incurred by the policy characterized by protection levels λ is minimized. The constraint $0 \leq \lambda \leq c$ should be understood as $0 \leq \lambda_{is} \leq c$ for all $i \in \mathcal{N}, s \in \mathcal{S}$ and x_1 is the initial state of the system, which is a part of the problem data. Using the fact that the decision function is differentiable with respect to the remaining capacities and protection levels w.p.1, it is possible to check by backward induction on (5) that $C_1(x_1, d, \alpha, \lambda)$ is differentiable with respect to the protection levels w.p.1 and it may be possible to solve problem (6) by using the sample path-based derivatives of $C_1(x_1, d, \alpha, \lambda)$.

We close this section by noting that we can assume without loss of generality that $\tilde{b}_i = 0$ for all $i \in \mathcal{N}$. To see this, we write (5) as $C_t(x_t, d, \alpha, \lambda) = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} [\tilde{h}_{is} - \tilde{b}_i] u_{is}(x_t, d_t, \alpha_t, \lambda) + \sum_{i \in \mathcal{N}} \tilde{b}_i d_{it} + C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda)$. Since $\sum_{i \in \mathcal{N}} \tilde{b}_i d_{it}$ is a constant, assuming that $\tilde{b}_i = 0$ for all $i \in \mathcal{N}$ is equivalent to letting $h_{is} = \tilde{h}_{is} - \tilde{b}_i$ and using the holding costs $\{h_{is} : i \in \mathcal{N}, s \in \mathcal{S}\}$. Consequently, we use the simplified cost function

$$C_t(x_t, d, \alpha, \lambda) = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} u_{is}(x_t, d_t, \alpha_t, \lambda) + C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda) \quad (7)$$

throughout the rest of the paper. Due to the assumption that $\tilde{h}_{is} \leq \tilde{b}_i$ for all $i \in \mathcal{N}, s \in \mathcal{S}$, we note that $h_{is} \leq 0$ for all $i \in \mathcal{N}, s \in \mathcal{S}$.

3 COMPUTING THE SAMPLE PATH-BASED DERIVATIVES

It is relatively easy to come up with a recursion that computes the sample path-based derivatives of the cost function. We begin by defining some new notation. We let $\partial_{jr}^\Lambda C_t(x_t, d, \alpha, \lambda)$ be the derivative of $C_t(\cdot, d, \alpha, \cdot)$ with respect to the protection level for priority j jobs on the r th day of the booking horizon evaluated at remaining capacities x_t and protection levels λ . In other words, we have

$$\partial_{jr}^\Lambda C_t(x_t, d, \alpha, \lambda) = \left. \frac{\partial C_t(z_t, d, \alpha, \kappa)}{\partial \kappa_{jr}} \right|_{(z_t, \kappa) = (x_t, \lambda)}.$$

Similarly, we let $\partial_r^X C_t(x_t, d, \alpha, \lambda)$ be the derivative of $C_t(\cdot, d, \alpha, \cdot)$ with respect to the remaining capacity on the r th day of the booking horizon evaluated at remaining capacities x_t and protection levels λ . That is, we have

$$\partial_r^X C_t(x_t, d, \alpha, \lambda) = \left. \frac{\partial C_t(z_t, d, \alpha, \kappa)}{\partial z_{rt}} \right|_{(z_t, \kappa) = (x_t, \lambda)}.$$

We use $\partial_{jr}^\Lambda u_{is}(x_t, d_t, \alpha_t, \lambda)$, $\partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda)$, $\partial_{jr}^\Lambda R_s(x_t, d_t, \alpha_t, \lambda)$ and $\partial_r^X R_s(x_t, d_t, \alpha_t, \lambda)$ with similar interpretations.

Differentiating (7) with respect to the protection level for priority j jobs on the r th day of the booking horizon and using the chain rule, we obtain

$$\begin{aligned} \partial_{jr}^\Lambda C_t(x_t, d, \alpha, \lambda) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \partial_{jr}^\Lambda u_{is}(x_t, d_t, \alpha_t, \lambda) + \partial_{jr}^\Lambda C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda) \\ &\quad + \sum_{s \in \mathcal{S}} \partial_s^X C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda) \partial_{jr}^\Lambda R_s(x_t, d_t, \alpha_t, \lambda). \end{aligned} \quad (8)$$

To compute the terms on the right side of the expression above, we differentiate (7) with respect to the remaining capacity on the r th day of the booking horizon to obtain

$$\begin{aligned} \partial_r^X C_t(x_t, d, \alpha, \lambda) &= \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda) \\ &\quad + \sum_{s \in \mathcal{S}} \partial_s^X C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda) \partial_r^X R_s(x_t, d_t, \alpha_t, \lambda). \end{aligned} \quad (9)$$

On the other hand, differentiating (3) and (4) with respect to the protection level for priority j jobs on the r th day of the booking horizon, we have

$$\partial_{j_r}^\Lambda R_s(x_t, d_t, \alpha_t, \lambda) = - \sum_{i \in \mathcal{N}} \partial_{j_r}^\Lambda u_{i,s+1}(x_t, d_t, \alpha_t, \lambda) \quad (10)$$

with the interpretation that $u_{i,s+1}(\cdot, \cdot, \cdot, \lambda) = 0$. Similarly, differentiating (3) and (4) with respect to the remaining capacity on the r th day of the booking horizon, we have

$$\partial_r^X R_s(x_t, d_t, \alpha_t, \lambda) = \mathbf{1}(r = s + 1) - \sum_{i \in \mathcal{N}} \partial_r^X u_{i,s+1}(x_t, d_t, \alpha_t, \lambda), \quad (11)$$

where $\mathbf{1}(\cdot)$ is the indicator function. Finally, differentiating (2) with respect to the protection level for priority j jobs on the r th day of the booking horizon, we obtain

$$\begin{aligned} & \partial_{j_r}^\Lambda u_{is}(x_t, d_t, \alpha_t, \lambda) \\ &= \begin{cases} -\partial_{j_r}^\Lambda u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - \partial_{j_r}^\Lambda u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad \text{if } d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad \leq x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is} \\ -\partial_{j_r}^\Lambda u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - \partial_{j_r}^\Lambda u_{1s}(x_t, d_t, \alpha_t, \lambda) - \mathbf{1}(j = i, r = s) \\ \quad \text{if } d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad > x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (12) \end{aligned}$$

We emphasize that the expression above is accurate only in w.p.1 sense. In particular, there are events under which the decision function is not differentiable. However, Lemma 2 in the appendix shows that these events occur with probability zero and we can define $\partial_{j_r}^\Lambda u_{is}(x_t, d_t, \alpha_t, \lambda)$ arbitrarily under these events. Similarly, differentiating (2) with respect to the remaining capacity on the r th day of the booking horizon, we obtain

$$\begin{aligned} & \partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda) \\ &= \begin{cases} -\partial_r^X u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - \partial_r^X u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad \text{if } d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad \leq x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is} \\ \mathbf{1}(r = s) - \partial_r^X u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - \partial_r^X u_{1s}(x_t, d_t, \alpha_t, \lambda) \\ \quad \text{if } d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ \quad > x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (13) \end{aligned}$$

Similar to (12), (13) is accurate only in w.p.1 sense.

We can compute $\partial_{j_r}^\Lambda C_1(x_1, d, \alpha, \lambda)$ simply by simulating the decisions of the policy characterized by protection levels λ under job arrivals d and capacity perturbations α . In this case, the capacities $X_{t+1}(x_1, d, \alpha, \lambda)$ over the booking horizon on day $t + 1$ are given recursively by

$$X_{t+1}(x_1, d, \alpha, \lambda) = R(X_t(x_1, d, \alpha, \lambda), d_t, \alpha_t, \lambda),$$

with $X_1(x_1, d, \alpha, \lambda) = x_1$. Using (12) and (13), we can compute $\partial_{j_r}^\Lambda u_{is}(X_t(x_1, d, \alpha, \lambda), d_t, \alpha_t, \lambda)$ and $\partial_r^X u_{is}(X_t(x_1, d, \alpha, \lambda), d_t, \alpha_t, \lambda)$ for all $i, j \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$. Similarly, using (10) and (11), we can compute $\partial_{j_r}^\Lambda R_s(X_t(x_1, d, \alpha, \lambda), d_t, \alpha_t, \lambda)$ and $\partial_r^X R_s(X_t(x_1, d, \alpha, \lambda), d_t, \alpha_t, \lambda)$ for all $j \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$. We can compute $\partial_r^X C_t(X_t(x_1, d, \alpha, \lambda), d, \alpha, \lambda)$ for all $r \in \mathcal{S}$, $t \in \mathcal{T}$ by using (9) and moving backwards over the planning horizon. Finally, we can compute $\partial_{j_r}^\Lambda C_t(X_t(x_1, d, \alpha, \lambda), d, \alpha, \lambda)$ for all $j \in \mathcal{N}$, $r \in \mathcal{S}$, $t \in \mathcal{T}$ by using (8) and moving backwards over the planning horizon. We emphasize that all of these sample path-based derivatives exist only in w.p.1 sense, since the decision function is differentiable only in w.p.1 sense.

4 AN ALGORITHM FOR THE SMOOTH PROBLEM

We propose the following algorithm to solve problem (6).

Algorithm 1 (A1)

Step 1 Initialize the iteration counter k to 1. Set the initial protection levels $\lambda^1 = \{\lambda_{is}^1 : i \in \mathcal{N}, s \in \mathcal{S}\}$ such that $0 \leq \lambda_{is}^1 \leq c$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$.

Step 2 Letting d^k be the job arrivals and α^k be the capacity perturbations at iteration k , compute $\partial_{is}^\Lambda C_1(x_1, d^k, \alpha^k, \lambda^k)$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$ by using (8)-(13).

Step 3 Letting σ^k be a step size parameter, compute the protection levels $\lambda^{k+1} = \{\lambda_{is}^{k+1} : i \in \mathcal{N}, s \in \mathcal{S}\}$ at the next iteration as

$$\lambda_{is}^{k+1} = \min \{ [\lambda_{is}^k - \sigma^k \partial_{is}^\Lambda C_1(x_1, d^k, \alpha^k, \lambda^k)]^+, c \}. \quad (14)$$

Step 4 Increase k by 1 and go to Step 2.

Algorithm 1 is a standard stochastic gradient search method driven by the random step direction $\{\partial_{is}^\Lambda C_1(x_1, d^k, \alpha^k, \lambda^k) : i \in \mathcal{N}, s \in \mathcal{S}\}$ at iteration k . The initial protection levels in Step 1 are set arbitrarily as long as they satisfy $0 \leq \lambda_{is}^1 \leq c$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. In our implementation, we set $\lambda_{is}^1 = 0$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. Letting \mathcal{F}^k be the filtration generated by the random variables $\{\lambda^1, d^1, \dots, d^{k-1}, \alpha^1, \dots, \alpha^{k-1}\}$, we assume that the joint distribution of (d^k, α^k) in Step 2 conditional on \mathcal{F}^k is the same as the joint distribution of (d, α) . The updating procedure in (14) in Step 3 ensures that the protection levels at every iteration are in the interval $[0, c]$ and the operator $\min\{[\cdot]^+, c\}$ can be interpreted as a projection onto this interval. The step size parameter σ^k plays an important role in the empirical convergence rate of Algorithm 1 and its choice requires some experimentation. In our implementation, we settle on $\sigma^k = 2/(40 + k)$ after some setup runs. It is possible to adopt a variety of stopping criteria in Algorithm 1. We can stop Algorithm 1 when we reach a fixed iteration limit, or when the step size parameter gets below a fixed threshold, or when the difference between the protection levels in two successive iterations gets below a fixed threshold. In our implementation, we stop Algorithm 1 when we reach a fixed iteration limit. We have the next convergence result for Algorithm 1.

