An Analysis of the Control-Algorithm Re-solving Issue in Inventory and Revenue Management

Nicola Secomandi

Tepper School of Business, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890, USA, ns7@andrew.cmu.edu

Tepper Working Paper 2004-06
Forthcoming in Manufacturing & Service Operations Management

While inventory- and revenue-management problems can be represented as Markov decision process (MDP) models, in some cases the well-known dynamic-programming curse-of-dimensionality makes it computationally prohibitive to solve them exactly. An alternative solution approach, called here the control-algorithm approach, is to employ a math program (MP) to approximately represent the MDP and use its optimal solution to heuristically instantiate the parameters of the decision rules of a given class of control policies. As new information is observed over time, the control algorithm can incorporate it by re-solving the MP and revising the parameters of the decision rules with the newly obtained solution. The re-solving issue arises when one reflects on the consequences of this revision: Does the performance of the control algorithm really improve by revising its decision-rule instantiation with the solution of the re-solved MP, or should rather an appropriate modification of the prior solution be used? This paper analyzes the control-algorithm re-solving issue for a class of finite-horizon inventory- and revenue-management problems. It establishes sufficient, but not necessary, conditions under which re-solving does not deteriorate the performance of a control algorithm, and it applies these results to control algorithms for network revenue-management and multiproduct make-to-order production with lost sales and positive leadtime.

1. Introduction

Mathematical representation of inventory- and revenue-management problems typically gives rise to Markov decision process (MDP) models (e.g., Zipkin 2000, Talluri and van Ryzin 2004). In some cases, as in network revenue-management and inventory management with lost sales and positive leadtime, exact solution of these models is impractical, due to the well-known dynamic programming curse-of-dimensionality phenomenon. Therefore, operations
management (OM) researchers and practitioners have developed approximate methods to solve them. One such approach relies on representing the MDP as a math program (MP) and using its optimal solution to instantiate, in a heuristic fashion, the decision rules of a given class of control policies. In this paper, a pair (MP, control-policy class) is called a control algorithm and this solution approach is referred to as the control-algorithm method for heuristically solving MDPs.

The re-solving issue. With the control-algorithm approach, one is intuitively tempted to periodically re-solve the MP in order to incorporate in the control policy the effects of modified environmental conditions, e.g., a swift inventory reduction due to the realization of a significant amount of demand. However, as discussed by Cooper (2002, §5) and pointed out by Talluri and van Ryzin (2004, p. 94) in the context of network revenue-management, the effect of this re-solving on the performance of the control algorithm is yet not well understood in theory. In particular, Cooper provides an example where re-solving can be detrimental, which illustrates the key aspect behind the re-solving issue. With the control-algorithm approach, one would like to re-solve the MP when new information becomes available over time. However, given that this is an approximate model of the MDP, re-solving it does not necessarily improve the performance of the control algorithm.

As Cooper (2002, p. 725) correctly states, in revenue management “there are significant practical benefits to [re-solving], such as the incorporation of user input, the adjustment of estimates of model parameters, and the use of overbooking techniques to deal with cancellations and no-shows.” But from a theoretical point of view, the question of “when and how to update heuristic policies” remains unanswered (Cooper 2002, p. 726). Shedding light on the control-algorithm re-solving issue is not only an academic pursuit. OM practitioners should be aware of the re-solving behavior of control algorithms employed in applications.

Focus, objective, contributions, and relevance. The paper’s focus is the control-algorithm approach to heuristically solve the MDP formulation of a class of finite-horizon inventory/revenue-management problems. As pointed out by one of the referees, it is important to highlight that this paper deals with model approximation, rather than data approximation issues, which may arise due to the inability of accurately forecasting data further in the planning horizon. In this paper, all the relevant parameters do not change over time. Instead, the additional information being dynamically gathered relates to the unfolding of uncertain demand events, i.e., the realization of stochastic demands whose probability distributions are known.
The paper’s objective is to develop a theory of control-algorithm re-solving for the stated class of problems that would enable an OM researcher to analyze the re-solving behavior of a particular control algorithm. This is a worthwhile endeavor for the following reasons. First, the OM literature seems to lack both a formalization of the control-algorithm approach to solving MDPs and a theory of re-solving for the class of problems considered in this paper. Second, this literature does not offer an explanation of the odd re-solving behavior displayed by the control algorithm studied by Cooper (2002) in his revenue-management example, it is not clear whether this behavior depends on the type of policy or the MP used, and it is not known whether re-solving affects other types of control algorithms, in particular the more commonly used bid-price-based control algorithms.

This paper makes the following contributions. Section 2 formalizes the control-algorithm approach for a class of finite-horizon inventory/revenue-management problems. Section 3 develops a simple theory of control-algorithm re-solving for this class of problems. In particular, this section establishes sufficient, but not necessary, conditions under which re-solving does not worsen the performance of a control algorithm. The main result is that if the control algorithm satisfies a certain sequential-improvement property, then its performance cannot worsen as a result of re-solving. When the control algorithm uses an MP that provides an accurate representation of the control policy and satisfies a certain sequential-consistency property related to sequential feasibility of opportunely updated MP solutions, the control algorithm also satisfies the sequential-improvement property, and re-solving cannot be detrimental. When the MP is an inaccurate representation of the control policy, it has a time-dependent formulation that satisfies a technical condition, and the control algorithm is sequentially consistent, it is possible to establish a weaker structural result on the benefit of re-solving. Throughout, sequential consistency emerges as an important property to obtain positive structural re-solving results for control algorithms.

Section 4 applies this theory to problems of network revenue-management (§4.1) and multiproduct make-to-order production with lost sales and positive leadtime (§4.2). In §4.1, this paper explains precisely the odd behavior of the control algorithm studied by Cooper (2002) by bringing to light the fact that this control algorithm can fail to be sequentially consistent, it illustrates that re-solving cannot negatively affect the policy studied by Cooper when its decision-rule parameters are instantiated with the solution of a different MP that explicitly models demand uncertainty, and it points out that bid-price-based control algorithms can suffer from re-solving. In §4.2, it establishes structural re-solving results for the case of
zero holding cost and uncapacitated resource-inventory. Section 5 concludes by highlighting further research avenues.

This paper is relevant to the OM research community because the control-algorithm approach seems to be widespread both in the OM academic literature and applications. This is evidently the case in network revenue-management, but the use of heuristics derived from sequential reoptimizations of an MP is common in other OM areas, such as production planning (Bitran and Yanasse 1984, Shapiro 1993, Chand et al. 2002) and logistics (Powell 2003).

Additional related literature. The control-algorithm re-solving theory developed in this paper is related to the theory of rollout policies/algorithms developed by Bertsekas and Tsitsiklis (1996) and Bertsekas, Tsitsiklis and Wu (1997), and summarized by Bertsekas (2000, Chapter 6). The basic idea behind rollout policies is to sequentially generate heuristic state-dependent actions for a given MDP by using the decision rules of a given heuristic base policy, which is easy to compute and evaluate, to obtain an estimate of the optimal value function associated with the available actions (see Bertsekas and Castanon 1999, Secomandi 2001, 2003, and Bertsimas and Popescu 2003 for stochastic scheduling/routing and network revenue-management applications). The theory of rollout policies is concerned with establishing conditions under which this scheme generates a sequence of state-dependent actions with improved expected objective-function value than that achievable by simply applying the base heuristic policy, whose decision-rule parameters are known to begin with.

This theory provides a natural starting point for studying the control-algorithm re-solving issue. However, it needs to be adapted to deal with the case where the parameters of the decision rules are not known in advance and, consequently, one employs an MP approximation of an MDP formulation to instantiate them. In other words, the rollout-policy theory deals with the properties of the state-dependent actions that are sequentially generated by using a given heuristic policy for the exact MDP model as “guide.” In contrast, the control-algorithm re-solving theory developed in this paper deals with the properties of the control policy obtained by sequentially optimizing, in the space of its decision-rule parameters, an MP approximation of the MDP formulation. This basic difference requires modifying and extending the theory of rollout policies.

This paper is also related to the model-predictive-control approach developed in the control literature (see, e.g., Garcia et al. 1989, Mayne et al. 2000, Camacho and Bordons 2004). This approach generates heuristic closed-loop controls (state-dependent actions) for
a given control problem by sequentially optimizing an open-loop controller (typically an MP) that approximates the true control problem (MDP) formulation over a given time horizon. The “first” open-loop control, i.e., the action corresponding to the immediate time, is implemented, the remaining ones are discarded, and the process is repeated. A fundamental difference sets apart the model-predictive-control and the control-algorithm approaches.

The optimization model used in the model-predictive-control approach optimizes open-loop controls, only one of which is ever taken. Instead, the control-algorithm MP optimizes in the space of the parameters of the decision rules of a given class of state-dependent policies, and an MP solution can be used to instantiate such a policy to control the system in a state-dependent fashion over the entire planning horizon. Hence, re-solving is not typically perceived as an issue in the model-predictive-control literature because it is necessary to obtain closed-loop control of the system. (However, whether sequentially reoptimizing an open-loop controller is better than using the open-loop controller itself has been studied by Bertsekas 2000, pp. 288-290.) In stark contrast with the model-predictive-control approach, with a control algorithm re-solving is not necessary to obtain a state-dependent control policy, which is obtained with a single optimization at the beginning of the planning horizon. The issue here is the merit of sequentially revising the parameters of its decision rules.

