Received: 1 August 2018 | Revised: 27 January 2020 | Accepted: 31 May 2020 | Published on: 20 July 2020
DOI: 10.1002/nav.21931

RESEARCH ARTICLE

WILEY

Optimal policies for stochastic clearing systems with time-dependent delay penalties

Qi-Ming He | **James H. Bookbinder** | **Qishu Cai**

Department of Management Sciences, University of Waterloo, Waterloo, Ontario, Canada

Correspondence
Qi-Ming He, Department of Management Sciences, University of Waterloo, Waterloo, ON N2L 3G1, Canada.
Email: q7he@uwaterloo.ca

Funding information
Natural Sciences and Engineering Research Council of Canada.

Abstract

We study stochastic clearing systems with a discrete-time Markovian input process, and an output mechanism that intermittently and instantaneously clears the system partially or completely. The decision to clear the system depends on both quantities and delays of outstanding inputs. Clearing the system incurs a fixed cost, and outstanding inputs are charged a delay penalty, which is a general increasing function of the quantities and delays of individual inputs. By recording the quantities and delays of outstanding inputs in a sequence, we model the clearing system as a tree-structured Markov decision process over both a finite and infinite horizon. We show that the optimal clearing policies, under realistic conditions, are of the on-off type or the threshold type. Based on the characterization of the optimal policies, we develop efficient algorithms to compute parameters of the optimal policies for such complex clearing systems for the first time. We conduct a numerical analysis on the impact of the nonlinear delay penalty cost function, the comparison of the optimal policy and the classical hybrid policy (ie, quantity and age thresholds), and the impact of the state of the input process. Our experiments demonstrate that (a) the classical linear approximation of the cost function can lead to significant performance differences; (b) the classical hybrid policy may perform poorly (as compared to the optimal policies); and (c) the consideration of the state of the input process makes significant improvement in system performance.

KEYWORDS

delay penalties, Markovian arrival process, shipment consolidation, stochastic clearing system

1 | INTRODUCTION

A stochastic clearing system receives and accumulates inputs of random quantities over random time intervals until certain predetermined criteria are met; then, there is a *clearing* of some or all of those inputs instantaneously. There are many practical applications of this type of systems in manufacturing, transportation, tourism, and healthcare industries, for example, (a) shuttle bus service at airports or train stations; (b) food delivery service for restaurants; and (c) goods shipments in transportation (this is the well-known *shipment consolidation problem*).

For the management of stochastic clearing systems, a key issue is to decide when and how much to clear (ie, to depart, deliver, or dispatch) in order to gain economies of scale. In the existing models of stochastic clearing systems, the aggregate total quantity is usually the only variable used to keep track of the system state and for decision making. Thus, a recently acquired input is perceived as equally urgent as another input that was received a long time ago. It is easy to see the limitation of this assumption since the time spent in a system is an important factor in deciding when a given input must be cleared. In this paper, we introduce a new formalism to model stochastic clearing systems and a novel approach to analyze and optimize them. More specifically, we introduce a sequence of numbers to record the accumulated input delays (i.e., elapsed times) and quantities that are necessary for system control. This set of information can enable better clearing decisions with respect to the varying levels of urgency among inputs. The system state space, based on the input sequence, possesses a tree structure that is utilized by our algorithms for computing the optimal clearing policies.

In this paper, we conduct our analysis on a quite general and complicated model. First, we use a batch Markovian arrival process (BMAP) to model both the quantities and interarrival times of inputs. Such an input process generalizes the discrete analog of the compound Poisson process, and is particularly potent at capturing the correlations between different inputs, as well as the variation of input rate over time. Second, we use more general functions for the fixed clearing costs and the variable delay penalty costs. Those functions are assumed to be *increasing* in both the quantities and delay times of inputs. The variable delay penalty cost (e.g., inventory carrying cost) portion of the total cost function has a different interpretation in shipment consolidation (SCL) than it does in the determination of optimal quantities in production or stock-control applications. In the latter two cases, the decision-making firm owns the goods in question. In the SCL case, the firm deciding when to dispatch is often a trucking company that does not own the goods. In the SCL context, although we may call it inventory carrying cost, that cost term represents the disutility of a customer whose order has not yet been shipped. It is well known that this disutility is an increasing function of the customer waiting time. To the best of our knowledge, this is the first time in the literature that both the delays and the quantities of individual inputs are modeled in nonlinear fashion for the analysis and optimization of stochastic clearing systems. Both the input process and the cost structure of the clearing system investigated in this paper are far more complicated and realistic than those in the existing literature. The use of the sequence of delays and quantities, the general input process, and the general cost functions constitute the main modeling contribution of this paper.