Proposition 1 *Assume that the sequence of protection levels $\{\lambda^k\}$ is generated by Algorithm 1 and the sequence of step size parameters $\{\sigma^k\}$ satisfies $\sigma^k \geq 0$ for all $k \geq 1$, $\sum_{k=1}^\infty \sigma^k = \infty$ and $\sum_{k=1}^\infty [\sigma^k]^2 < \infty$. If we use Θ to denote the set of Kuhn-Tucker points for problem (6) and Θ is connected, then we have $\lambda^k \rightarrow \Theta$ in probability as $k \rightarrow \infty$.*

Proof Propositions 4, 5 and 6 in Appendices C, D and E show that the following statements hold.

(S.1) We have $|\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda)| \leq B_C^\Lambda$ w.p.1 for all $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$ for a finite scalar B_C^Λ .

(S.2) Using $\partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}$ to denote the derivative of $\mathbb{E}\{C_1(x_1, d, \alpha, \cdot)\}$ with respect to the protection level for priority i jobs on the s th day of the booking horizon evaluated at protection levels λ , we have

$$\mathbb{E}\{\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda)\} = \partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}$$

for all $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$.

(S.3) Using $\|\cdot\|$ to denote the Euclidean norm, we have

$$\mathbb{E}\{|\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda) - \partial_{is}^\Lambda C_1(x_1, d, \alpha, \kappa)|\} \leq L_C^\Lambda \|\lambda - \kappa\|$$

for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$ for a finite scalar L_C^Λ .

In this case, the result follows from Theorem 6.3.1 in Kushner and Clark (1978), which we briefly state in Appendix B for completeness. In particular, (S.2) and (S.3) imply that

$$\begin{aligned} |\partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\} - \partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \kappa)\}| &= |\mathbb{E}\{\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda) - \partial_{is}^\Lambda C_1(x_1, d, \alpha, \kappa)\}| \\ &\leq \mathbb{E}\{|\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda) - \partial_{is}^\Lambda C_1(x_1, d, \alpha, \kappa)|\} \leq L_C^\Lambda \|\lambda - \kappa\| \end{aligned}$$

so that the objective function of problem (6) satisfies (B.1) in Appendix B. Letting $\bar{C}_1(x_1, \lambda) = \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}$ and using the fact that the joint distribution of (d^k, α^k) conditional on \mathcal{F}^k is the same as the joint distribution of (d, α) , (S.2) implies that we have $\mathbb{E}\{\partial_{is}^\Lambda C_1(x_1, d^k, \alpha^k, \lambda^k) | \mathcal{F}^k\} = \partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d^k, \alpha^k, \lambda^k) | \mathcal{F}^k\} = \partial_{is}^\Lambda \bar{C}_1(x_1, \lambda^k)$. On the other hand, we have $|\partial_{is}^\Lambda C_1(x_1, d^k, \alpha^k, \lambda^k) - \partial_{is}^\Lambda \bar{C}_1(x_1, \lambda^k)| \leq 2B_C^\Lambda$ w.p.1 by (S.1) and (S.2). Therefore, (B.2) in Appendix B is satisfied. Since $c > 0$, the constraints $\lambda_{is} \geq 0$ and $\lambda_{is} \leq c$ in the feasible set in problem (6) cannot be simultaneously active. Therefore, (B.3) in Appendix B is satisfied and the result follows. \square

(S.1) implies that the norm of the step direction that we use in Step 3 of Algorithm 1 is uniformly bounded. (S.2) implies that the expected value of the step direction is an ascent direction of the objective function of problem (6). (S.3) implies that the expected value of the step direction is Lipschitz.

A shortcoming of Proposition 1 is that it is not easy to verify the connectedness of the set of Kuhn-Tucker points for problem (6). Theorem 6.3.1 in Kushner and Clark (1978) also provides a weaker but more technical convergence result without the assumption that the set of Kuhn-Tucker points is connected. Roughly speaking, this result focuses on the averages of the protection levels obtained over a certain number of iterations in Algorithm 1 and shows that the probability that these averages lie away from the set of Kuhn-Tucker points for an extended period of time diminishes as the number of iterations goes to infinity. We do not go into the details of this weaker result, but emphasize that Algorithm 1 can still be shown to have a convergence guarantee when the set of Kuhn-Tucker points for problem (6) is not necessarily connected.

We note that all of the statements in the proof of Proposition 1 go through with no modifications when the job arrivals on different days are dependent. Therefore, Algorithm 1 can still be used to find protection levels when the job arrivals are correlated.

5 PERFECT HINDSIGHT LOWER BOUNDS

It is clearly difficult to compute the optimal policy for practical instances of the capacity allocation problem. In this section, we show how to obtain lower bounds on the optimal total expected cost. Such lower bounds become useful when establishing the optimality gap of a suboptimal policy. The method that we use to obtain lower bounds on the optimal total expected cost use the perfect hindsight linear program. The idea behind the perfect hindsight linear program is to make the decisions after observing all of the job arrivals. In particular, letting u_{ist} be the number of priority i jobs that we schedule on the s th day of the booking horizon on day t , if the job arrivals over the whole planning horizon are given by $d = \{d_{it} : i \in \mathcal{N}, t \in \mathcal{T}\}$, then the perfect hindsight linear program can be written as

$$\min \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} \tilde{h}_{is} u_{ist} + \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \tilde{b}_i \left[d_{it} - \sum_{s \in \mathcal{S}} u_{ist} \right] \quad (15)$$

$$\text{subject to } \sum_{i \in \mathcal{N}} u_{is1} + \dots + \sum_{i \in \mathcal{N}} u_{i1s} \leq x_{s1} \quad \text{for all } s = 1, \dots, S \quad (16)$$

$$\sum_{i \in \mathcal{N}} u_{iS,t-S+1} + \dots + \sum_{i \in \mathcal{N}} u_{i1t} \leq c \quad \text{for all } t = S+1, \dots, T \quad (17)$$

$$\sum_{s \in \mathcal{S}} u_{ist} \leq d_{it} \quad \text{for all } i \in \mathcal{N}, t \in \mathcal{T} \quad (18)$$

$$u_{ist} \geq 0 \quad \text{for all } i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}. \quad (19)$$

In the problem above, constraints (16) ensure that the numbers of jobs that we schedule on the first S days of the planning horizon do not violate the remaining capacities, whereas constraints (17) ensure that the numbers of jobs that we schedule on the last $T - S$ days of the planning horizon do not violate the remaining capacities. The initial capacities $x_1 = \{x_{s1} : s \in \mathcal{S}\}$ in constraints (16) are a part of the problem data. In constraints (16), we use the observation that the s th day of the booking horizon on day 1 and the first day of the booking horizon on day s correspond to the same day. We use a similar observation in constraints (17). Constraints (18) ensure that the numbers of jobs that we schedule do not exceed the job arrivals.

As a function of d , we let $z(d)$ be the optimal objective value of problem (15)-(19) and $\pi^*(d)$ be the total cost incurred by the optimal policy when the job arrivals over the whole planning horizon are given by d . Since problem (15)-(19) makes the decisions after observing all of the job arrivals, we naturally have $z(d) \leq \pi^*(d)$ and we obtain $\mathbb{E}\{z(d)\} \leq \mathbb{E}\{\pi^*(d)\}$ by taking expectations. Noting that $z(d)$ is a convex function of d by linear programming duality, Jensen's inequality also implies that $z(\mathbb{E}\{d\}) \leq \mathbb{E}\{z(d)\}$ and we obtain $z(\mathbb{E}\{d\}) \leq \mathbb{E}\{z(d)\} \leq \mathbb{E}\{\pi^*(d)\}$. Therefore, both $z(\mathbb{E}\{d\})$ and $\mathbb{E}\{z(d)\}$ provide lower bounds on the optimal total expected cost. The lower bound provided by $z(\mathbb{E}\{d\})$ is looser, but this lower bound can be computed simply by replacing d with $\mathbb{E}\{d\}$ in problem (15)-(19) and solving this problem once. On the other hand, the lower bound provided by $\mathbb{E}\{z(d)\}$ is tighter, but computing the expectation $\mathbb{E}\{z(d)\}$ requires estimation through simulation. In particular, we solve problem (15)-(19) for many samples of d and estimate $\mathbb{E}\{z(d)\}$ through a sample average. Nevertheless, our computational experiments indicate that the gap between $z(\mathbb{E}\{d\})$ and $\mathbb{E}\{z(d)\}$ can be significant and it can be worthwhile to estimate the expectation $\mathbb{E}\{z(d)\}$ through simulation.

6 COMPUTATIONAL EXPERIMENTS

In this section, we test the performance of the protection levels computed by Algorithm 1.

6.1 BENCHMARK STRATEGIES

When dealing with a problem that takes place under uncertainty, a common engineering practice is to compute the optimal decisions under the assumption that the future random variables take their expected values. This is the approach that we follow in our benchmark strategy. In particular, given the remaining capacities $x_t = \{x_{st} : s \in \mathcal{S}\}$ over the booking horizon and the job arrivals $d_t = \{d_{it} : i \in \mathcal{N}\}$ on day t , we make the decisions by solving a linear program. This linear program finds the optimal decisions over the next L days under the assumption that the job arrivals on days $\{t+1, \dots, t+L-1\}$ take their expected values. This approach is frequently used in practice and it is known as the rolling horizon strategy. The set of time periods $\{t, \dots, t+L-1\}$ is referred to as the rolling horizon on day t and L is the length of the rolling horizon. The length of the rolling horizon captures how far into the future we look when we make the decisions at a particular time period. Although it is not guaranteed, the performance of the rolling horizon strategy is expected to improve with increasing L . However, the rolling horizon strategy also gets computationally more demanding with increasing L and L is usually chosen significantly smaller than T . We let $\mathcal{L}_t = \{t, \dots, t+L-1\}$ for notational brevity.