2. Modeling Setting and Problem Statement

A company operates with a finite planning-horizon consisting of a set of time periods indexed by set $\mathcal{T} = \{1, \ldots, n\}$, with $n \geq 1$ a finite integer. At the beginning of period $t \in \mathcal{T}$, the company owns a nonnegative amount $z_{it} \in \mathcal{Z}_i$ of resource $i \in I$. Set $\mathcal{Z}_i \subseteq \mathbb{R}^+$ includes the allowed levels of resource $i$ (it does not depend on $t$). The company incurs a cost $h_{it}(z_{it})$ for holding an amount $z_{it}$ of resource $i$ during period $t \in \mathcal{T}$. At the beginning of period $t \in \mathcal{T} \setminus \{n\}$, the company orders a quantity $q_{it}(z_{it}) \in \mathbb{R}^+$ of resource $i \in I$ at cost $g_{it}(q_{it}(z_{it}))$. The delivery leadtime is equal to one period and is identical for all the resources, i.e., the array of orders $q_t(z_t) = (q_{it}(z_{it}), i \in I)$ placed at time $t$ is delivered at the beginning of the next period, time $t+1$. (When a subscript is dropped from a quantity, the resulting quantity denotes the array of quantities whose index is being dropped.)

During period $t \in \mathcal{T} \setminus \{n\}$, the company employs the available resources to satisfy a nonnegative random demand vector $D_t$ for products in set $\mathcal{J}$. Satisfying an amount
\[ u_{jt}(z_t, D_t) \in [0, D_{jt}] \] \text{ of product } j \in \mathcal{J} \text{ demand yields a revenue equal to } f_{jt}(u_{jt}(z_t, D_t)) \text{ and requires allocating a nonnegative amount } a_{ij} \text{ of resource } i \text{ per unit of product-}j \text{ demand-satisfied (for simplicity, the resource-allocation cost is zero). The company makes demand-management choices with full knowledge of realized demand, hence the notation } u_{jt}(z_t, D_t), \text{ and can reject part of the observed demand, which in this case is lost. By assumption the demand random variables are mutually stochastically independent within each period and across different periods, and are not affected by how the company satisfies demand. The company’s objective is to maximize the expected profit collected during the planning horizon.}

This joint inventory- and demand-management problem resembles that studied by Van Mieghem and Rudi (2002, §4.1). In fact, it is partially a simplification of theirs because it ignores the capacity investment problem and each product demand can only be satisfied by a single processing activity. However, it is fundamentally more difficult to solve optimally because it features a positive leadtime and lost sales, a situation that Van Mieghem and Rudi (2002, p. 332) describe as “a deadly combination.”

**Finite-horizon MDP formulation.** This problem can be formulated as a finite-horizon MDP (see, e.g., Puterman 1994 for a treatment of MDP theory and applications). Set \( T \) indexes the stages and vector \( z_t \) is the state in stage \( t \in T \). The state set is \( Z = \times_{i \in I} Z_i \), i.e., it includes the set of all possible vectors of allowed resource-levels. The optimal value function in state \( z_t \) is denoted \( v_t(z_t) \). Define \( E_t[\cdot] := E[\cdot \mid Z_t = z_t] \). The Bellman equation at time \( t \in T \setminus \{n\} \) in state \( z_t \in Z \) is

\[
\begin{align*}
v_t(z_t) &= \max_{q_t \geq 0} E_t[\phi_t(z_t, q_t, D_t)] - \left[ \sum_{i \in I} g_{it}(q_{it}) + \sum_{i \in I} h_{it}(z_{it}) \right] \\
\phi_t(z_t, q_t, d_t) &:= \max_{u_{t+1}} \sum_{j \in \mathcal{J}} f_{jt}(u_{jt}) + v_{t+1}(z_{t+1}) \quad \text{s.t. } z_{it+1} = z_{it} + q_{it} - \sum_{j \in \mathcal{J}} a_{ij} u_{jt}, \forall i \in I \\
&\quad \sum_{j \in \mathcal{J}} a_{ij} u_{jt} \leq z_{it}, \forall i \in I \\
&\quad 0 \leq u_{jt} \leq d_{jt}, \forall j \in \mathcal{J} \\
&\quad z_{it+1} \in Z_i, \forall i \in I.
\end{align*}
\]

The boundary conditions are \( v_t(0) := 0, \forall t \in T \), and \( v_n(z_n) := -\sum_{i \in I} h_{in}(z_{in}), \forall z_n \in Z \).

**Control algorithms.** Solving this MDP formulation becomes quickly intractable as the cardinality of set \( \mathcal{J} \) increases, because of the well-known dynamic-programming curse-of-
dimensionality. The control-algorithm method is an alternative heuristic solution-approach. Denote by $\Pi$ the set of all admissible policies for this MDP. Consider a subset $\Pi$ of set $\Pi$ and denote $\delta_\pi^t(z_t, D_t)$ the time-$t$ decision-rule vector of policy $\pi \in \Pi$, i.e., $\pi = \{\delta_\pi^t(\cdot), \ t \in T\backslash\{n\}\}$. Note that $\delta_\pi^t(z_t, D_t) = (q_\pi^t(z_t), u_\pi^t(z_t, D_t))$, where $q_\pi^t(z_t)$ and $u_\pi^t(z_t, D_t)$ are policy-$\pi$ time-$t$ decision-rule vectors that compute, respectively, inventory and demand-management action-vectors. These decision rules depend on some parameters that need to be instantiated for the decision rules to be usable (see §4 for examples). A control algorithm for this MDP is the pair $(\text{MP}, \Pi)$: it uses the optimal solution of an MP representation of this MDP to instantiate the parameters of the decision rules of policy $\pi \in \Pi$.

**MP formulations.** This paper considers MP formulations that at any time $t \in T$ employ set $S(t) \subseteq T$ to index the remaining time periods in the planning horizon. Two generic MP formulations are analyzed: a time-dependent and a time-aggregate formulation. In the time-dependent case, set $S(t)$ is equal to $\{t, t+1, \ldots, n\}$, e.g., if $n = 3$ then $S(1) = \{1, 2, 3\}$. At time $t \in T \backslash\{n\}$, in state $z_t \in Z$, this generic MP is

$$
\max_{x,y,w} \sum_{s \in S(t)\backslash\{n\}} \left[ \sum_{j \in J} \varphi_{js}(x_{js}) - \sum_{i \in I} g_{is}(y_{is}) - \sum_{i \in I} h_{is}(w_{is}) \right] - \sum_{i \in I} h_{in}(w_{in})
$$

s.t. $w_{it} = z_{it}, \ \forall i \in I$ (2)

$$
 w_{is+1} = w_{is} + y_{is} - \sum_{j \in J} a_{ij} x_{js}, \ \forall i \in I, \ s \in S(t) \backslash \{n\}
$$

(3)

$$
\sum_{j \in J} a_{ij} x_{js} \leq w_{is}, \ \forall i \in I, \ s \in S(t) \backslash \{n\}
$$

(4)

$$
0 \leq x_{js} \leq b_{js}, \ \forall j \in J, \ s \in S(t) \backslash \{n\}
$$

(5)

$$
y_{is} \geq 0, \ \forall i \in I, \ s \in S(t) \backslash \{n\}
$$

(6)

$$
w_{is} \in Z, \ \forall i \in I, \ s \in S(t).
$$

(7)

Decision variables $x_{js}$ and $y_{is}$ indicate, respectively, the amounts of product-$j$ demand satisfied in period $s \in S(t) \backslash \{n\}$ and of resource $i$ ordered at the beginning of this period. Variable $w_{is}$ is available resource-$i$ inventory at time $s$. Function $\varphi_{js}(\cdot)$ is a “surrogate” for the revenue of satisfying product-$j$ demand in period $s \in S(t) \backslash \{n\}$. Hence, the objective function (1) to be maximized is a “surrogate” profit over the remaining periods in the planning horizon. Constraint-set (2) initializes $w_{it}$ to $z_{it}$, (3) enforces resource-balance conditions, (4) limits inventory usage, and (5)-(7) place bounds on the decision variables. In some of the MPs presented in §4 upper bound $b_{js}$ is set equal to period-$s$ expected-demand.
In the time-aggregate case, set \( S(t) \) is equal to \( \{t, n\} \), e.g., again if \( n = 3 \) then \( S(1) = \{1, 3\} \). Here, one aggregates the remaining periods in the planning horizon, except the very last one, into a single “long” period starting at time \( t \) and ending at time \( n \). Hence, decision variables \( x_{jt} \) and \( y_{it} \), respectively, represent the amounts of product-\( j \) demand-satisfied before time \( n \) and resource \( i \) ordered at time \( t \) and delivered at time \( n \), i.e., the delivery leadtime is also long. It is also assumed that the quantities \( \varphi_{jt}(\cdot) \) and \( h_{it}(\cdot) \) reflect the time aggregation. With these qualifications, the generic time-dependent and time-aggregate MP formulations are essentially identical, except for the inventory-balance constraints that in the latter case read
\[
 w_{in} = w_{it} + y_{it} - \sum_{j \in J} a_{ij} x_{jt}, \quad \forall i \in I.
\]
The aggregate case is mainly useful when inventory replenishments are not allowed during the planning horizon, so that only the demand-management problem is relevant, e.g., the network revenue-management problem discussed in §4.1. To see this, note that the optimal order size of any resource \( i \in I \) is zero at time \( t \) because additional resources at time \( n \) incur the holding cost but have no productive use. Hence, the quantities related to time-\( t \) ordering could be suppressed, but for uniformity of exposition this is not done below.