In this case, letting $u_{is\tau}$ be the number of priority i jobs that we schedule on the s th day of the booking horizon on day τ , the rolling horizon strategy solves the linear program

$$\min \quad \sum_{\tau \in \mathcal{L}_t} \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} \tilde{h}_{is} u_{is\tau} + \sum_{\tau \in \mathcal{L}_t} \sum_{i \in \mathcal{N}} \tilde{b}_i \left[\bar{d}_{i\tau} - \sum_{s \in \mathcal{S}} u_{is\tau} \right] \quad (20)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{N}} u_{ist} + \dots + \sum_{i \in \mathcal{N}} u_{i1,t+s-1} \leq x_{st} \quad \text{for all } s = 1, \dots, S \quad (21)$$

$$\sum_{i \in \mathcal{N}} u_{iS,t+s-S} + \dots + \sum_{i \in \mathcal{N}} u_{i1,t+s-1} \leq c \quad \text{for all } s = S+1, \dots, L \quad (22)$$

$$\sum_{s \in \mathcal{S}} u_{is\tau} \leq \bar{d}_{i\tau} \quad \text{for all } i \in \mathcal{N}, \tau \in \mathcal{L}_t \quad (23)$$

$$u_{is\tau} \geq 0 \quad \text{for all } i \in \mathcal{N}, s \in \mathcal{S}, \tau \in \mathcal{L}_t \quad (24)$$

to make the decisions on day t . In the problem above, we let $\bar{d}_{it} = d_{it}$ for all $i \in \mathcal{N}$ and $\bar{d}_{i\tau} = \mathbb{E}\{d_{i\tau}\}$ for all $i \in \mathcal{N}, \tau \in \mathcal{L}_t \setminus \{t\}$ so that we use the actual job arrivals on day t , but the expected values of the job arrivals on days $\{t+1, \dots, t+L-1\}$. Constraints (21) and (22) ensure that the decisions do not violate the remaining capacities. We note that the remaining capacities over the booking horizon on day t are given by $x_t = \{x_{st} : s \in \mathcal{S}\}$, but the remaining capacities beyond the booking horizon are always c . In constraints (21), we use the observation that the s th day of the booking horizon on day t and the first day of the booking horizon on day $t+s-1$ correspond to the same day. We use a similar observation in constraints (22). After solving the problem above, we implement the decisions only for day t given by the decision variables $\{u_{ist} : i \in \mathcal{N}, s \in \mathcal{S}\}$ and solve a similar linear program when making the decisions for the next day. Although the rolling horizon strategy does not entirely capture the randomness in the problem, we expect it to perform reasonably well since we solve the linear program above after observing the realizations of the job arrivals on each day.

The other benchmark strategy that we use schedules the jobs on a first come first serve basis starting with the highest priority jobs and the earliest day in the booking horizon. In other words, this benchmark strategy does not protect capacity for the future job arrivals and it is equivalent to using a protection level policy with zero protection levels. This benchmark strategy is likely to perform poorly, but we use it to demonstrate the importance of explicitly accounting for the future job arrivals.

6.2 EXPERIMENTAL SETUP

In our experimental setup, we generate a base problem and modify its different attributes to generate test problems with different characteristics. In the base problem, we have $T = 100$, $N = 3$, $S = 7$ and $c = 70$. The holding costs are $h_{is} = 2^{3-i} 1.25^{s-1}$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$ and the penalty costs are $b_i = 2^{5-i} 1.25^6$ for all $i \in \mathcal{N}$. The daily job arrivals for the first, second and third priority levels are uniformly distributed respectively over $[5, 15]$, $[10, 30]$ and $[20, 60]$. Although the results that we present are under uniformly distributed job arrivals, our observations remain the same under a variety of distributions that we experimented with, including the normal and lognormal distribution.

We work with two variants of Algorithm 1. In the first variant, we use Algorithm 1 once at the beginning of the planning horizon to compute a set of protection levels and use these protection levels throughout. In the second variant, we divide the planning horizon into five equal segments and use Algorithm 1 at the beginning of each segment to compute a new set of protection levels. In particular, given the remaining capacities $x_{1+(n-1)T/5}$ over the booking horizon at the beginning of segment n , the second variant solves the problem $\min_{0 \leq \lambda \leq c} \mathbb{E}\{C_{1+(n-1)T/5}(x_{1+(n-1)T/5}, d, \alpha, \lambda)\}$ to compute a new set of protection levels and uses these protection levels until we reach segment $n + 1$. The goal of the second variant is to mimic a situation where the protection levels are retuned periodically as the remaining capacities change and the hope is that retuning the protection levels periodically improves the performance. We note that when computing the protection levels at the beginning of segment $n + 1$, we can use the protection levels computed at the beginning of segment n as the initial solution. We tried to implement this idea, but it did not provide any noticeable improvement over starting with the initial solution of zeros. We also tried to retune the protection levels more than five times over the planning horizon, but this extra effort provided barely noticeable improvements. Throughout the paper, we refer to the first and second variants of Algorithm 1 respectively as A1 and A1-5. We use the step size parameter $\sigma^k = 2/(40 + k)$ for A1 and A1-5. We run A1 and A1-5 for 1,000 iterations.

We refer to the rolling horizon strategy as RH. We tested using $L = 15, 25, 50, 75$ in several setup runs and a rolling horizon length of 50 provided the best results. Therefore, we use $L = 50$ in all of our computational experiments. We refer to the first come first serve strategy as FC.

We round the protection levels obtained by A1 and A1-5 to the nearest integers when testing the performance of a protection level policy. This ensures that the numbers of jobs that we schedule on each day of the booking horizon take integer values. It is possible to show that the linear program that RH solves is a min-cost network flow problem. Therefore, RH also schedules integer numbers of jobs on each day of the booking horizon as long as the daily available capacity and the expected values of the job arrivals take integer values.

6.3 COMPUTATIONAL RESULTS

Table 1 shows the computational results for the base problem. The second, third, fourth and fifth columns in this table respectively show the total expected costs incurred by A1, A1-5, RH and FC. We estimate these total expected costs by simulating the performances of the different methods under 100 job arrival trajectories. We use common random numbers to reduce the effect of noise when simulating the performances of the different methods; see Law and Kelton (2000). The sixth column shows the percent gap between the total expected costs incurred by A1-5 and A1. The seventh and eighth columns do the same thing, but they focus on the performance gap between A1-5 and the remaining two benchmark methods. A1-5 turns out to be one of the better benchmark methods and we use it as a reference point. The ninth, tenth, eleventh and twelfth columns respectively show what percents of the lowest priority jobs are rejected by A1, A1-5, RH and FC. In our test problems, higher priority jobs are almost never rejected and we only provide statistics for the lowest priority jobs. The thirteenth and fourteenth columns show the lower bounds on the optimal total expected cost provided by the perfect hindsight linear program. The fifteenth column shows the percent gap between the total expected cost incurred by A1-5 and the tighter lower bound on the optimal total expected cost. In other words, this column shows the percent gap between the third and fourteenth columns and it gives a feel for the maximum possible gap between the performances of A1-5 and the optimal policy.

The results indicate that A1 performs noticeably better than RH despite the fact that A1 computes a set of protection levels at the beginning of the planning horizon and uses these protection levels throughout, whereas RH solves a linear program on each day. FC performs substantially worse than A1 and RH showing the importance of protecting capacity for the future job arrivals. Comparing the performances of A1 and A1-5 indicates that retuning the protection levels multiple times over the planning horizon improves the performance by a small but consistent margin. Comparing the rejected jobs indicates that A1-5 rejects smaller number of jobs than A1 and A1 rejects smaller number of jobs than RH. FC rejects the smallest number of jobs, but since FC schedules the jobs without protecting capacity for the future job arrivals, it tends to allow the high priority jobs to wait for a long time in the system. Therefore, rejecting a smaller number of jobs does not immediately translate to more satisfactory performance. As mentioned in Section 5, the lower bound $z(\mathbb{E}\{d\})$ is looser than the lower bound $\mathbb{E}\{z(d)\}$. For the base problem, the gap between the two lower bounds is 4.76%. The last column indicates that the performance of A1-5 is quite satisfactory as the total expected cost incurred by A1-5 deviates from the optimal total expected cost by at most 2.15%.

In the first chart in Figure 1, we plot $\mathbb{E}\{C_1(x_1, d, \alpha, \lambda^k)\}$ for the base problem as a function of the iteration counter k for A1. We estimate the expectation $\mathbb{E}\{C_1(x_1, d, \alpha, \lambda^k)\}$ through simulation. This figure indicates that the quality of the protection levels stabilizes in 50 iterations. Nevertheless, to compensate for the lack of good stopping criteria for stochastic approximation methods in general, we run A1 and A1-5 for 1,000 iterations. In the second chart in Figure 1, we plot the protection levels obtained by A1 for different days of the booking horizon in the base problem. For our test problems, it is never desirable to protect capacity when making the decisions for the first priority level and we always have $\lambda_{1s} = 0$ for all $s \in \mathcal{S}$. Therefore, we only plot the protection levels for the second and third priority levels. We observe two patterns. First, the protection levels for the later days of the booking

horizon tend to be larger. Intuitively speaking, there are more opportunities to utilize the capacity on the later days before the capacity on these days expires. Therefore, we are willing to protect a larger portion of the capacity on the later days for the future high priority job arrivals. Second, the protection levels for the third priority level are larger than the protection levels for the second priority level. This is due to the fact that the portion of the capacity that we protect when making the decisions for the third priority jobs is at least as large as the portion of the capacity that we protect when making the decisions for the second priority jobs. We note that the protection level on the last day of the booking horizon is zero for the second priority level. This is simply due to the fact that the second priority jobs never wait more than 6 days and the last protection level for them is essentially inconsequential.

In the first chart in Figure 2, we plot the protection levels in the base problem after increasing the holding costs for the second priority level by a factor of two. Comparing this chart with the second chart in Figure 1 indicates that the protection levels for the second priority level decrease. This is due to the fact that since the holding costs for the second priority jobs are higher, we are not willing to hold the second priority jobs in the system for long periods of time and we protect a smaller portion of the capacity when making the decisions for the second priority jobs. In the second chart in Figure 2, we plot the protection levels in the base problem after increasing the penalty cost for the third priority level by a factor of two. Comparing this chart with the second chart in Figure 1 indicates that the protection levels for the third priority level decrease. This is due to the fact that since the penalty cost for the third priority jobs is higher, we are willing to reject fewer third priority jobs and we protect a smaller portion of the capacity when making the decisions for the third priority jobs.

In Tables 2-7, we modify different attributes of the base problem to generate test problems with different characteristics. The first column in these tables indicates the value of the problem attribute that we modify. The other columns have the same interpretations as those in Table 1.

In Table 2, we vary the number of days in the booking horizon. The performance gap between A1-5 and RH is larger for the test problems with shorter booking horizons. Although one expects the role of uncertainty to grow when we have longer booking horizons, longer booking horizons also give us the opportunity to correct for mistakes and both A1 and RH perform well for longer booking horizons.

In Table 3, we vary the number of priority levels. The test problem with two priority levels appears to be easy and even FC performs quite well on this test problem. Furthermore, noting the lower bound on the optimal total expected cost, the performances of A1, A1-5, RH and FC are essentially optimal for this test problem. However, A1 and A1-5 consistently perform better than RH when we have three or more priority levels.

In Table 4, we vary the coefficient of variation for the job arrivals. A1 and A1-5 perform better than RH when the coefficient of variation is large. Therefore, addressing the randomness explicitly pays off when the job arrivals are highly variable. We note that the performance gap between A1-5 and RH can be as high as 11.82% when the coefficient of variation is large.

In Table 5, we vary the daily available capacity. It appears that A1 and A1-5 are particularly effective when the daily available capacity is tight. This is encouraging since the test problems with tight capacities tend to require more careful planning.

In Tables 6 and 7, we vary the holding and penalty costs. We work with holding costs of the form $\tilde{h}_{is} = \phi^{3-i} 1.25^{s-1}$ and penalty costs of the form $\tilde{b}_i = \beta 2^{3-i} 1.25^6$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$ and vary the values of ϕ and β . The base problem corresponds to $\phi = 2$ and $\beta = 4$. The results indicate that A1 and A1-5 generally perform better than RH and small holding and penalty costs favor A1 and A1-5.