When only the demand-management problem is relevant, some of the dual variables are of interest. In the time-dependent formulation, the nonnegativity of variables \( w_{in} \)'s and recursive usage of the inventory-balance constraints (2)-(3) imply the constraints
\[
 \sum_{s=t}^{n-1} \sum_{j \in J} a_{ij} x_{js} \leq z_{it}, \quad \forall i \in I.
\]
In the time-aggregate formulation, these constraints coincide with the inventory-usage constraints (4) with \( w_{it} = z_{it} \), i.e., \( \sum_{j \in J} a_{ij} x_{jt} \leq z_{it}, \quad \forall i \in I \). The dual variables associated with these constraints, denoted \( p_{it} \)'s both in the time-dependent and time-aggregate cases, can be used to instantiate demand-management decision rules based on the shadow-price interpretation of their optimal values, each of which provides an estimate of the value of a marginal unit of resource-inventory \( z_{it} \) (see §4.1).

The sequence of events. Consider a specific MP formulation of either the time-dependent or time-aggregate type (see §4 for examples). At time \( t \in T \setminus \{n\} \), in state \( z_t \in Z \), this MP is solved to optimality. Suppose that it has a unique optimal primal-solution (this assumption is discussed at the end of this section). It is the set of vectors \( \{(x^*_s(t), y^*_s(t)), s \in S(t) \setminus \{n\}\} \cup \{w^*_s(t), s \in S(t)\} \), where suffix \( (t) \) is a time-stamp. The placeholder \( \zeta(t) \) denotes this solution (an asterisk is not used here to indicate optimality to
simplify notation in the later development). When relevant, \(\zeta(t)\) also includes set \(\{p^*_it(t), i \in I\}\). Solution \(\zeta(t)\) is used to instantiate the decision rules in the current state. Once updated as in Definition 1 below, this solution can also be used to instantiate the decision rules in all possible future states, effectively yielding a policy \(\pi(t) \in \Pi\).

**Remark 1 (Dependence of \(\zeta(t)\) and \(\pi(t)\) on \(z_t\)).** Both the quantities \(\zeta(t)\) and \(\pi(t)\) depend on the state \(z_t\), but this dependence is not explicitly expressed in the employed notation to ease the notational burden.

Continuing with the sequence of events, in state \(z_t\) an order is placed according to the action vector yielded by decision-rule vector \(q^*_it(t)(z_t)\), a realization \(d_t\) of \(D_t\) is observed, and this demand is satisfied according to the action vector computed by decision-rule vector \(u^*_it(t)(z_t, d_t)\). The application of these action vectors yields state \(z_{t+1}^{\pi(t)}\), which is a realization of \(Z_{t+1}^{\pi(t)}\), the random variable state-of-the-system at time \(t+1\), reached from state \(z_t\) at time \(t\) by applying the time-\(t\) decision rules of policy \(\pi(t)\). In this new state, the optimal solution \(\zeta(t)\) is updated as in Definition 1.

**Definition 1 (Updated MP solution).** Fix time \(t \in T \setminus \{n\}\) and states \(z_t, z_{t+1}^{\pi(t)} \in \mathcal{Z}\). Consider the solution \(\{(x_s, y_s), s \in \mathcal{S}(t) \setminus \{n\}\} \cup \{w_s, s \in \mathcal{S}(t)\}\). Suppose that this solution is used to instantiate the decision-rule vectors of policy \(\pi(t)\) in state \(z_t\). If \(t = n - 1\), the time-\(n\) update of this solution is \(w'_n := z_{n-1}\). If \(t \neq n - 1\), its time \((t+1)\)-update is defined as in cases 1) and 2) below.

1) Time-dependent MP: \(\mathcal{S}(t) = \{t, t + 1, \ldots, n\} \Rightarrow \mathcal{S}(t + 1) = \{t + 1, t + 2, \ldots, n\}\). The time-\((t+1)\) solution-update is \(x'_s := x_s, y'_s := y_s, \forall s \in \mathcal{S}(t+1) \setminus \{n\}\), \(w'_{it+1} := z_{it+1}^{\pi(t)}\), and \(w'_{is+1} := w'_is + y'_is - \sum_{j \in J} a^Tjs\), \(\forall i \in I, s \in \mathcal{S}(t+1) \setminus \{n\}\).

2) Time-aggregate MP: \(\mathcal{S}(t) = \{t, n\} \Rightarrow \mathcal{S}(t + 1) = \{t + 1, n\}\). The time-\((t+1)\) solution-update is \(x'_{t+1} := x_t - u^{\pi(t)}_t(z_t, d_t), y'_{t+1} := y_t - q^{\pi(t)}_t(z_t), w'_{it+1} := z_{it+1}^{\pi(t)}\), and \(w'_m := w'_{it+1} + y'_{dt+1} - \sum_{j \in J} a^Tjs_{jt+1}\), \(\forall i \in I\).

When the dual solution \(\{p^*_it, i \in I\}\) is relevant, its time-\((t+1)\) update is \(p'_{it+1} := p^*_it, \forall i \in I\).

The rationale behind this definition is fairly self-explanatory. In particular, following application of the decision rules in period \(t\), in case 1) it will typically hold that \(w'_{it+1} \neq z_{it+1}^{\pi(t)}\); in case 2) the new time-\((t+1)\) variables must reflect the inventory- and demand-management choices made in period \(t\). Note that, when relevant, the dual-variable vector
\( p_t^\pi(t) \) remains unchanged. After updating solution \( \zeta(t) \), policy \( \pi(t) \) can be used to make decisions in period \( t + 1 \). Alternatively, the MP is reformulated and re-solved to obtain \( \zeta(t + 1) \) and, consequently, new decision-rule parameters, i.e., the new policy \( \pi(t + 1) \in \Pi \). The process is then repeated.

For simplicity, but with one exception in §4.1, this paper assumes that in the re-solving case MP is solved at all times \( t \in T \setminus \{n\} \). Also, note that, when re-solving is not employed, the solution \( \zeta(t) \) updated at time \( t + 1 \) in state \( z_{t+1}^{\pi(t)} \) could be reused in state \( z_{\tau}^{\pi(t)} \) at time \( \tau = t + 2, \ldots, n - 1 \), provided that one keeps updating it according to Definition 1. To facilitate bookkeeping, the quantity \( \zeta'(t, \ell) \), with \( t, \ell \in T \) and \( t < \ell \), denotes an optimal MP solution obtained at time \( t \) in state \( z_t \) and subsequently updated in all states \( z_{\tau}^{\pi(t)} \) at times \( \tau = t + 1, t + 2, \ldots, \ell \). Notice that the superscript in \( \zeta'(t, \ell) \) indicates that this is an updated solution, the index \( t \) the time when this solution was originally obtained, and the index \( \ell \) the time of its last update.

**The re-solving issue.** The control-algorithm re-solving issue arises because one may not be better off, in expectation as of time 1, by replacing \( \pi(t) \) with \( \pi(t + 1) \) at each time \( t + 1 \in T \), i.e., by using the optimal MP solution obtained at time \( t + 1 \), rather than the update of that obtained at time \( t \), to instantiate the time-\((t + 1)\) decision rules. More formally, define the random profit of using the action vectors computed by decision-rule vectors \( q_t^{\pi(t)}(z_t) \) and \( u_t^{\pi(t)}(z_t, D_t) \) in period \( t \in T \setminus \{n\} \) and state \( z_t \in Z \) as

\[
\gamma_t^{\pi(t)}(z_t, D_t) := \sum_{j \in J} f_{jt}(u_t^{\pi(t)}(z_t, D_t)) - \sum_{i \in I} g_{it}(q_t^{\pi(t)}(z_t)) - \sum_{i \in I} h_{it}(z_it).
\]

The random profit-to-come to the end of period \( t \in T \) from the initial state \( z_1 \) for control algorithm (MP, \( \Pi \)) when re-solving at all times \( \tau \in T \setminus \{n\}, \tau \leq t \) is

\[
V_t^{(MP, \Pi)}(z_1) := \sum_{\tau \in T \setminus \{n\}} \gamma_\tau^{\pi(\tau)}(Z_\tau^{\pi(\tau-1)}, D_\tau) - 1_{\{t=n\}} \sum_{i \in I} h_{in}(Z_{in}^{\pi(n-1)}),
\]

where \( Z_1^{\pi(0)} := z_1 \) and \( 1_{\{t=n\}} \) is the indicator function of event \( \{t = n\} \). The expected profit-to-go to the end of period \( n \) from state \( z_t \in Z \) at time \( t \in T \setminus \{n\} \) under any policy \( \pi \in \Pi \) is

\[
v_t^\pi(z_t) := E_t \left[ \sum_{\tau \in T \setminus \{n\}, \tau \geq t} \gamma_\tau^{\pi}(Z_\tau^{\pi}, D_\tau) - \sum_{i \in I} h_{in}(Z_{in}^{\pi}) \right],
\]

where \( Z_t^{\pi} \equiv z_t \). (Note that with this notation \( v_t(z_t) = \max_{\pi \in \Pi} v_t^\pi(z_t) \).) If \( t = n \), define \( v_n^\pi(z_n) := v_n(z_n) := -\sum_{i \in I} h_{in}(z_{in}), \forall z_n \in Z \); in particular, this holds for \( \pi(n) \), which is
otherwise undefined at time \( n \). Analyzing the re-solving issue for control algorithm (\( MP, \Pi \)) entails establishing conditions under which the following inequality holds:

\[
v_{t}^{\pi(1)}(z_{1}) \leq E_{1}\left[V_{n}^{(MP,\Pi)}(z_{1})\right].
\]

**Uniqueness of optimal MP primal-solution.** The assumption that MP has a unique optimal primal-solution that can be efficiently computed in all possible states simplifies the exposition but can be otherwise relaxed. The theory presented in §3 only relies on the weaker assumption that an algorithm to find a feasible MP primal-solution is used that, at a given re-solving time, always evaluates the update of the solution that was employed to instantiate the parameters of the decision rules at the previous re-solving time.

### 3. Re-solving Theory

The theory of re-solving is built with reference to sequentially improving (SI) and sequentially consistent (SC) control algorithms. These concepts extend those introduced by Bertsekas, Tsitsiklis and Wu (1997) and Secomandi (2003) by adapting them to the setting of this paper. The analysis is also based on the novel concept of control algorithm with \( \Pi \)-accurate MP. Except for Remark 3, the presentation is self-contained and does not require previous familiarity with the theory of rollout policies.

To ease the exposition, in this section the following notation is used with respect to generic quantity \( G \):

\[
\hat{G}_{\tau}^{d} := G_{\tau}^{\pi(t)}, \ t = 1, \ldots, n, \ \tau = t, \ldots, n \n\]

\[
\hat{G}_{t} := \hat{G}_{t}^{d}, \ t = 1, \ldots, n.
\]

Hence, \( v_{t}^{\pi(t)}(\cdot) \) is replaced by \( \hat{v}_{t}(\cdot) \), \( v_{t+1}^{\pi(t)}(\cdot) \) by \( \hat{v}_{t+1}^{t}(\cdot) \), \( Z_{t+1}^{\pi(t)} \) and \( \hat{Z}_{t+1}^{t} \) respectively, and \( \gamma_{t}^{\pi(t)}(\cdot) \) by \( \hat{\gamma}_{t}(\cdot) \). In addition, superscript \( (MP, \Pi) \) is suppressed.

Definition 2 defines an SI control algorithm as one for which switching from policy \( \pi(t) \), obtained at time \( t \) in state \( z_{t} \), to policy \( \pi(t + 1) \), obtained at time \( t + 1 \) in state \( \hat{z}_{t+1}^{t} \), is always beneficial. Lemma 1 records the fact that for such a control algorithm this switching is beneficial in expectation as of time \( t \) in state \( z_{t} \). Proposition 1 establishes that re-solving is beneficial for an SI control algorithm. Hence, sequential-improvement is a desirable property of a control algorithm.
Definition 2 (SI control algorithm). Control algorithm \((MP, \bar{\Pi})\) is SI if for every time \(t \in T \setminus \{n\}\) and every state \(z_t \in \mathcal{Z}\), the policy \(\pi(t)\) obtained in state \(z_t\) at time \(t\) satisfies \(\hat{v}_{t+1}^l(\hat{z}_{t+1}^l) \leq \hat{v}_{t+1}(\hat{z}_{t+1}^l)\) for all \(\hat{z}_{t+1}^l \in \mathcal{Z}\).

Lemma 1 (Characterization of SI control algorithm). If control algorithm \((MP, \bar{\Pi})\) is SI then \(\hat{v}_t(z_t) \leq E_t \left[ \hat{\gamma}_t(z_t, D_t) + \hat{v}_{t+1}(\hat{Z}_{t+1}^l) \right], \forall t \in T \setminus \{n\}, z_t \in \mathcal{Z}\).

**Proof.** From Definition 2, for all \(t \in T \setminus \{n\}\) and \(z_t \in \mathcal{Z}\), it holds that
\[
\hat{v}_t(z_t) = E_t \left[ \hat{\gamma}_t(z_t, D_t) + \hat{v}_{t+1}(\hat{Z}_{t+1}^l) \right] \leq E_t \left[ \hat{\gamma}_t(z_t, D_t) + \hat{v}_{t+1}(\hat{Z}_{t+1}^l) \right].
\]

Proposition 1 (Re-solving with SI control algorithm). If control algorithm \((MP, \bar{\Pi})\) is SI, the following inequalities hold:
\[
\hat{v}_1(z_1) \leq E_1 \left[ V_1(z_1) + \hat{v}_2(\hat{Z}_2^l) \right]
\]
\[
E_1 \left[ V_t(z_1) + \hat{v}_{t+1}(\hat{Z}_{t+1}^l) \right] \leq E_1 \left[ V_{t+1}(z_1) + \hat{v}_{t+2}(\hat{Z}_{t+2}^l) \right], t = 1, \ldots, n - 2.
\]

Therefore it also holds that \(\hat{v}_1(z_1) \leq E_1 \left[ V_n(z_1) \right]\).

**Proof.** The first inequality holds by Lemma 1 because \(\hat{\gamma}_1(z_1, D_1) \equiv V_1(z_1)\). For \(t = 1, \ldots, n - 2\), Lemma 1 implies that
\[
E_1 \left[ V_t(z_1) + \hat{v}_{t+1}(\hat{Z}_{t+1}^l) \right] \leq E_1 \left[ V_t(z_1) + E_{t+1} \left[ \hat{\gamma}_{t+1}(\hat{z}_{t+1}^l, D_{t+1}) + \hat{v}_{t+2}(\hat{Z}_{t+2}^l) \right] \right] = E_1 \left[ V_{t+1}(z_1) + \hat{v}_{t+2}(\hat{Z}_{t+2}^l) \right].
\]
Since then \(\hat{v}_1(z_1) \leq E_1 \left[ V_{n-1}(z_1) + \hat{v}_n(\hat{Z}_{n-1}^l) \right]\) and \(V_{n-1}(z_1) + \hat{v}_n(\hat{Z}_{n-1}^l) \equiv V_n(z_1)\), the last claimed inequality follows. \(\square\)

Proposition 2 establishes that sequential consistency of a control algorithm (Definition 3) and \(\bar{\Pi}\)-accurateness of its MP (Definition 4) are sufficient conditions for the control algorithm to be SI. An SC control algorithm is one for which the time-\((t + 1)\) update of the MP optimal solution obtained at time \(t\) in state \(z_t\) remains MP-feasible in any state \(\hat{z}_{t+1}^l\). A control algorithm with \(\bar{\Pi}\)-accurate MP is one whose MP objective function provides an accurate representation of the expected profit-to-go of the policy in set \(\bar{\Pi}\) associated with the MP optimal solution obtained at time \(t\) in state \(z_t\), and subsequently updated in state \(\hat{z}_{t+1}^l\) at the next re-solving time \(t + 1\).
In the following, at any time $t \in T \setminus \{n\}$ and in any state $z_t \in Z$, consider an MP solution $\zeta$, and define $\psi_s(z_t, \zeta)$ and $\nu_t(z_t, \zeta)$, respectively, the period-$s$ and total profits of this solution:

$$\psi_s(z_t, \zeta) := \sum_{j \in J} \varphi_{js}(x_{js}) - \sum_{i \in I} g_{is}(y_{is}) - \sum_{i \in I} h_{is}(w_{is}), \forall s \in S(t) \setminus \{n\}$$

$$\psi_n(z_t, \zeta) := -\sum_{i \in I} h_{in}(w_{in})$$

and

$$\nu_t(z_t, \zeta) := \sum_{s \in S(t)} \psi_s(z_t, \zeta).$$

Therefore, the optimal MP objective function value in state $z_t$, denoted $\nu^*_t(z_t)$, is equal to $\nu_t(z_t, \zeta(t)), \forall t \in T \setminus \{n\}$ and $z_t \in Z$. Since the MP introduced in §2 is not defined at time $n$, given updated solution $\zeta'(t, n)$, with $t \in T \setminus \{n\}$, for the purposes of the ensuing analysis it is useful to define $\nu_n(z_n, \zeta'(t, n)) := \nu^*_n(z_n) := v_n(z_n), \forall z_n \in Z$.

**Definition 3 (SC control algorithm).** Control algorithm $(MP, \bar{\Pi})$ is SC if for every time $t \in T \setminus \{n\}$ and every state $z_t \in Z$, it holds that $\zeta'(t, t+1)$ is feasible for MP for all $\hat{z}_{t+1}^t \in Z$.

**Definition 4 (\(\bar{\Pi}\)-accurate MP).** Control algorithm $(MP, \bar{\Pi})$ has a $\bar{\Pi}$-accurate MP if for every time $t \in T \setminus \{n\}$ and every state $z_t \in Z$, it holds that $\check{\nu}_t(z_t) = \nu^*_t(z_t)$ and $\check{\nu}_{t+1}(\hat{z}_{t+1}^t) = \nu_{t+1}(\hat{z}_{t+1}^t, \zeta'(t, t+1))$ for all $\hat{z}_{t+1}^t \in Z$.

**Proposition 2 (SC control algorithm with $\bar{\Pi}$-accurate MP).** If control algorithm $(MP, \bar{\Pi})$ is SC and has a $\bar{\Pi}$-accurate MP then it is SI.