Figure 3 shows the CPU seconds required to compute a set of protection levels as a function of the number of priority levels and the length of the booking horizon. Overall, A1 and A1-5 take a few seconds to compute a set of protection levels. The CPU seconds scale approximately quadratically with N and S . This is expected since A1 and A1-5 spend a large amount of time to compute the sample path-based derivative $\partial_{j_r}^\Lambda u_{is}^\lambda(x_t^\lambda, d_t, \alpha_t, \lambda)$ for all $i, j \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$ and the number of these sample path-based derivatives increases quadratically with N and S .

7 CONCLUSIONS

In this paper, we considered a capacity allocation problem that involves allocating a fixed amount of daily processing capacity among jobs of different priority levels that arrive randomly over time. We focused on a class of policies that are characterized by protection levels and developed a stochastic approximation method to find a good set of protection levels. Computational results indicated that the protection levels computed by our stochastic approximation method perform consistently better than using a deterministic approximation and the performance gap can be as high as 11.82% when the coefficient of variation for the job arrivals is high. In all of our test problems, the performance gap between the protection levels obtained by our method and the optimal policy is at most 4.08%.

A straightforward but possibly useful extension of our approach involves using a different set of protection levels to make the decisions on different days. In particular, we can use the protection levels $\{\lambda_{ist} : i \in \mathcal{N}, s \in \mathcal{S}\}$ to make the decisions on day t . In this case, the total expected cost becomes a function of $\lambda = \{\lambda_{ist} : i \in \mathcal{N}, s \in \mathcal{S}, t \in \mathcal{T}\}$, but we can still compute the sample path-based derivatives of the total expected cost function by using a recursion that is very similar to the one in (8)-(13). This extension may be particularly useful when the job arrivals are nonstationary.

REFERENCES

- Bashyam, S. and Fu, M. C. (1998), ‘Optimization of (s, S) inventory systems with random lead times and a service level constraint’, *Management Science* **44**(12), 243–256.
- Bertsekas, D. P. and Tsitsiklis, J. N. (1996), *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.
- Bertsimas, D. and de Boer, S. (2005), ‘Simulation-based booking limits for airline revenue management’, *Operations Research* **53**(1), 90–106.
- Brumelle, S. L. and McGill, J. I. (1993), ‘Airline seat allocation with multiple nested fare classes’, *Operations Research* **41**, 127–137.
- Fu, M. (1994), ‘Sample path derivatives for (s, S) inventory systems’, *Operations Research* **42**(2), 351–363.
- Gerchak, Y., Gupta, D. and Henig, M. (1996), ‘Reservation planning for elective surgery under uncertain demand for emergency surgery’, *Management Science* **42**, 321–334.

- Glasserman, P. (1994), Perturbation analysis of production networks, in D. D. Yao, ed., ‘*Stochastic Modeling and Analysis of Manufacturing Systems*’, Springer, New York, NY.
- Glasserman, P. and Tayur, S. (1995), ‘Sensitivity analysis for base-stock levels in multiechelon production-inventory systems’, *Management Science* **41**(2), 263–281.
- Gupta, D. and Wang, L. (2007), ‘Revenue management for a primary-care clinic in presence of patient choice’, *Operations Research* (to appear).
- Karaesmen, I. and van Ryzin, G. (2004), ‘Overbooking with substitutable inventory classes’, *Operations Research* **52**(1), 83–104.
- Kunnumkal, S. and Topaloglu, H. (2007), ‘Using stochastic approximation algorithms to compute optimal base-stock levels in inventory control problems’, *Operations Research* (to appear).
- Kushner, H. J. and Clark, D. S. (1978), *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, Berlin.
- Law, A. L. and Kelton, W. D. (2000), *Simulation Modeling and Analysis*, McGraw-Hill, Boston, MA.
- L’Ecuyer, P. and Glynn, P. (1994), ‘Stochastic optimization by simulation: Convergence proofs for the GI/G/1 queue in steady state’, *Management Science* **40**, 1245–1261.
- Littlewood, K. (1972), Forecasting and control of passengers, in ‘Proceedings of the 12th AGIFORS’, New York, pp. 95–117.
- Mahajan, S. and van Ryzin, G. (2001), ‘Stocking retail assortments under dynamic customer substitution’, *Operations Research* **49**(3), 334–351.
- Patrick, J., Puterman, M. and Queyranne, M. (2008), ‘Dynamic multi-priority patient scheduling for a diagnostic resource’, *Operations Research* (to appear).
- Phillips, R. L. (2005), *Pricing and Revenue Optimization*, Stanford University Press, Stanford, CA.
- Talluri, K. T. and van Ryzin, G. J. (2004), *The Theory and Practice of Revenue Management*, Kluwer Academic Publishers.
- van Ryzin, G. and Vulcano, G. (2004), Computing virtual nesting controls for network revenue management under customer choice behavior, Technical Report DRO-2004-09, Columbia Business School.
- van Ryzin, G. and Vulcano, G. (2006), ‘Simulation-based optimization of virtual nesting controls for network revenue management’, *Operations Research* (to appear).
- Wollmer, R. D. (1992), ‘An airline seat management model for a single leg route when lower fare classes book first’, *Operations Research* **40**, 26–37.

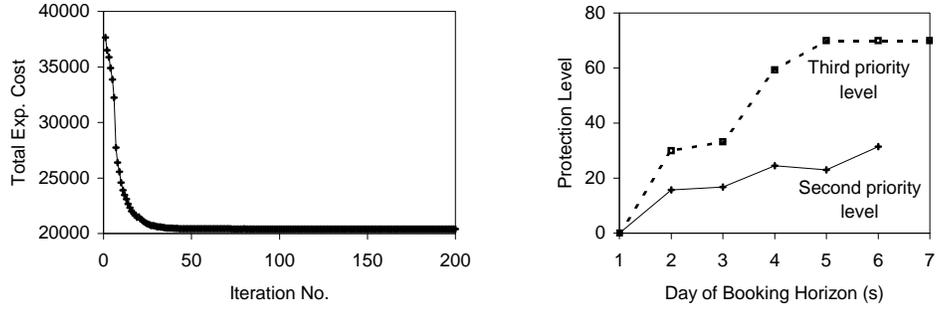


Figure 1: Total expected cost incurred by the protection levels obtained by A1 as a function of the iteration counter (left) and the protection levels obtained by A1 (right).

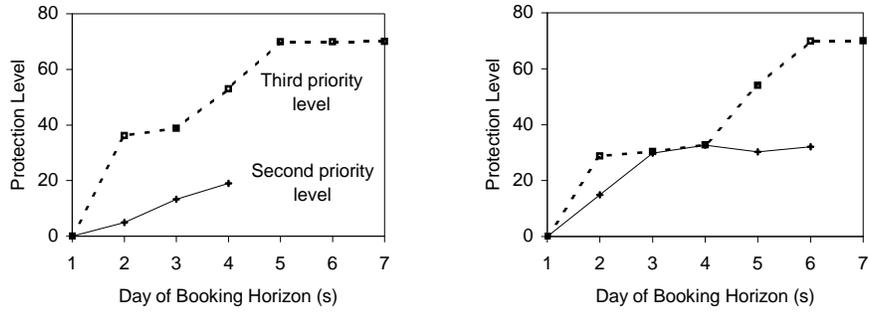


Figure 2: Protection levels obtained by A1 when we increase the holding costs for the second priority level (left) and when we increase the penalty cost for the third priority level (right).

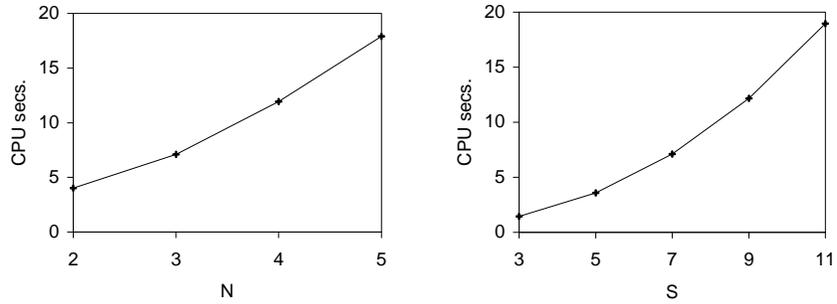


Figure 3: CPU seconds required to compute a set of protection levels as a function N and S .

Table 1: Computational results for the base problem.

Test Prob.	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
Base	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15

Table 2: Computational results with varying number of days in the booking horizon.

S	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
3	13,976	13,957	15,658	15,905	0.14	12.19	13.96	5.03	5.86	8.58	1.21	12,810	13,694	1.92
5	16,166	16,089	19,381	24,115	0.48	20.47	49.89	4.17	3.77	8.05	1.18	14,791	15,769	2.02
7	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
9	27,796	27,756	28,003	58,840	0.15	0.89	111.99	3.60	3.03	3.63	1.17	26,703	27,212	2.00
11	41,565	40,939	41,785	91,937	1.53	2.07	124.57	3.69	2.85	2.84	1.17	39,995	40,352	1.46

Table 3: Computational results with varying number of priority levels.

N	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
2	4,450	4,442	4,441	4,450	0.17	-0.02	0.19	0.00	0.00	0.00	0.00	4,440	4,440	0.04
3	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
4	47,942	47,733	48,903	89,173	0.44	2.45	86.82	4.69	4.31	6.08	0.47	44,695	46,471	2.72
5	120,323	118,730	122,268	245,209	1.34	2.98	106.53	5.27	5.45	7.29	1.14	113,758	117,196	1.31

Table 4: Computational results with varying coefficient of variation values.

CV	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
0.1	19,313	19,236	19,303	41,835	0.40	0.34	117.48	5.17	4.97	5.13	0.39	18,898	19,129	0.56
0.2	19,808	19,686	19,984	39,566	0.62	1.51	100.98	5.23	4.89	5.78	0.79	18,898	19,429	1.33
0.3	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
0.4	21,024	20,837	22,492	35,992	0.90	7.94	72.73	5.54	5.02	7.96	1.58	18,898	20,223	3.03
0.5	21,743	21,542	24,088	34,594	0.93	11.82	60.59	5.78	5.19	9.34	1.97	18,898	20,698	4.08

Table 5: Computational results with varying daily available capacities.

c	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
55	37,840	36,210	39,026	61,078	4.50	7.78	68.68	21.34	18.57	22.70	17.30	34,879	34,996	3.47
70	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
110	17,098	17,064	17,049	20,767	0.20	-0.09	21.70	2.28	2.45	2.40	0.09	16,735	16,934	0.77
120	15,805	15,783	15,754	16,730	0.14	-0.19	6.00	0.61	1.13	1.04	0.02	15,616	15,709	0.48

Table 6: Computational results with varying holding costs.

ϕ	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
1.2	15,934	15,740	16,684	26,161	1.23	6.00	66.20	5.22	4.80	6.37	1.18	14,474	15,356	2.50
2.0	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
3.0	28,159	28,001	28,850	57,985	0.57	3.03	107.09	5.39	4.95	6.75	1.18	26,615	27,524	1.73
4.0	38,153	37,932	38,774	84,025	0.58	2.22	121.51	5.48	4.97	6.75	1.18	36,502	37,420	1.37

Table 7: Computational results with varying penalty costs.