**Proof.** The result follows because for any $t \in T \setminus \{n\}$ and $z_t, \hat{z}_{t+1}^t \in Z$ it holds that

$$\hat{\nu}_{t+1}(\hat{z}_{t+1}^t) = \nu_{t+1}(\hat{z}_{t+1}^t, \zeta'(t, t+1)); \bar{\Pi}\text{-accurateness w.r.t. } \pi(t)$$

$$\leq \nu^*_{t+1}(\hat{z}_{t+1}^t); \text{ feasibility of } \zeta'(t, t+1) \text{ and optimality of } \zeta(t+1)$$

$$= \check{\nu}_{t+1}(\hat{z}_{t+1}^t); \bar{\Pi}\text{-accurateness w.r.t. } \pi(t+1).\Box$$

When MP is $\bar{\Pi}$-inaccurate, in some state $z_t$ at time $t \in T \setminus \{n\}$, its optimal objective function value may not be equal to the expected profit-to-go of policy $\pi(t) \in \bar{\Pi}$, i.e., $\nu^*_t(z_t) \neq \hat{\nu}_t(z_t)$, a fact that is captured by stating that the control algorithm embeds model error. Definition 5 introduces two model errors, $\alpha_1$ and $\alpha_n$, for a control algorithm with time-dependent MP.
Definition 5 (Model errors with time-dependent MP). The quantities $\alpha_1$ and $\alpha_n$ are the model errors of a control algorithm with time-dependent MP at times 1 and $n$:

\[
\begin{align*}
\alpha_1 & := \max \left\{ \hat{v}_1^1(z_1) - \nu_1^*(z_1), 0 \right\} \\
\alpha_n & := \max \left\{ A_n - E_1 [V_n(z_1)], 0 \right\} \\
A_n & := \psi_1(z_1, \zeta(1)) + E_1 \left[ \psi_2(\hat{Z}_2^1, \zeta'(1,2)) + E_2 \left[ \ldots + E_{n-1} \left[ \psi_n \left( \hat{Z}_{n-1}^1, \zeta'(n-1,n) \right) \right] \right] \right].
\end{align*}
\]

The first error, $\alpha_1$, measures by how much the true expected profit of the control policy obtained by solving at times 1 falls short, if at all, of the expected profit-to-go of policy $\pi(1)$. The second error, $\alpha_n$, quantifies by how much the true expected profit of the control policy obtained by solving at all times $t \in T \setminus \{n\}$ falls short, if at all, of the expected value of this profit as “seen” by MP, i.e., $A_n$. For clarity, note that for $n = 2$ and $n = 3$ the terms $A_2$ and $A_3$ are

\[
A_2 = \psi_1(z_1, \zeta(1)) + E_1 \left[ \psi_2 \left( \hat{Z}_2^1, \zeta'(1,2) \right) \right] \\
A_3 = \psi_1(z_1, \zeta(1)) + E_1 \left[ \psi_2 \left( \hat{Z}_2^1, \zeta'(2) \right) + E_2 \left[ \psi_3 \left( \hat{Z}_3^1, \zeta'(2,3) \right) \right] \right].
\]

When a control algorithm has a $\overline{\Pi}$-inaccurate MP, an exact result such as Proposition 1 is not available. Proposition 3 establishes an “approximate” version of this result for an SC control algorithm with time-dependent, possibly $\overline{\Pi}$-inaccurate, MP that satisfies inequality (9) in Lemma 2.

Lemma 2 (Characterization of SC control algorithm with time-dependent MP). Suppose that control algorithm (MP, $\overline{\Pi}$) is SC, MP has a time-dependent formulation, and for all $t \in T \setminus \{n\}$ and $z_t \in Z$ it holds that

\[
\sum_{s \in S(t) \setminus \{t\}} \psi_s(z_t, \zeta(t)) \leq E_t \left[ \nu_{t+1} \left( \hat{Z}_{t+1}^t, \zeta'(t, t+1) \right) \right]. \tag{9}
\]

Then it also holds that $\nu_t^* \leq \psi_t(z_t, \zeta(t)) + E_t \left[ \nu_{t+1}^* \left( \hat{Z}_{t+1}^t \right) \right], \forall t \in T \setminus \{n\}, z_t \in Z$.

Proof. Definition 3 implies that

\[
\nu_{t+1} \left( \hat{z}_{t+1}^t, \zeta'(t, t+1) \right) \leq \nu_{t+1}^* \left( \hat{z}_{t+1}^t \right), \forall t \in T \setminus \{n\}, z_t, \hat{z}_{t+1}^t \in Z. \tag{10}
\]

Then, for all $t \in T \setminus \{n\}$ and $z_t \in Z$, it holds that

\[
\begin{align*}
\nu_t^* (z_t) & = \psi_t(z_t, \zeta(t)) + \sum_{s \in S(t) \setminus \{t\}} \psi_s(z_t, \zeta(t)) \\
& \leq \psi_t(z_t, \zeta(t)) + E_t \left[ \nu_{t+1} \left( \hat{Z}_{t+1}^t, \zeta'(t, t+1) \right) \right]; \text{ from (9)} \\
& \leq \psi_t(z_t, \zeta(t)) + E_t \left[ \nu_{t+1}^* \left( \hat{Z}_{t+1}^t \right) \right]; \text{ from (10).} \Box
\end{align*}
\]
Proposition 3 (Re-solving with SC control algorithm and time-dependent MP).
If control algorithm \((MP, \Pi)\) is SC and has a time-dependent \(MP\) that satisfies condition (9), then it holds that \(\bar{v}_1(z_1) \leq E_1[V_n(z_1)] + \alpha_1 + \alpha_n\).

Proof. Repeated applications of Lemma 2 yield

\[
\nu_1^*(z_1) \leq \psi_1(z_1, \zeta(1)) + E_1\left[\nu_2^*\left(\hat{Z}_2^1\right)\right] =: B_2 \\
\leq \psi_1(z_1, \zeta(1)) + E_1\left[\nu_2^*\left(\hat{Z}_2^1, \zeta(2)\right) + E_2\left[\nu_3^*\left(\hat{Z}_3^2\right)\right]\right] =: B_3 \\
\leq \ldots \leq B_n \equiv A_n,
\]

or \(\nu_1^*(z_1) \leq A_n\). The result follows because this inequality and Definition 5 imply that \(\bar{v}_1(z_1) \leq \nu_1^*(z_1) + \alpha_1 \leq A_n + \alpha_1 \leq E_1[V_n(z_1)] + \alpha_1 + \alpha_n\). \(\square\)

Remark 2 (Relevance of Proposition 3). The relevance of Proposition 3 is not its practical usefulness, since, in general, estimating \(\alpha_1\) and \(\alpha_n\) requires evaluating a control policy. Once this is done, the result is of no use. Its relevance is theoretical in that it reinforces the importance of sequential consistency in deriving structural re-solving results for a control algorithm.

Remark 3 (Proposition 3 and a result of Secomandi 2003). Proposition 3 is similar to a result obtained by Secomandi (2003, Appendix A), which, however, deals with data error for rollout policies and algorithms rather than model error for control algorithms.

4. Applications

This section applies the theory developed in §3 to problems of network revenue-management and multiproduct make-to-order production. The first application illustrates a situation where a company manages demand for a given amount of inventory, the second one where a company periodically makes both inventory-management and demand-management decisions. In this section, the pair \((MP, CP)\) denotes a control algorithm that uses a control policy in class \(CP\), and \(v_{tMP}^{CP(t)}(z_t)\) indicates the expected profit-to-go of the policy in class \(CP\) obtained from the optimal solution of \(MP\) at time \(t\) in state \(z_t\). In addition, given the \(MP\)-solution \(\zeta\), the notation \(\delta_t^{CP}(\cdot; \zeta)\) highlights the dependence of this decision-rule vector on \(\zeta\).
4.1 Network Revenue-Management

Consider a company that sells products by taking advance reservations during a booking period for a given future delivery date, when bookings are operationally satisfied. Delivering a product at this date requires usage of multiple resources, which gives rise to a network problem. The resource-inventory amounts are given. The company’s objective is to devise a booking control-policy that maximizes the expected revenue collected during the booking period. (The possibility that the company overbooks its resources to account for cancellations and no-shows is not considered.) This problem can be modeled as a simplified version of the model of §2.

MDP. Since only the demand-management problem is relevant, the following quantities are not defined: \( q_{it}(\cdot), g_{it}(\cdot), \) and \( h_{it}(\cdot), \) \( \forall i \in I \) and \( t \in T. \) As is standard in the revenue-management literature (Talluri and van Ryzin 2004, p. 87), set \( T \) divides the booking period into a collection of time periods during which at most one booking request can occur with positive probability, and, if a booking request occurs in period \( t, \) it entails a positive amount of demand for only one product \( j \in J. \) Therefore, in any time period \( t \in T \setminus \{n\}, \) a realization of random vector \( D_t \) can have at most one positive entry, the one corresponding to the product being requested. Product delivery for the accepted bookings occurs at time \( n. \) The quantity \( r_j \) is the price of product \( j \) and \( f_{jt}(u_{jt}(z_t, D_t)) = r_j u_{jt}(z_t, D_t). \) For brevity, the MDP formulation is not displayed here because it is a special case of that presented in §2. In particular, note the simplified boundary conditions \( v_n(z_n) := 0, \forall z_n \in Z. \)

MPs. The deterministic linear program (DLP) is the most widely employed MP in network revenue-management applications (see Talluri and van Ryzin 2004, pp. 93-95). It is a special case of the time-aggregate MP formulation of §2. For a given \( t \in T \setminus \{n\}, \) let \( S(t) = \{t, n\}, \) \( \varphi_{jt}(x_{jt}) = r_j x_{jt}, \) and \( b_{jt} = E[D_j(t, n - 1)], \) where \( D_j(t, n - 1) \) is the random variable cumulative demand-to-come for product \( j \) during periods \( t, t + 1, \ldots, n - 1, \) i.e., \( D_j(t, n - 1) := \sum_{\tau=t}^{n-1} D_{j\tau}, \forall j \in J. \) Quantity \( y_{it} \) is not defined, \( \forall i \in I. \) Removing decision variable \( w_{it} \) and substituting \( w_{it} \) with \( z_{it}, \forall i \in I, \) and further suppressing subscript \( t \) (except from \( z_{it}, \)) the model is

\[
\begin{align*}
\max & \quad \sum_{j \in J} r_j x_j \\
\text{s.t.} & \quad (4), (5).
\end{align*}
\]

The probabilistic nonlinear program (PNLP) is a time-aggregate simple-recourse stochastic
program that accounts for demand uncertainty by setting \( \varphi_{jt}(x_{jt}) = r_j E[\min\{D_j(t, n - 1), x_{jt}\}] \) and \( b_{jt} = +\infty, \forall j \in J \). Removing subscript \( t \) from the decision variables, the model is
\[
\max \sum_{j \in J} r_j E[\min\{D_j(t, n - 1), x_j\}]
\]
subject to (4), (5).

Ciancimino et al. (1999) employ PNLP in a passenger-railway application and de Boer et al. (2002) analyze computationally the performance of DLP- and PNLP-based control algorithms (see also Talluri and van Ryzin 2004, pp. 95-98).

Although usually not employed in applications, one may also formulate time-dependent versions of DLP and PNLP (denoted TDLP and TPNLP, respectively). In TDLP, \( \varphi_{js}(x_{js}) = r_j x_{js} \) and \( b_{js} = E[D_{js}], \forall j \in J, s \in S(t) \setminus \{n\} \), and the model is
\[
\max \sum_{j \in J} \sum_{s \in S(t) \setminus \{n\}} r_j x_{js}
\]
subject to (5), (8).

In TPNLP, \( \varphi_{js}(x_{js}) = r_j E[\min\{D_{js}, x_{js}\}] \) and \( b_{js} = +\infty, \forall j \in J, s \in S(t) \setminus \{n\} \), and the model is
\[
\max \sum_{j \in J} \sum_{s \in S(t) \setminus \{n\}} r_j E[\min\{D_{js}, x_{js}\}]
\]
subject to (5), (8).

**Remark 4 (Integral demand).** In the following analysis, in particular Proposition 4, if demand is integer-valued, the objective functions of PNLP and TPNLP should be modified by letting \( \varphi_{jt}(x_{jt}) = r_j E[\min\{D_j(t, n - 1), \lfloor x_{jt}\rfloor\}] \) and \( \varphi_{js}(x_{js}) = r_j E[\min\{D_{js}, \lfloor x_{js}\rfloor\}] \).

Alternatively, one could add integrality conditions on decision variables \( x_j \) and \( x_{js} \) in these models. These modifications will be implicitly assumed in the ensuing analysis.

**Decision rules.** Only decision-rule vector \( u_t(\cdot) \) is relevant. Faced with a request for an amount \( d_{jt} > 0 \) of product \( j \) in time period \( t \in T \setminus \{n\} \), the company needs to decide how much demand to accept, i.e., partial acceptance is allowed. The partitioned booking-limit (BL) policy uses the primal solution to the above MPs as product-allocation vector. For example, consider DLP or PNLP. In this case the decision rule is \( u^{BL}_{jt}(z_t, d_t; x^*) = \min\{d_{jt}, x^*_j\}, \forall j \in J, t \in T \setminus \{n\}, z_t \in Z \). With TDLP or TPNLP, simply replace \( x^* \) with \( x^*_t \). The bid price (BP) policy employs the optimal duals \( p^*_i \)'s to value each unit of resource-
inventory requested in time period $t$. In this case the decision rule, $\forall j \in J$, $t \in T \setminus \{n\}$, $z_t \in Z$, is

$$u_{jt}^{BP}(z_t, d_t; p^*) = \begin{cases} 
\min\{d_{jt}, \min\{z_t/a_{ij} : a_{ij} > 0, i \in I\}\} & \text{if } r_j \geq \sum_{i \in I} a_{ij} p_i^* \\
0 & \text{otherwise}.
\end{cases}$$

Re-solving analysis. The previous MPs and control policies generate eight control algorithms, i.e., those obtained by using DLP, PNLP, TDLP, and TPNLP to instantiate BL and BP. Proposition 4 deals with control algorithms (PNLP, BL) and (TPNLP, BL), Proposition 5 with (TDLP, BL). There are no structural results for the remaining five control algorithms and they can all suffer from re-solving. This is shown by revisiting Example 1 in Cooper (2002, p. 726) and pointing out that they can fail to be SC.

Proposition 4 (Re-solving with (PNLP, BL) and (TPNLP, BL)). The following inequalities hold for control algorithms (PNLP, BL) and (TPNLP, BL):

$$v_1^{BL(1),PNLP}(z_1) \leq E \left[ V_n^{(PNLP,BL)}(z_1) \right] \quad \text{and} \quad v_1^{BL(1),TPNLP}(z_1) \leq E \left[ V_n^{(TPNLP,BL)}(z_1) \right].$$

Proof. Consider PNLP. Fix arbitrary time $t \in T \setminus \{n\}$ and state $z_t \in Z$. In this state, policy BL uses $x^*(t)$. At time $t + 1$, the next re-solving time, given a realization $d_t$ of $D_t$, the corresponding updated solution is $x^*(t) - u_t^{BL}(z_t, d_t; x^*(t))$ and the new state is $z_{t+1}^{BL(t)} = z_t - \sum_{j \in J} a_{ij} u_{jt}^{BL}(z_t, d_t; x^*(t))$. The updated solution satisfies constraints (5) because it is nonnegative and $\sum_{j \in J} a_{ij} x_j^*(t) \leq z_{t+1}^{BL(t)} + \sum_{j \in J} a_{ij} u_{jt}^{BL}(z_t, d_t; x^*(t)) = z_t$, which holds for all $i \in I$ by the feasibility of $x^*(t)$ at time $t$ in state $z_t$. Hence, the updated solution is feasible in the new state and (PNLP, BL) is SC. It is easy to show that (TPNLP, BL) is also SC. It is clear that (PNLP, BL) and (TPNLP, BL) are both BL-accurate. Hence, they are both SI by Proposition 2. The claimed inequalities then follow from Proposition 1. $\square$

Proposition 5 (Re-solving with (TDLP, BL)). The following inequality holds for control algorithm (TDLP, BL):

$$v_1^{BL(1),TDLP}(z_1) \leq E \left[ V_n^{(TDLP,BL)}(z_1) \right] + \alpha_n^{(TDLP,BL)}.$$

Proof. Fix arbitrary time $t \in T \setminus \{n\}$ and state $z_t \in Z$. Control algorithm (TDLP, BL) is SC because $u_{jt}^{BL}(z_t, D_t; x_j^*(t)) = \min\{D_{jt}, x_j^*(t)\}$, $\forall j \in J$. Condition (9) holds trivially as an equality because the equality $\psi_s^{(BL(t), x^*(t))} = \sum_{j \in J} r_j x_j^*(t)$, for $s = t + 1, \ldots, n - 1$ and all possible realizations of $Z_{t+1}^{BL(t)}$, implies $\nu_{t+1}^{BL(t), x^*(t)} = \sum_{s=t+1}^{n-1} \sum_{j \in J} r_j x_j^*(t)$, which then implies $\sum_{s=t+1}^{n-1} \psi_s^{(z_t, x^*(t))} = E \left[ \nu_{t+1}^{BL(t), x^*(t)} \right]$ (note that by Definition 1 the update of $x_s^*(t)$ at time $t + 1$ is identical to $x_s^*(t)$ for $s = t + 1, \ldots, n - 1$). A simple
adaptation of Lemma 2 in Cooper (2002, p. 724) shows that $\nu_1^*(z_1) \geq v_{1,TDLP}^{BL(1)}$. The result then follows from Proposition 3 by noting that $\alpha_{1}^{TDLP,BL} = 0$. □

Cooper’s example revisited. Cooper (2002) considers a single resource, with two available units of inventory, and two products. (While the example entails only one resource, it is meant to illustrate the potentially negative re-solving behavior of a control algorithm for network revenue-management.) The price of product 1 is $10, that of product 2 is $2. The demands for the two products are independent homogeneous Poisson processes with unitary rates. The reservation period consists of two aggregate time periods each of unitary length, i.e., each aggregate period includes several smaller time periods during which at most one arrival can occur with positive probability. Furthermore, departing from the simplifying assumption made in §2, re-solving occurs only at the beginning of the second aggregate period. Table 4.1 summarizes the re-solving behavior of the eight control algorithms in this example. For each control algorithm, the second and third columns display the expected revenue without and with re-solving at the beginning of the second aggregate period, respectively, as a percentage of the value of the optimal policy. The fourth column reports the percentage gain (over the solution without re-solving) yielded by re-solving.

As originally pointed out by Cooper, the performance of (DLP, BL) worsens with re-solving. In addition, consistently with Propositions 4 and 5, Table 4.1 shows that re-solving has no effect on (PNLP, BL) and it improves the performance of the BL policies associated with TDLP and TPNLP. (However, the performance of (TDLP, BL) and (TPNLP, BL) is substantially lower than that of (DLP, BL) and (PNLP, BL).) Interestingly, re-solving is detrimental for (DLP, BP), (PNLP, BP), and (TPNLP, BP), and has no effect on (TDLP, BP). But the dual optimal solution of TDLP is not unique. It is easy to verify that there
are optimal inventory-usage-constraint duals for which (TDLP, BP) becomes equivalent to (DLP, BP). Hence, re-solving can also be detrimental for (TDLP, BP).