β	Total Expected Cost				% Gap with A1-5			% Reject. Jobs				Low. Bnd.		A1-5 % Gap
	A1	A1-5	RH	FC	A1	RH	FC	A1	A1-5	RH	FC	$z(\mathbb{E}\{d\})$	$\mathbb{E}\{z(d)\}$	
2.0	16,896	16,851	19,008	36,936	0.27	12.80	119.18	6.17	5.84	11.11	1.18	15,629	16,509	2.07
4.0	20,381	20,224	21,098	37,657	0.77	4.32	86.20	5.35	4.90	6.75	1.18	18,898	19,798	2.15
8.0	26,428	26,021	26,590	39,101	1.56	2.19	50.27	4.58	3.94	3.96	1.18	24,991	25,407	2.42

A TECHNICAL APPENDIX: DIFFERENTIABILITY OF THE DECISION FUNCTION

Lemma 2 *At fixed remaining capacities \hat{x}_t and protection levels $\hat{\lambda}$, the decision function $u_{is}(\cdot, d_t, \alpha_t, \cdot)$ in (2) is differentiable with respect to the remaining capacities and protection levels for almost all realizations of d_t and α_t .*

Proof We only show the differentiability with respect to the protection levels. The differentiability with respect to the remaining capacities can be handled by using a similar argument. We show the result by induction over the priority levels and the days of the booking horizon. In particular, for all $1 \leq j \leq i$ and $1 \leq r \leq s$ with $(j, r) \neq (i, s)$, we assume that $u_{jr}(\hat{x}_t, d_t, \alpha_t, \cdot)$ is differentiable at protection levels $\hat{\lambda}$ w.p.1 and we show that the same result holds for $u_{is}(\hat{x}_t, d_t, \alpha_t, \cdot)$. We consider numerous events whose total probability is 1.

Case 1 Under the event $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) < \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is}$, we have $u_{is}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) = d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda})$ by (2). The decision function is a continuous function of the protection levels. This implies that if we let e_{is} be the $N \times S$ dimensional unit vector with a 1 in the element corresponding to i and s , then we still have $u_{is}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) = d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is})$ for all small enough ϵ . Since for all $1 \leq r \leq s-1$, $u_{ir}(\hat{x}_t, d_t, \alpha_t, \cdot)$ is differentiable at protection levels $\hat{\lambda}$ w.p.1 by the induction assumption, the result holds under the event mentioned at the beginning of this case.

Case 2 Under the events $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) > \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is} > 0$ or $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) > 0 > \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is}$, it is possible to show that the result holds by using the same argument in Case 1.

Case 3 Under the event $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) = 0 > \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is}$, we have $u_{is}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) = 0$ by (2). Since the decision function is a continuous function of the protection levels, we have $\hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) - [\hat{\lambda}_{is} + \epsilon] < 0$ for all small enough ϵ . Noting that the decision function never violates the job availabilities, we also have $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) \geq 0$ for all ϵ . The last two inequalities imply that we still have $u_{is}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda} + \epsilon e_{is}) = 0$ for all small enough ϵ . Therefore, the result holds under the event mentioned at the beginning of this case.

Case 4 The events $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) > \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is} = 0$ or $d_{it} - u_{i,s-1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{i1}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) = \hat{x}_{st} + \alpha_{ist} - u_{i-1,s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \dots - u_{1s}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda}) - \hat{\lambda}_{is}$ both occur with probability zero, since α_{ist} is independent of d_t and $u_{jr}(\hat{x}_t, d_t, \alpha_t, \hat{\lambda})$ for all $1 \leq j \leq i$ and $1 \leq r \leq s$ with $(j, r) \neq (i, s)$. Since we claim that the result in the lemma holds in w.p.1 sense, we do not need to consider the events in this case, which occur with probability zero.

The events considered above show that the result holds for $u_{is}(\hat{x}_t, d_t, \alpha_t, \cdot)$. Finally, we complete the induction argument by noting that $u_{11}(\hat{x}_t, d_t, \alpha_t, \lambda) = \min\{d_{1t}, [\hat{x}_{1t} + \alpha_{11t} - \lambda_{11}]^+\}$ and $u_{11}(\hat{x}_t, d_t, \alpha_t, \cdot)$ can easily be shown to be differentiable at protection levels $\hat{\lambda}$ w.p.1. \square

B TECHNICAL APPENDIX: THEOREM 6.3.1 IN KUSHNER AND CLARK (1978)

For a function $g(\cdot) : \mathfrak{R}^n \rightarrow \mathfrak{R}$ and a convex set $\mathcal{Z} = \{\lambda : z_i(\lambda) \leq 0 \text{ for all } i = 1, \dots, I\} \subset \mathfrak{R}^n$, we consider solving the problem $\min_{\lambda \in \mathcal{Z}} g(\lambda)$ by using the algorithm

$$\lambda^{k+1} = \operatorname{argmin}_{z \in \mathcal{Z}} \|\lambda^k - \sigma^k s^k - z\|,$$

where $\{\sigma^k\}$ is a sequence of step size parameters, $\{s^k\}$ is a sequence of random step directions and $\operatorname{argmin}_{z \in \mathcal{Z}} \|\cdot - z\|$ projects its argument onto \mathcal{Z} . We assume that the following statements hold.

(B.1) The objective function $g(\cdot)$ is continuously differentiable.

(B.2) We have $\mathbb{E}\{s^k \mid \lambda^1, s^1, \dots, s^{k-1}\} = \nabla g(\lambda^k)$ and $\|s^k - \nabla g(\lambda^k)\| \leq M$ w.p.1 for all $k \geq 1$ for a finite scalar M .

(B.3) The feasible set \mathcal{Z} is closed and bounded, $z_i(\cdot)$ is continuously differentiable for all $i = 1, \dots, I$ and the gradients of the active constraints at any point in \mathcal{Z} are linearly independent.

In this case, the next proposition is a somewhat specialized version of Theorem 6.3.1 in Kushner and Clark (1978). This specialized version is shown in Theorem 3 in van Ryzin and Vulcano (2006).

Proposition 3 *Assume that the sequence of points $\{\lambda^k\}$ is generated by the algorithm above, (B.1)-(B.3) hold and the sequence of step size parameters $\{\sigma^k\}$ satisfies $\sigma^k \geq 0$ for all $k \geq 1$, $\sum_{k=1}^{\infty} \sigma^k = \infty$ and $\sum_{k=1}^{\infty} [\sigma^k]^2 < \infty$. If we use Θ to denote the set of Kuhn-Tucker points for the problem $\min_{\lambda \in \mathcal{Z}} g(\lambda)$ and Θ is connected, then we have $\lambda^k \rightarrow \Theta$ in probability as $k \rightarrow \infty$.*

C TECHNICAL APPENDIX: PROPOSITION 4

The next proposition shows that (S.1) in the proof of Proposition 1 holds.

Proposition 4 *We have $|\partial_{is}^{\Lambda} C_1(x_1, d, \alpha, \lambda)| \leq B_C^{\Lambda}$ w.p.1 for all $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$ for a finite scalar B_C^{Λ} .*

Proof All statements in the proof are in w.p.1 sense. We begin by showing that $|\partial_s^X C_t(x_t, d, \alpha, \lambda)| \leq B_C^X$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar B_C^X . By (13), we have

$$\begin{aligned} |\partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda)| &\leq 1 + |\partial_r^X u_{i,s-1}(x_t, d_t, \alpha_t, \lambda)| + \dots + |\partial_r^X u_{i1}(x_t, d_t, \alpha_t, \lambda)| \\ &\quad + |\partial_r^X u_{i-1,s}(x_t, d_t, \alpha_t, \lambda)| + \dots + |\partial_r^X u_{1s}(x_t, d_t, \alpha_t, \lambda)|. \end{aligned} \quad (25)$$

We define the scalars $\{K_{is} : i \in \mathcal{N}, s \in \mathcal{S}\}$ recursively by

$$K_{is} = 1 + K_{i,s-1} + \dots + K_{i1} + K_{i-1,s} + \dots + K_{1s} \quad (26)$$

with $K_{11} = 1$. In this case, it is easy to show that $|\partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda)| \leq K_{is}$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $r, s \in \mathcal{S}$. To see this, we note that $|\partial_r^X u_{11}(x_t, d_t, \alpha_t, \lambda)| \leq 1 = K_{11}$ by (13). If we assume that $|\partial_r^X u_{jp}(x_t, d_t, \alpha_t, \lambda)| \leq K_{jp}$ for all $1 \leq j \leq i$, $1 \leq p \leq s$ with $(j, p) \neq (i, s)$,

then (25) implies that $|\partial_r^X u_{is}(x_t, d_t, \alpha_t, \lambda)| \leq K_{is}$ and the result follows. Therefore, by (11), we also have $|\partial_r^X R_s(x_t, d_t, \alpha_t, \lambda)| \leq 1 + NK_{NS}$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $r, s \in \mathcal{S}$. In this case, letting $H = \max_{i \in \mathcal{N}, s \in \mathcal{S}} |h_{is}|$, (9) implies that

$$|\partial_r^X C_t(x_t, d, \alpha, \lambda)| \leq NSHK_{NS} + [1 + NK_{NS}] \sum_{s \in \mathcal{S}} |\partial_s^X C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda)|.$$

Using the inequality above and moving backwards over the planning horizon, it is easy to see that $|\partial_r^X C_t(x_t, d, \alpha, \lambda)| \leq NSHK_{NS} [1 + S [1 + NK_{NS}] + \dots + S^{T-t} [1 + NK_{NS}]^{T-t}]$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $r \in \mathcal{S}$, $t \in \mathcal{T}$. Thus, it suffices to let $B_C^X = NSHK_{NS} [1 + S [1 + NK_{NS}] + \dots + S^{T-1} [1 + NK_{NS}]^{T-1}]$.

Proceeding as in (25) but using (12) instead of (13), it is possible to show that $|\partial_{jr}^\Lambda u_{is}(x_t, d_t, \alpha_t, \lambda)| \leq K_{is}$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $i, j \in \mathcal{N}$, $r, s \in \mathcal{S}$, in which case (10) implies that $|\partial_{jr}^\Lambda R_s(x_t, d_t, \alpha_t, \lambda)| \leq NK_{NS}$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $j \in \mathcal{N}$, $r, s \in \mathcal{S}$. Therefore, by (8), we have

$$|\partial_{is}^\Lambda C_t(x_t, d, \alpha, \lambda)| \leq NSHK_{NS} + |\partial_{is}^\Lambda C_{t+1}(R(x_t, d_t, \alpha_t, \lambda), d, \alpha, \lambda)| + SB_C^X NK_{NS}.$$

Using the inequality above and moving backwards over the planning horizon, it is easy to see that $|\partial_{is}^\Lambda C_t(x_t, d, \alpha, \lambda)| \leq [T - t + 1] [NSHK_{NS} + SB_C^X NK_{NS}]$ for all $x_t \in \mathfrak{R}^S$, $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$ and the final result follows by letting $B_C^\Lambda = T [NSHK_{NS} + SB_C^X NK_{NS}]$. \square

D TECHNICAL APPENDIX: PROPOSITION 5

The next proposition shows that (S.2) in the proof of Proposition 1 holds.

Proposition 5 *We have $\mathbb{E}\{\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda)\} = \partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}$ for all $\lambda \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$.*

Proof We begin by noting that $C_1(x_1, d, \alpha, \cdot)$ is Lipschitz in the sense that there exists a finite scalar L_C that satisfies $|C_1(x_1, d, \alpha, \lambda) - C_1(x_1, d, \alpha, \kappa)| \leq L_C \|\lambda - \kappa\|$ w.p.1 for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$. To see this, Proposition 6.3.3 in Glasserman (1994) shows that $\min\{\cdot, \cdot\} : \mathfrak{R}^2 \rightarrow \mathfrak{R}$ and $[\cdot]^+ : \mathfrak{R} \rightarrow \mathfrak{R}$ are Lipschitz. Therefore, $u_{11}(\cdot, d_t, \alpha_t, \cdot)$ is Lipschitz when viewed as a function of the remaining capacities and protection levels. Since composition of Lipschitz functions is Lipschitz by Proposition 6.3.3 in Glasserman (1994), we can use induction over the priority levels and the days of the booking horizon to show that the decision function in (2) is Lipschitz when viewed as a function of the remaining capacities and protection levels. In this case, moving backwards over the planning horizon and again using the fact that composition of Lipschitz functions is Lipschitz, it is easy to see that the cost function in (7) is Lipschitz when viewed as a function of the remaining capacities and protection levels.

Therefore, letting e_{is} be the $N \times S$ dimensional unit vector with a 1 in the element corresponding to i and s , we have $|C_1(x_1, d, \alpha, \lambda + \epsilon e_{is}) - C_1(x_1, d, \alpha, \lambda)|/\epsilon \leq L_C$ for all $\epsilon \in \mathfrak{R}$ w.p.1. This implies that

$$\begin{aligned} \mathbb{E}\{\partial_{is}^\Lambda C_1(x_1, d, \alpha, \lambda)\} &= \mathbb{E}\left\{\lim_{\epsilon \rightarrow 0} [C_1(x_1, d, \alpha, \lambda + \epsilon e_{is}) - C_1(x_1, d, \alpha, \lambda)]/\epsilon\right\} \\ &= \lim_{\epsilon \rightarrow 0} \mathbb{E}\left\{[C_1(x_1, d, \alpha, \lambda + \epsilon e_{is}) - C_1(x_1, d, \alpha, \lambda)]/\epsilon\right\} = \partial_{is}^\Lambda \mathbb{E}\{C_1(x_1, d, \alpha, \lambda)\}, \end{aligned}$$

where the second equality follows from the dominated convergence theorem. We note that the existence of the first limit above follows from the fact that the cost function is differentiable with respect to the protection levels w.p.1. \square

E TECHNICAL APPENDIX: PROPOSITION 6

In this section, we show that (S.3) in the proof of Proposition 1 holds. We begin by defining some new notation. We let x_t^λ be the remaining capacities over the booking horizon on day t when we use the policy characterized by protection levels λ . In other words, the random variables $\{x_t^\lambda : t \in \mathcal{T}\}$ are given recursively by $x_{t+1}^\lambda = R(x_t^\lambda, d_t, \alpha_t, \lambda)$ with $x_1^\lambda = x_1$. We are now ready to show that (S.3) holds.

Proposition 6 *We have $\mathbb{E}\{|\partial_{is}^\Lambda C_t(x_t^\lambda, d, \alpha, \lambda) - \partial_{is}^\Lambda C_t(x_t^\kappa, d, \alpha, \kappa)|\} \leq L_C^\Lambda \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_C^Λ .*

Proof All statements in the proof are in w.p.1 sense. For notational brevity, we define the random variables $\partial_{jr}^\Lambda C_t^\lambda = \partial_{jr}^\Lambda C_t(x_t^\lambda, d, \alpha, \lambda)$, $\partial_{jr}^\Lambda u_{is}^\lambda = \partial_{jr}^\Lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda)$, $\partial_s^X C_{t+1}^\lambda = \partial_s^X C_{t+1}(x_{t+1}^\lambda, d, \alpha, \lambda)$ and $\partial_{jr}^\Lambda R_s^\lambda = \partial_{jr}^\Lambda R_s(x_t^\lambda, d_t, \alpha_t, \lambda)$. Since we have $\partial_{jr}^\Lambda C_{t+1}^\lambda = \partial_{jr}^\Lambda C_{t+1}(R(x_t^\lambda, d_t, \alpha_t, \lambda), d, \alpha, \lambda)$ and $\partial_s^X C_{t+1}^\lambda = \partial_s^X C_{t+1}(R(x_t^\lambda, d_t, \alpha_t, \lambda), d, \alpha, \lambda)$, we can write (8) as

$$\partial_{jr}^\Lambda C_t^\lambda = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \partial_{jr}^\Lambda u_{is}^\lambda + \partial_{jr}^\Lambda C_{t+1}^\lambda + \sum_{s \in \mathcal{S}} \partial_s^X C_{t+1}^\lambda \partial_{jr}^\Lambda R_s^\lambda. \quad (27)$$

The proof of Proposition 4 shows that $|\partial_s^X C_{t+1}^\lambda| \leq B_C^X$ and $|\partial_{jr}^\Lambda R_s^\lambda| \leq NK_{NS}$ for all $\lambda \in \mathfrak{R}^{N \times S}$, $s \in \mathcal{S}$. Therefore, we have

$$\begin{aligned} |\partial_s^X C_{t+1}^\lambda \partial_{jr}^\Lambda R_s^\lambda - \partial_s^X C_{t+1}^\kappa \partial_{jr}^\Lambda R_s^\kappa| &\leq |\partial_s^X C_{t+1}^\lambda| |\partial_{jr}^\Lambda R_s^\lambda - \partial_{jr}^\Lambda R_s^\kappa| + |\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa| |\partial_{jr}^\Lambda R_s^\kappa| \\ &\leq B_C^X |\partial_{jr}^\Lambda R_s^\lambda - \partial_{jr}^\Lambda R_s^\kappa| + NK_{NS} |\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa|, \end{aligned}$$

in which case (27) implies that

$$\begin{aligned} \mathbb{E}\{|\partial_{jr}^\Lambda C_t^\lambda - \partial_{jr}^\Lambda C_t^\kappa|\} &\leq \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \mathbb{E}\{|\partial_{jr}^\Lambda u_{is}^\lambda - \partial_{jr}^\Lambda u_{is}^\kappa|\} + \mathbb{E}\{|\partial_{jr}^\Lambda C_{t+1}^\lambda - \partial_{jr}^\Lambda C_{t+1}^\kappa|\} \\ &\quad + \sum_{s \in \mathcal{S}} B_C^X \mathbb{E}\{|\partial_{jr}^\Lambda R_s^\lambda - \partial_{jr}^\Lambda R_s^\kappa|\} + \sum_{s \in \mathcal{S}} NK_{NS} \mathbb{E}\{|\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa|\}. \quad (28) \end{aligned}$$

Lemmas 11, 12 and 15 in Appendix F respectively show that $\mathbb{E}\{|\partial_{jr}^\Lambda u_{is}^\lambda - \partial_{jr}^\Lambda u_{is}^\kappa|\} \leq L_u^\Lambda \|\lambda - \kappa\|$, $\mathbb{E}\{|\partial_{jr}^\Lambda R_s^\lambda - \partial_{jr}^\Lambda R_s^\kappa|\} \leq L_R^\Lambda \|\lambda - \kappa\|$ and $\mathbb{E}\{|\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa|\} \leq L_C^X \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$ for finite scalars L_u^Λ , L_R^Λ and L_C^X . Therefore, by (28), we have

$$\begin{aligned} \mathbb{E}\{|\partial_{jr}^\Lambda C_t^\lambda - \partial_{jr}^\Lambda C_t^\kappa|\} &\leq NSHL_u^\Lambda \|\lambda - \kappa\| + \mathbb{E}\{|\partial_{jr}^\Lambda C_{t+1}^\lambda - \partial_{jr}^\Lambda C_{t+1}^\kappa|\} \\ &\quad + SB_C^X L_R^\Lambda \|\lambda - \kappa\| + SNK_{NS} L_C^X \|\lambda - \kappa\|, \end{aligned}$$

where H is as in the proof of Proposition 4. Using the inequality above and moving backwards over the planning horizon, it is easy to see that $\mathbb{E}\{|\partial_{jr}^\Lambda C_t^\lambda - \partial_{jr}^\Lambda C_t^\kappa|\} \leq [T - t + 1][NSHL_u^\Lambda + SB_C^X L_R^\Lambda + SNK_{NS} L_C^X] \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $t \in \mathcal{T}$. In this case, the final result follows by letting $L_C^\Lambda = T[NSHL_u^\Lambda + SB_C^X L_R^\Lambda + SNK_{NS} L_C^X]$. \square

F TECHNICAL APPENDIX: LEMMAS 11, 12 AND 15

Lemmas 7-10 are useful when we prove Lemmas 11 and 12.

Lemma 7 *If $\|x_t - z_t\| \leq L \|\lambda - \kappa\|$, then we have $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| \leq M(1+L) \|\lambda - \kappa\|$ w.p.1 for a finite scalar M .*

Proof All statements in the proof are in w.p.1 sense. We define the random variables

$$\begin{aligned} D_{ist}^{x_t, \lambda} &= d_{it} - u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t, d_t, \alpha_t, \lambda) \\ X_{ist}^{x_t, \lambda} &= x_{st} + \alpha_{ist} - u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t, d_t, \alpha_t, \lambda) - \lambda_{is}. \end{aligned}$$

We note that since the decision function never violates the job availabilities, we always have $D_{ist}^{x_t, \lambda} \geq 0$. By (2), we have $u_{is}(x_t, d_t, \alpha_t, \lambda) = \min\{D_{ist}^{x_t, \lambda}, [X_{ist}^{x_t, \lambda}]^+\}$. We consider four cases.

Case 1 Assume that $D_{ist}^{x_t, \lambda} \leq X_{ist}^{x_t, \lambda}$ and $D_{ist}^{z_t, \kappa} \leq X_{ist}^{z_t, \kappa}$. We have $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| = |D_{ist}^{x_t, \lambda} - D_{ist}^{z_t, \kappa}|$.

Case 2 Assume that $D_{ist}^{x_t, \lambda} \leq X_{ist}^{x_t, \lambda}$ and $D_{ist}^{z_t, \kappa} > X_{ist}^{z_t, \kappa}$. We have $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| = |D_{ist}^{x_t, \lambda} - [X_{ist}^{z_t, \kappa}]^+|$. We consider two (sub)cases.

Case 2.a If $D_{ist}^{x_t, \lambda} \leq X_{ist}^{z_t, \kappa}$, then we have $|D_{ist}^{x_t, \lambda} - [X_{ist}^{z_t, \kappa}]^+| = |D_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}| < |D_{ist}^{x_t, \lambda} - D_{ist}^{z_t, \kappa}|$, where the equality follows from the fact that $X_{ist}^{z_t, \kappa} \geq D_{ist}^{x_t, \lambda} \geq 0$.