**Sample-path analysis of re-solving with (DLP, BL) in Cooper’s example.** It is insightful to elucidate the precise reason why re-solving is detrimental for (DLP, BL) in this example. To this aim, denote a sample path by two pairs that show the cumulative accepted demand of each of the two products through each aggregate period, e.g., \{(0, 1), (1, 1)\} indicates that in aggregate period 1 a product-2 booking is accepted and, additionally, in aggregate period 2 a product-1 request is accepted. An analysis of the sample paths for (DLP, BL), without and with re-solving, reveals the reason of the adverse re-solving behavior.

At the beginning of the first aggregate period, DLP is

\[
\begin{align*}
\text{max} & \quad 10x_1 + 2x_2 \\
\text{s.t.} & \quad x_1 + x_2 \leq 2 \\
& \quad 0 \leq x_1, x_2 \leq 2.
\end{align*}
\]

Its optimal solution is \((x_1^*, x_2^*) = (2, 0)\). The BL control policy associated with this solution is to accept up to two product-1 requests and reject all product-2 requests during the entire
booking period. Part a) of Figure 1 displays the possible sample paths of this policy without re-solving, i.e., when DLP is solved only once. If two or more product-1 requests occur in aggregate period 1, then there is no need to re-solve at the beginning of aggregate period 2, since there is no more available inventory. If only one product-1 request occurs in the first aggregate period, then the updated solution is also optimal at the start of the second aggregate period. However, if no product-1 request materializes during aggregate period 1, then one may wish to re-solve DLP at the beginning of aggregate period 2 to update the control policy parameters. In this case the following DLP is solved:

$$\begin{align*}
\text{max} & \quad 10x_1 + 2x_2 \\
\text{s.t.} & \quad x_1 + x_2 \leq 2 \\
& \quad 0 \leq x_1, x_2 \leq 1.
\end{align*}$$

(11)

Part b) of Figure 1 displays the possible sample paths of the updated BL policy when DLP is re-solved at the beginning of aggregate period 2. Re-solving has the effect of substituting sample path 4 with sample paths 5 and 6. This substitution occurs because the update of the original DLP optimal solution is unchanged and therefore is infeasible when DLP is re-solved when no product-1 request is received in aggregate period 1. Note that this is a demand-constraint infeasibility that arises because the right-hand side of demand constraints (11) is equal to 1 but the updated optimal-solution is still $(2, 0)$. Hence, the adverse sample-path substitution is a direct consequence of control algorithm (DLP, BL) failing to be SC.

**Re-solving with BP-based control algorithms.** As previously discussed, the BP-based control algorithms can suffer from re-solving. Consider control algorithms (TPNLP, BP) and (TDLP, BP). TPNLP and TDLP are BP-inaccurate, and one may wish to establish a result similar to Proposition 5. However, this is not possible because, as shown below, these control algorithms are not SC. Consider two successive re-solving times $t$ and $t+1$. Interpret the random vector $D_{BP}^t$ as the accepted demand in period $t$ by policy BP($t$). Suppose that for some resource $i$ it holds that

$$\Pr \left( \sum_{j \in J} a_{ij} D_{jt}^{BP(t)} + \sum_{j \in J} \sum_{s=t+1}^{n-1} a_{ij} x_{js}^t(t) > z_{it} \right) > 0.$$ 

This is plausible since BP($t$) does not limit the number of product-$j$ requests accepted in period $t$ to be less than or equal to $x_{jt}^t(t)$. Then, for some realization of $D_{BP}^t$, the following “violating” condition holds: $\sum_{j \in J} \sum_{s=t+1}^{n-1} a_{ij} x_{js}^t(t) > z_{it} - \sum_{j \in J} a_{ij} D_{jt}^{BP(t)} \equiv z_{it+1}^{BP(t)}$. This
inequality shows that the update of $x_i^*(t)$ at time $t+1$ in state $z_{it+1}^{BP(t)}$ is infeasible because it violates the inventory-usage constraint of resource $i$.

Consider now (PNLP, BP) and (DLP, BP). PNLP and DLP are BP-inaccurate. It is an open question whether a re-solving result similar to Proposition 3 could be established for a control algorithm with time-aggregate and $\Pi$-inaccurate MP, and how such a result would relate to sequential consistency. Nevertheless, it is important to point out that the two control algorithms under consideration are not SC. (DLP, BP) cannot be SC in the same way of (DLP, BL). Moreover, both for (DLP, BP) and (PNLP, BP), suppose that for some product $j$ it holds that $\Pr\left(D_{jt}^{BP(t)} > x_j^*(t)\right) > 0$. Then the violating condition is $x_j^*(t) - d_{jt}^{BP(t)} < 0$, i.e., the update of $x_j^*(t)$ is no longer feasible at time $t+1$ in state $z_{it+1}^{BP(t)}$ because it violates the nonnegativity constraint.

Finally, it is important to point out the following statement by Talluri and van Ryzin (1998, p. 1587): “Williamson’s (1992, Chapter 6) extensive simulation studies showed that with frequent reoptimization, the performance of DLP bid prices is quite good.” (On a related note, Maglaras and Meissner 2006 establish that re-solving makes one of their price-based revenue-management control algorithms achieve asymptotic optimality.)

### 4.2 Multiproduct Make-To-Order Production

Consider a manufacturing company that holds inventories of multiple components and assembles them to produce multiple products in a make-to-order fashion. It operates with a finite planning horizon divided into time periods, e.g., weekly time-intervals for the current quarter. At the beginning of each period, it decides how much component inventory to order (the inventory-replenishment leadtime is one period). After placing the order, but before receiving it, the company observes a realization of a stochastic demand for its products and uses the available component inventories to satisfy it (for simplicity the production leadtime and the assembly cost are both equal to zero). Demand can be rejected and unmet demand is lost. For simplicity of exposition, demand is assumed to be continuous. With integral demand one must modify the formulation of TPNLP1, introduced below, in a manner analogous to what stated in Remark 4 in §4.1 with respect to PNLP and TPNLP. The company’s objective is to manage component-inventory and product-demand to maximize the expected profit during the planning horizon.

**MDP.** Sets $T$, $I$, and $J$ index the time periods, components, and products, respectively.
The state variable $z_{it}$ is the $i$-th component inventory available at the beginning of period $t \in T$. The component-inventory system is assumed uncapacitated, so that $Z_i := \mathbb{R}^+$, $\forall i \in I$. Components can be purchased during any period $t \in T \setminus \{n\}$. The holding and purchasing costs are linear for all $i \in I$: $h_{it}(z_t) = h_{it} z_t$, $\forall t \in T$, and $g_{it}(q_{it}(z_t)) = c_{it} q_{it}(z_t)$, $\forall t \in T \setminus \{n\}$, with $h_{it}$ and $c_{it}$ the component- $i$ unitary holding and purchasing costs in period $t$. The coefficient $a_{ij}$ is the amount of component $i \in I$ needed to produce one unit of product $j \in J$. As in §4.1, $f_{jt}(u_{jt}(z_t, D_t)) = r_{jt} u_{jt}(z_t, D_t)$, $\forall j \in J$ and $t \in T \setminus \{n\}$. The MDP formulation closely resembles that presented in §2 and is not displayed here.

**MPs.** Two time-dependent MPs are considered: TDLP1 and TPNLP1. Model TDLP1 ignores demand uncertainty by letting $b_{js} = E[D_{js}]$ and $\varphi_{js}(x_{js}) = r_j x_{js}$, $\forall j \in J$, $s \in S(t) \setminus \{n\}$. The model is

$$\max \sum_{s \in S(t) \setminus \{n\}} \left[ \sum_{j \in J} r_j x_{js} - \sum_{i \in I} (c_{is} y_{is} + h_{is} w_{is}) \right] - \sum_{i \in I} h_{in} w_{in} \quad \text{s.t. (2)-(7)}.$$ 

Model TPNLP1 accounts for demand uncertainty by letting $\varphi_{js}(x_{js}) = E[\min\{D_{js}, x_{js}\}]$ and $b_{js} = +\infty$, $\forall j \in J$, $s \in S(t) \setminus \{n\}$. The model is

$$\max \sum_{s \in S(t) \setminus \{n\}} \left[ \sum_{j \in J} r_j E[\min\{D_{js}, x_{js}\}] - \sum_{i \in I} (c_{is} y_{is} + h_{is} w_{is}) \right] - \sum_{i \in I} h_{in} w_{in} \quad \text{s.t. (2)-(7)}.$$ 

**Decision rules.** The partitioned-allocation (PA) policy uses $x_{jt}$ to make the demand-management decision in state $z_t$ in period $t \in T \setminus \{n\}$: $u_{jt}^P(z_t, d_t; x_{jt}^*) := \min\{d_{jt}, x_{jt}^*\}$, $\forall j \in J$. The joint-allocation (JA) policy pools together all variables $x_{jt}^*$'s when making this choice:

$$u_{jt}^J(z_t, d_t; x_{jt}^*) := \arg\max_u \sum_{j \in J} r_j u_j \quad \text{s.t.} \quad \sum_{j \in J} a_{ij} u_j \leq \sum_{j \in J} a_{ij} x_{jt}^*, \forall i \in I$$

$$0 \leq u_j \leq d_{jt}, \forall j \in J.$$ 

For both policies, the component-purchasing decision-rule vector is $q_{it}^P(z_t; y_{it}^*) = q_{it}^J(z_t; y_{it}^*) := y_{it}^*$. 

**Re-solving analysis.** The following analysis shows that control algorithms in set $\{(MP, CP), \, MP \in \{TDLP1, \, TPNLP1\}, \, CP \in \{PA, \, JA\}\}$ are SC, and that if they satisfy
condition (9), then this condition can only hold as an equality. Consider time \( t \in T \setminus \{n\} \).

To simplify the notation, suppress the superscripted asterisk on the decision variables.