Case 2.b If $D_{ist}^{x_t, \lambda} > X_{ist}^{z_t, \kappa}$, then we have $|D_{ist}^{x_t, \lambda} - [X_{ist}^{z_t, \kappa}]^+| \leq |D_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}| \leq |X_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}|$, where the first inequality follows from the fact that $D_{ist}^{x_t, \lambda} \geq 0$ and $D_{ist}^{x_t, \lambda} > X_{ist}^{z_t, \kappa}$ so that $D_{ist}^{x_t, \lambda} \geq [X_{ist}^{z_t, \kappa}]^+ \geq X_{ist}^{z_t, \kappa}$. Combining Cases 2.a and 2.b, we obtain $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| \leq |D_{ist}^{x_t, \lambda} - D_{ist}^{z_t, \kappa}| + |X_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}|$ for Case 2.

Case 3 Assume that $D_{ist}^{x_t, \lambda} > X_{ist}^{x_t, \lambda}$ and $D_{ist}^{z_t, \kappa} \leq X_{ist}^{z_t, \kappa}$. The same argument in Case 2 shows that $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| \leq |D_{ist}^{x_t, \lambda} - D_{ist}^{z_t, \kappa}| + |X_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}|$.

Case 4 Assume that $D_{ist}^{x_t, \lambda} > X_{ist}^{x_t, \lambda}$ and $D_{ist}^{z_t, \kappa} > X_{ist}^{z_t, \kappa}$. We have $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| = |[X_{ist}^{x_t, \lambda}]^+ - [X_{ist}^{z_t, \kappa}]^+| \leq |X_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}|$, where we use the fact that $|[a]^+ - [b]^+| \leq |a - b|$ for all $a, b \in \mathfrak{R}$.

Therefore, the four cases above imply that

$$\begin{aligned} |u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| &\leq |D_{ist}^{x_t, \lambda} - D_{ist}^{z_t, \kappa}| + |X_{ist}^{x_t, \lambda} - X_{ist}^{z_t, \kappa}| \\ &\leq (1+L) \|\lambda - \kappa\| \\ &\quad + |u_{i,s-1}(x_t, d_t, \alpha_t, \lambda) - u_{i,s-1}(z_t, d_t, \alpha_t, \kappa)| + \dots + |u_{i1}(x_t, d_t, \alpha_t, \lambda) - u_{i1}(z_t, d_t, \alpha_t, \kappa)| \\ &\quad + |u_{i-1,s}(x_t, d_t, \alpha_t, \lambda) - u_{i-1,s}(z_t, d_t, \alpha_t, \kappa)| + \dots + |u_{1s}(x_t, d_t, \alpha_t, \lambda) - u_{1s}(z_t, d_t, \alpha_t, \kappa)|, \end{aligned} \quad (29)$$

where the second inequality follows from the definitions of the random variables $D_{ist}^{x_t, \lambda}$ and $X_{ist}^{x_t, \lambda}$, and the fact that $\|x_t - z_t\| \leq L \|\lambda - \kappa\|$. In this case, letting K_{is} be as in (26), we can show by induction over the priority levels and the days of the booking horizon that $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| \leq K_{is}(1+L) \|\lambda - \kappa\|$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. In particular, if we assume that the result holds for all $1 \leq j \leq i$ and $1 \leq r \leq s$ with $(j, r) \neq (i, s)$, then (29) implies that $|u_{is}(x_t, d_t, \alpha_t, \lambda) - u_{is}(z_t, d_t, \alpha_t, \kappa)| \leq (1+L)[1 + K_{i,s-1} + \dots + K_{i1} + K_{i-1,s} + \dots + K_{1s}] \|\lambda - \kappa\| = (1+L) K_{is} \|\lambda - \kappa\|$. We can complete the induction argument by noting that $|u_{11}(x_t, d_t, \alpha_t, \lambda) - u_{11}(z_t, d_t, \alpha_t, \kappa)| = |\min\{d_{1t}, [x_{1t} - \lambda_{11}]^+\} - \min\{d_{1t}, [z_{1t} - \kappa_{11}]^+\}| \leq |[x_{1t} - \lambda_{11}]^+ - [z_{1t} - \kappa_{11}]^+| \leq |[x_{1t} - \lambda_{11}] - [z_{1t} - \kappa_{11}]| \leq \|x_t - z_t\| + \|\lambda - \kappa\| \leq (1+L) \|\lambda - \kappa\|$. In this case, the final result follows by letting $M = K_{NS}$. \square

Table 8: List of cases considered in the proofs of Lemmas 10 and 11.

1	$D_{ist}^\lambda \leq X_{ist}^\lambda$	$D_{ist}^\kappa \leq X_{ist}^\kappa$	4	$D_{ist}^\lambda > X_{ist}^\lambda \geq 0$	$D_{ist}^\kappa \leq X_{ist}^\kappa$	7	$D_{ist}^\lambda \geq 0 > X_{ist}^\lambda$	$D_{ist}^\kappa \leq X_{ist}^\kappa$
2	$D_{ist}^\lambda \leq X_{ist}^\lambda$	$D_{ist}^\kappa > X_{ist}^\kappa \geq 0$	5	$D_{ist}^\lambda > X_{ist}^\lambda \geq 0$	$D_{ist}^\kappa > X_{ist}^\kappa \geq 0$	8	$D_{ist}^\lambda \geq 0 > X_{ist}^\lambda$	$D_{ist}^\kappa > X_{ist}^\kappa \geq 0$
3	$D_{ist}^\lambda \leq X_{ist}^\lambda$	$D_{ist}^\kappa \geq 0 > X_{ist}^\kappa$	6	$D_{ist}^\lambda > X_{ist}^\lambda \geq 0$	$D_{ist}^\kappa \geq 0 > X_{ist}^\kappa$	9	$D_{ist}^\lambda \geq 0 > X_{ist}^\lambda$	$D_{ist}^\kappa \geq 0 > X_{ist}^\kappa$

Lemma 8 We have $\|x_t^\lambda - x_t^\kappa\| \leq L_x \|\lambda - \kappa\|$ w.p.1 for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $t \in \mathcal{T}$ for a finite scalar L_x .

Proof All statements in the proof are in w.p.1 sense. Since $x_{t+1}^\lambda = R(x_t^\lambda, d_t, \alpha_t, \lambda)$, we have

$$\|x_{t+1}^\lambda - x_{t+1}^\kappa\| \leq \|x_t^\lambda - x_t^\kappa\| + \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} |u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \quad (30)$$

by (3) and (4). Using the inequality above and moving backwards over the planning horizon, we can show that $\|x_t^\lambda - x_t^\kappa\| \leq 2^{t-1} [(NSM) + \dots + (NSM)^t] \|\lambda - \kappa\|$ for all $t \in \mathcal{T}$, where M is as in Lemma 7. In particular, if we assume that the result holds for day t , then Lemma 7 and (30) imply that

$$\begin{aligned} \|x_{t+1}^\lambda - x_{t+1}^\kappa\| &\leq 2^{t-1} [(NSM) + \dots + (NSM)^t] \|\lambda - \kappa\| \\ &\quad + \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} M \left[1 + 2^{t-1} [(NSM) + \dots + (NSM)^t] \right] \|\lambda - \kappa\| \\ &\leq 2^t [(NSM) + \dots + (NSM)^{t+1}] \|\lambda - \kappa\|. \end{aligned}$$

We can complete the induction argument by noting that $\|x_1^\lambda - x_1^\kappa\| = \|x_1 - x_1\| = 0$. In this case, the final result follows by letting $L_x = 2^{T-1} [(NSM) + \dots + (NSM)^T]$. \square

Lemma 9 We have $|u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq L_u \|\lambda - \kappa\|$ w.p.1 for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_u .

Proof We have $\|x_t^\lambda - x_t^\kappa\| \leq L_x \|\lambda - \kappa\|$ w.p.1 by Lemma 8, in which case Lemma 7 implies that $|u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq M(1 + L_x) \|\lambda - \kappa\|$ w.p.1 and the result follows by letting $L_u = M(1 + L_x)$. \square

For notational brevity, we define the random variables

$$\begin{aligned} D_{ist}^\lambda &= d_{it} - u_{i,s-1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \dots - u_{i1}(x_t^\lambda, d_t, \alpha_t, \lambda) \\ X_{ist}^\lambda &= x_{st}^\lambda + \alpha_{ist} - u_{i-1,s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \dots - u_{1s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \lambda_{is}. \end{aligned}$$

Similar to $D_{ist}^{x_t, \lambda}$ in the proof of Lemma 7, since the decision function never violates the job availabilities, we always have $D_{ist}^\lambda \geq 0$. Table 8 lists nine events concerning the random variables D_{ist}^λ , X_{ist}^λ , D_{ist}^κ and X_{ist}^κ . We note that the union of these events is the whole sample space. The next lemma provides a bound on the probabilities of these events.

Lemma 10 Except for Events 1, 5 and 9, the probabilities of the events in Table 8 are bounded by $L_P \|\lambda - \kappa\|$ for a finite scalar L_P .

Proof We show the result only for Event 2. The other events can be handled by using similar arguments. By Lemmas 8 and 9, we have $\|x_t^\lambda - x_t^\kappa\| \leq L_x \|\lambda - \kappa\|$ and $|u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq L_u \|\lambda - \kappa\|$ w.p.1 for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. Therefore, the definitions of D_{ist}^λ and X_{ist}^λ imply that

$$\begin{aligned} & |(D_{ist}^\kappa - X_{ist}^\kappa + \alpha_{ist}) - (D_{ist}^\lambda - X_{ist}^\lambda + \alpha_{ist})| \\ & \leq |D_{ist}^\lambda - D_{ist}^\kappa| + |X_{ist}^\lambda - X_{ist}^\kappa| \leq SL_u \|\lambda - \kappa\| + [L_x + NL_u + 1] \|\lambda - \kappa\| \end{aligned} \quad (31)$$

w.p.1. In this case, we have

$$\begin{aligned} & \mathbb{P}\{D_{ist}^\lambda \leq X_{ist}^\lambda \text{ and } D_{ist}^\kappa > X_{ist}^\kappa \geq 0\} \leq \mathbb{P}\{D_{ist}^\lambda \leq X_{ist}^\lambda \text{ and } D_{ist}^\kappa > X_{ist}^\kappa\} \\ & = \mathbb{P}\{D_{ist}^\lambda - X_{ist}^\lambda + \alpha_{ist} \leq \alpha_{ist} < D_{ist}^\kappa - X_{ist}^\kappa + \alpha_{ist}\} \leq [1 + L_x + (N + S)L_u] \|\lambda - \kappa\|/\varepsilon, \end{aligned}$$

where we use (31) and the fact that α_{ist} is uniformly distributed over $[0, \varepsilon]$ and it is independent of $D_{ist}^\lambda - X_{ist}^\lambda + \alpha_{ist}$ and $D_{ist}^\kappa - X_{ist}^\kappa + \alpha_{ist}$. The result follows by letting $L_P = [1 + L_x + (N + S)L_u]/\varepsilon$. \square

Lemma 11 *We have $\mathbb{E}\{|\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq L_u^\lambda \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i, j \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_u^λ .*

Proof By (12), we have

$$\begin{aligned} & |\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq |\partial_{jr}^\lambda u_{i,s-1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{i,s-1}(x_t^\kappa, d_t, \alpha_t, \kappa)| \\ & \quad + \dots + |\partial_{jr}^\lambda u_{i1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{i1}(x_t^\kappa, d_t, \alpha_t, \kappa)| \end{aligned} \quad (32)$$

under Event 1 in Table 8, whereas we have

$$\begin{aligned} & |\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq |\partial_{jr}^\lambda u_{i-1,s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{i-1,s}(x_t^\kappa, d_t, \alpha_t, \kappa)| \\ & \quad + \dots + |\partial_{jr}^\lambda u_{1s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{1s}(x_t^\kappa, d_t, \alpha_t, \kappa)| \end{aligned} \quad (33)$$

under Event 5 in Table 8. Finally, we have $|\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| = 0$ under Event 9 in Table 8. The proof of Proposition 4 shows that $|\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda)| \leq K_{NS}$ and we obtain the trivial bound $|\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq 2K_{NS}$ under the other events.