For \( s = t + 1 \) and an arbitrary realization of demand vector \( d_t \), the decision-rule vectors \( u_{it}^{CP}(z_t, d_t; x_t(t)) \) and \( q_{it}^{CP}(z_t; y_t(t)) \), with \( CP \in \{PA, JA\} \), are such that

\[
w_{it+1}(t) = w_{it}(t) + \sum_{j \in I} y_{jt}(t) - \sum_{j \in J} a_{ij} x_{jt}(t)
\leq z_{it} + \sum_{j \in I} q_{it}^{CP}(z_t; y_t(t)) - \sum_{j \in J} a_{ij} u_{jt}^{CP}(z_t, d_t; x_t(t)) = z_{it+1}^{CP(t)}, \forall i \in I.
\]

Then, Definition 1 implies that the updated optimal-solution is such that

\[
w_{it+1}'(t, t+1) = z_{it+1}^{CP(t)} = w_{it+1}(t)
\]

\[
w_{it+2}'(t, t+1) = z_{it+1}^{CP(t)} = w_{it+1}(t) + \sum_{i \in I} y_{it}(t) - \sum_{j \in J} a_{ij} x_{jt}(t, t+1)
\geq w_{it+1}(t) + \sum_{i \in I} y_{it}(t) - \sum_{j \in J} a_{ij} x_{jt}(t, t+1) = w_{it+2}(t), \forall i \in I.
\]

Repeating this argument for \( s = t + 3, \ldots, n \) shows that \( w_{is}(t) \leq w_{is}'(t, t+1) \) holds for all \( s \in S(t+1) \) and \( i \in I \). The feasibility of the updated optimal-solution follows from Definition 1 and the assumption that the component-inventory system is uncapacitated.

Suppose now that condition (9) holds for all \( t \in T \setminus \{n\} \) and \( z_t \in Z \). For all realizations of random variable \( Z_{i+1}^{CP(t)} \), Definition 1 and inequalities \( w_{is}(t) \leq w_{is}'(t, t+1) \), \( \forall s \in S(t+1), i \in I \), imply that

\[
\nu_{t+1}\left(z_{i+1}^{CP(t)}, \zeta(t, t+1)\right) = \sum_{s=t+1}^{n-1} \left\{ \sum_{j \in J} \varphi_{js}(x_{js}(t, t+1)) - \sum_{i \in I} [c_{is}y_{is}(t+1) + h_{is}w_{is}'(t, t+1)] \right\}
- \sum_{i \in I} h_{is}w_{is}(t, t+1)
\leq \sum_{s=t+1}^{n-1} \left\{ \sum_{j \in J} \varphi_{js}(x_{js}(t)) - \sum_{i \in I} [c_{is}y_{is}(t) + h_{is}w_{is}(t)] \right\} - \sum_{i \in I} h_{is}w_{is}(t)
= \sum_{s=t+1}^{n-1} \psi_{s}(z_t, \zeta(t)),
\]

so that \( E_t\left[\nu_{t+1}\left(Z_{i+1}^{CP(t)}, \zeta(t, t+1)\right)\right] \leq \sum_{s=t+1}^{n-1} \psi_{s}(z_t, \zeta(t)) \), and condition (9) must hold as an equality. In general, when the holding-cost coefficients are positive, it is highly unlikely that (9) would always hold as an equality. However, it is clear that this is the case when they are zero. Under this assumption, Propositions 6 and 7 establish the re-solving behaviors
of control algorithms (TDLP1, PA) and (TDLP1, JA), and (TPNLP1, PA) and (TPNLP1, JA), respectively.

**Proposition 6 (Re-solving with (TDLP1, PA) and (TDLP1, JA)).** If \( Z_i = \mathbb{R}^+ \) and \( h_{it} = 0, \forall i \in I \) and \( t \in T \), then \( v_1^{PA_1,TDLP1}(z_1) \leq E_1 \left[ V_n^{(TDLP1,PA)}(z_1) \right] + \alpha_n^{(TDLP1,PA)} \) and \( v_1^{JA_1,TDLP1}(z_1) \leq E_1 \left[ V_n^{(TDLP1,JA)}(z_1) \right] + \alpha_n^{(TDLP1,JA)} \).

**Proof.** As shown before, control algorithms (TDLP1, PA) and (TDLP1, JA) are SC under the first assumption. Condition (9) holds for both control algorithms under the second assumption. Similarly to Cooper (2002, p. 724), for any policy \( \pi \in \Pi \) the expectations of random vectors \( Z_\pi^t, q_\pi^t(Z_\pi^t), \) and \( u_\pi^t(Z_\pi^t, D_t) \) are a feasible solution to TDL1, so that 

\[
v_1^\pi(z_1) \leq \nu_1^\pi(z_1) \text{ and Proposition 3 applies with } \alpha_1^{(TDLP1,PA)} = \alpha_1^{(TDLP1,JA)} = 0. \]

**Proposition 7 (Re-solving with (TPNLP1, PA) and (TPNLP1, JA)).** If \( Z_i = \mathbb{R}^+ \) and \( h_{it} = 0, \forall i \in I \) and \( t \in T \), then \( v_1^{PA_1,TPNLP1}(z_1) \leq E_1 \left[ V_n^{(TPNLP1,PA)}(z_1) \right] \) and \( v_1^{JA_1,TPNLP1}(z_1) \leq E_1 \left[ V_n^{(TPNLP1,JA)}(z_1) \right] + \alpha_1^{(TPNLP1,JA)} + \alpha_n^{(TPNLP1,JA)} \).

**Proof.** It has already been shown that control algorithms (TPNLP1, PA) and (TPNLP1, JA) are SC under the first assumption. Under the second assumption, TPNLP1 is PA-accurate and, by Proposition 2, (TPNLP1, PA) is SI. Therefore, the result for this control algorithm follows from Proposition 1. Since condition (9) holds for (TPNLP1, JA), the second claimed inequality follows from Proposition 3.

**Remark 5 (Re-solving with (TPNLP1, PA)).** The component-purchasing decision rules of the policy of control algorithm (TPNLP1, PA) obtained at time 1 are likely to order “too much” at the beginning of time periods \( t + 1, \ldots, n - 1 \), because TPNLP1 makes component-replenishment decisions assuming that inventory depletes according to the maximum accepted demand of each product. Then, intuitively, re-solving should be beneficial for this control algorithm because replenishments can be based on the true available inventory. Proposition 7 supports this informal argument.

## 5. Conclusions

This paper studies the control-algorithm re-solving issue for a class of inventory/revenue-management problems. It formalizes the control-algorithm approach to heuristically solve a Markov decision process formulation of this class of problems, it develops a theory of
re-solving, and it applies it to problems that arise in the contexts of network revenue-management and multiproduct make-to-order production.

This paper shows that when the control algorithm is sequentially improving re-solving is (weakly) beneficial. If the math program of a given control algorithm is an accurate representation of the control policy and the control algorithm is sequentially consistent, then the control algorithm is also sequentially improving and, therefore, re-solving cannot worsen its performance. If the math program is an inaccurate representation of the control policy, it is time-dependent, it satisfies a technical condition, and the control algorithm is sequentially consistent, then a weaker result can be established. While sequential consistency does not necessarily characterize a sequentially-improving control algorithm, overall it emerges as an important property to obtain positive re-solving results.

From an application standpoint, this paper individuates two related network revenue-management control algorithms for which re-solving is provably beneficial, and shows that lack of sequential consistency is one of the reasons explaining the odd re-solving behavior exhibited by a linear-programming-based control algorithm in the example discussed by Cooper (2002). For the multiproduct make-to-order-production problem, structural results can be established by assuming that the inventory holding cost is zero and the component-inventory is uncapacitated.

From a practical standpoint, when a control algorithm suffers substantially from re-solving, Proposition 1 suggests that the following simple, albeit potentially computationally demanding, algorithmic adjustment eliminates the re-solving issue. (This method is analogous to the fortified rollout algorithm discussed by Bertsekas, Tsitsiklis and Wu 1997 in the context of combinatorial optimization problems.) At each re-solving time, evaluate, perhaps by simulation, the control policy using both the new and the updated math-program solution and retain the one with better performance. Since this algorithmic adjustment makes the control algorithm sequentially improving (ignoring any potential simulation error), re-solving cannot worsen its performance.

As suggested by one of the referees, while it is tempting to conclude that with a sequentially-improving control algorithm re-solving more often is always beneficial, this statement should be treated with care. This paper establishes that re-solving is advantageous for sequentially-improving control algorithms, but does not deal with the problem of how to select the re-solving times. For example, suppose that the planning horizon includes three periods. Then, for the same sequentially-improving control algorithm, the results of this paper do not
help to tell whether re-solving at all times would necessarily be better than only re-solving at the first and third times, but not at the second time. How to select the re-solving times is a topic that deserves further research.

This paper has analyzed the control-algorithm re-solving issue for a class of problems with no data approximations and a finite horizon that contracts with the passage of time, i.e., the end of the planning horizon does not shift. Extending the analysis to deal with data approximation or finite rolling-horizons, for which the length of the planning horizon does not shrink with the passage of time, are additional research avenues. Finally, it is an open question whether a positive re-solving result could be established for a control algorithm with time-aggregate and inaccurate math-program, and how such a result would relate to sequential consistency.

Acknowledgments

The suggestions of two anonymous referees, the senior editor, and the Editor in Chief, Garrett van Ryzin, led to a substantially improved version of this paper. The comments of Bill Cooper and Darius Walczak on an initial version of this paper are also gratefully acknowledged.

References