We consider the positive random variables A , B_0 , B_1 , B_2 and B_3 and the disjoint events E_0 , E_1 , E_2 and E_3 such that we have $A \leq B_k$ under the event E_k and $\cup_{k=0}^3 E_k$ is the whole sample space. We have $\mathbb{E}\{A | E_k\} \leq \mathbb{E}\{B_k | E_k\}$ by definition, in which case $\mathbb{E}\{A\} \leq \sum_{k=0}^3 \mathbb{E}\{B_k | E_k\} \mathbb{P}\{E_k\}$. Since B_0 , B_1 , B_2 and B_3 are positive, we also have $\mathbb{E}\{B_k | E_k\} \mathbb{P}\{E_k\} \leq \sum_{l=0}^3 \mathbb{E}\{B_l | E_l\} \mathbb{P}\{E_l\} = \mathbb{E}\{B_k\}$. The last two inequalities imply that $\mathbb{E}\{A\} \leq \mathbb{E}\{B_0 | E_0\} \mathbb{P}\{E_0\} + \mathbb{E}\{B_1\} + \mathbb{E}\{B_2\} + \mathbb{E}\{B_3\}$. Consequently, associating the events E_1 , E_2 and E_3 respectively with Events 1, 5 and 9 in Table 8 and the event E_0 with the union of the remaining six events, and noting (32) and (33), we have

$$\begin{aligned} & \mathbb{E}\{|\partial_{jr}^\lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq 2K_{NS} [6L_P \|\lambda - \kappa\|] \\ & \quad + \mathbb{E}\{|\partial_{jr}^\lambda u_{i,s-1}(x_t^\lambda, d_t, \alpha_t, \lambda) - u_{i,s-1}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \\ & \quad \quad + \dots + \mathbb{E}\{|\partial_{jr}^\lambda u_{i1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{i1}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \\ & \quad \quad \quad + \mathbb{E}\{|\partial_{jr}^\lambda u_{i-1,s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{i-1,s}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \\ & \quad \quad \quad \quad + \dots + \mathbb{E}\{|\partial_{jr}^\lambda u_{1s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\kappa u_{1s}(x_t^\kappa, d_t, \alpha_t, \kappa)|\}, \end{aligned} \quad (34)$$

where the first term on the right side of the inequality above uses the bound on the probabilities of Events 2-4 and 6-8 given in Lemma 10.

In this case, we can show by induction over the priority levels and the days of the booking horizon that $\mathbb{E}\{|\partial_{jr}^\Lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\Lambda u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq K_{is} 12 K_{NS} L_P \|\lambda - \kappa\|$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. In particular, if we assume that the result holds for all $1 \leq j' \leq i$ and $1 \leq r' \leq s$ with $(j', r') \neq (i, s)$, then (34) implies that

$$\begin{aligned} & \mathbb{E}\{|\partial_{jr}^\Lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\Lambda u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \\ & \leq [1 + K_{i,s-1} + \dots + K_{i1} + K_{i-1,s} + \dots + K_{1s}] 12 K_{NS} L_P \|\lambda - \kappa\| = K_{is} 12 K_{NS} L_P \|\lambda - \kappa\|. \end{aligned}$$

It is possible to complete the induction argument by showing in a straightforward fashion that $\mathbb{E}\{|\partial_{jr}^\Lambda u_{11}(x_t^\lambda, d, \alpha, \lambda) - \partial_{jr}^\Lambda u_{11}(x_t^\kappa, d, \alpha, \kappa)|\} \leq K_{11} 12 K_{NS} L_P \|\lambda - \kappa\|$. In this case, the final result follows by letting $L_u^\Lambda = 12 [K_{NS}]^2 L_P$. \square

Lemma 12 *We have $\mathbb{E}\{|\partial_{jr}^\Lambda R_s(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\Lambda R_s(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq L_R^\Lambda \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $j \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_R^Λ .*

Proof By (10) and Lemma 11, we have

$$\begin{aligned} & \mathbb{E}\{|\partial_{jr}^\Lambda R_s(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\Lambda R_s(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \\ & \leq \sum_{i \in \mathcal{N}} \mathbb{E}\{|\partial_{jr}^\Lambda u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_{jr}^\Lambda u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq N L_u^\Lambda \|\lambda - \kappa\| \end{aligned}$$

and the result follows by letting $L_R^\Lambda = N L_u^\Lambda$. \square

The next two lemmas are useful when we prove Lemma 15.

Lemma 13 *We have $\mathbb{E}\{|\partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq L_u^X \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $i \in \mathcal{N}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_u^X .*

Proof The proof follows from the same argument in the proof of Lemma 11. By (13), we have

$$\begin{aligned} |\partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| & \leq |\partial_r^X u_{i,s-1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{i,s-1}(x_t^\kappa, d_t, \alpha_t, \kappa)| \\ & \quad + \dots + |\partial_r^X u_{i1}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{i1}(x_t^\kappa, d_t, \alpha_t, \kappa)| \end{aligned}$$

under Event 1 in Table 8, whereas we have

$$\begin{aligned} |\partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| & \leq |\partial_r^X u_{i-1,s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{i-1,s}(x_t^\kappa, d_t, \alpha_t, \kappa)| \\ & \quad + \dots + |\partial_r^X u_{1s}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{1s}(x_t^\kappa, d_t, \alpha_t, \kappa)| \end{aligned}$$

under Event 5 in Table 8. The proof of Proposition 4 shows that $|\partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda)| \leq K_{NS}$ and we obtain the trivial bound $|\partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X u_{is}(x_t^\kappa, d_t, \alpha_t, \kappa)| \leq 2 K_{NS}$ under the other events in Table 8. The final result follows by the same induction argument in the proof of Lemma 11. \square

Lemma 14 We have $\mathbb{E}\{|\partial_r^X R_s(x_t^\lambda, d_t, \alpha_t, \lambda) - \partial_r^X R_s(x_t^\kappa, d_t, \alpha_t, \kappa)|\} \leq L_R^X \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $r, s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_R^X .

Proof Following the same argument in the proof of Lemma 12, but using (11) and Lemma 13 instead of (10) and Lemma 11, it is easy to see that the result holds by letting $L_R^X = NL_u^X$. \square

Lemma 15 We have $\mathbb{E}\{|\partial_s^X C_t(x_t^\lambda, d, \alpha, \lambda) - \partial_s^X C_t(x_t^\kappa, d, \alpha, \kappa)|\} \leq L_C^X \|\lambda - \kappa\|$ for all $\lambda, \kappa \in \mathfrak{R}^{N \times S}$, $s \in \mathcal{S}$, $t \in \mathcal{T}$ for a finite scalar L_C^X .

Proof The proof follows from an argument similar to the proof of Proposition 6. For notational brevity, we define the random variables $\partial_r^X C_t^\lambda = \partial_r^X C_t(x_t^\lambda, d, \alpha, \lambda)$, $\partial_r^X u_{is}^\lambda = \partial_r^X u_{is}(x_t^\lambda, d_t, \alpha_t, \lambda)$ and $\partial_r^X R_s^\lambda = \partial_r^X R_s(x_t^\lambda, d_t, \alpha_t, \lambda)$. In this case, we can write (9) as

$$\partial_r^X C_t^\lambda = \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \partial_r^X u_{is}^\lambda + \sum_{s \in \mathcal{S}} \partial_s^X C_{t+1}^\lambda \partial_r^X R_s^\lambda. \quad (35)$$

The proof of Proposition 4 shows that $|\partial_r^X C_{t+1}^\lambda| \leq B_C^X$ and $|\partial_r^X R_s^\lambda| \leq 1 + NK_{NS}$ for all $\lambda \in \mathfrak{R}^{N \times S}$, $s \in \mathcal{S}$. Therefore, we can use (35) and proceed as in the proof of Proposition 6 to obtain

$$\begin{aligned} \mathbb{E}\{|\partial_r^X C_t^\lambda - \partial_r^X C_t^\kappa|\} &\leq \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}} h_{is} \mathbb{E}\{|\partial_r^X u_{is}^\lambda - \partial_r^X u_{is}^\kappa|\} + \sum_{s \in \mathcal{S}} B_C^X \mathbb{E}\{|\partial_r^X R_s^\lambda - \partial_r^X R_s^\kappa|\} \\ &\quad + \sum_{s \in \mathcal{S}} (1 + NK_{NS}) \mathbb{E}\{|\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa|\}. \end{aligned} \quad (36)$$

Lemmas 13 and 14 show that $\mathbb{E}\{|\partial_r^X u_{is}^\lambda - \partial_r^X u_{is}^\kappa|\} \leq L_u^X \|\lambda - \kappa\|$ and $E\{|\partial_r^X R_s^\lambda - \partial_r^X R_s^\kappa|\} \leq L_R^X \|\lambda - \kappa\|$ for all $i \in \mathcal{N}$, $s \in \mathcal{S}$. Therefore, by (36), we have

$$\begin{aligned} \mathbb{E}\{|\partial_r^X C_t^\lambda - \partial_r^X C_t^\kappa|\} &\leq NSHL_u^X \|\lambda - \kappa\| + SB_C^X L_R^X \|\lambda - \kappa\| \\ &\quad + \sum_{s \in \mathcal{S}} (1 + NK_{NS}) \mathbb{E}\{|\partial_s^X C_{t+1}^\lambda - \partial_s^X C_{t+1}^\kappa|\}, \end{aligned} \quad (37)$$

where H is as in the proof of Proposition 4. Letting $\Psi = NSHL_u^X + SB_C^X L_R^X$ and $\Gamma = 1 + NK_{NS}$, and moving backwards over the planning horizon, we can use (37) to show that

$$\mathbb{E}\{|\partial_r^X C_t^\lambda - \partial_r^X C_t^\kappa|\} \leq [\Psi + \Psi S \Gamma + \dots + \Psi S^{T-t} \Gamma^{T-t}] \|\lambda - \kappa\|$$

for all $r \in \mathcal{S}$, $t \in \mathcal{T}$. In particular, if we assume that the result holds for day $t+1$, then (37) implies that $\mathbb{E}\{|\partial_r^X C_t^\lambda - \partial_r^X C_t^\kappa|\} \leq \Psi \|\lambda - \kappa\| + S \Gamma [\Psi + \Psi S \Gamma + \dots + \Psi S^{T-t-1} \Gamma^{T-t-1}] \|\lambda - \kappa\|$ and the result holds for day t . We complete the induction argument by noting that $\mathbb{E}\{|\partial_r^X C_T^\lambda - \partial_r^X C_T^\kappa|\} \leq \Psi \|\lambda - \kappa\|$ by (37). In this case, the final result follows by letting $L_C^X = \Psi + \Psi S \Gamma + \dots + \Psi S^{T-1} \Gamma^{T-1}$. \square