While the preceding assumptions render the clearing systems more complex but more realistic, their analysis and optimization become challenging. For example, the tree-structured state space is complicated for analysis and can be large in size for computation. To attack the space complexity and space dimensionality issues, we

- utilize the Markov decision process (MDP) and matrix-analytic methods, which make the clearing systems analytically and numerically tractable;
- bring in the partial order of vector dominance to compare system states, and show that, under realistic conditions, the optimal clearing policy is of the *on-off* type or of the *threshold* type, which constitutes the main methodological contribution of the paper; and
- based on the characterization of the optimal policies, combine the value-iteration method for MDP, matrix-analytic methods, and the branch-and-bound method to develop efficient algorithms for finding the optimal clearing policies, which constitutes the main algorithmic contribution of the paper and is the first in the study of complex clearing systems.

Using the computational methods, in Section 6, we conduct an in-depth numerical analysis on issues such as: (a) the benefit of employing nonlinear delay penalty cost functions of both the delays and quantities of individual inputs; (b) the necessity to use optimal policies based on delays and quantities of individual inputs (vs the classical hybrid policy); and (c) the benefit in utilizing Markov modulated processes to model the inputs. Numerical examples demonstrate a significant gain in system performance of the proposed models, which justifies the added complexity in system modeling and optimization.

The remainder of this paper is organized as follows. In Section 2, a comprehensive literature review is presented. The stochastic model of interest is introduced in Section 3. In Section 4, for the finite time horizon case, the optimal policies are characterized and a computational method is developed. Section 5 deals with the infinite horizon case with discounted total cost. Three examples are examined numerically in Section 6. Section 7 concludes the paper. Proofs of all theorems are provided in the Appendix.

2 | LITERATURE REVIEW

Stochastic clearing systems were first carefully modeled and studied by Stidham (1974). In that paper, the process was considered as regenerative, and an explicit expression for the stationary distribution of the quantity in the system was derived. Subsequently, Stidham (1977) generalized the clearing operations so that not everything must be cleared at once. While the development of stochastic clearing systems has been ongoing for decades, more research has recently been done on its sub- or related areas: inventory control, shipment consolidation, and queueing control. Thus, our literature review will focus on those three types of stochastic models.

Inventory control was touted as one of the main applications of stochastic clearing systems. Inventory problems belong to the broader definition of stochastic clearing systems in two ways. First, if the system keeps no stock on hand, but instead backlogs all demands and satisfies them later, those backlogged demands are the inputs to the stochastic clearing system, and instantaneous inventory replenishment is equivalent to the clearing event. Second, more generally, we can treat any stock replenishment as a clearing, after which the inventory is reset to a desirable level. Then, any demand received after replenishment can be considered an input to the system; the total amount sold is the accumulated input quantity. Some inventory models, such as systems with perishable products, have both inventory and clearance features (Li & Yu, 2013; Li, Yu, & Wu, 2016). One of the well-known inventory results is the optimality of the (s, S) policy, which was first shown by Scarf (1960) using the concept of *K-convexity* (also see Iglehart, 1963; Veinott &

Wagner, 1965; Zheng, 1991). Stidham (1986) and Kim and Seila (1993) studied inventory models with an (s, S) policy from stochastic clearing system's point of view, and identified necessary and sufficient conditions on the cost function and input process for optimality of the clearing parameters. Recent inventory literature studied systems with Markovian demand processes. Song and Zipkin (1993) introduced a model where the demand rate varies with an underlying state-of-the-world variable (see also Beyer, Cheng, Sethi, & Taksar, 2010; Chen & Song, 2001). In those works, the optimal policies were proved to be state-dependent (s, S) policies, which is significant due to the simplicity of the optimal control policy. The importance of such studies is further increased by the fact that the Markovian arrival process can approximate any stochastic input/demand processes. In this paper, the input process of our stochastic clearing system is assumed to be a Markovian input process (see Asmussen & Koole, 1993). We find the state-dependent and on-off/threshold type optimal clearing policies for our systems. Although information on the state of the input process is usually not available in applications, the optimal policies can lead to good heuristic clearing policies that do not depend on the state to improve system performance. We refer to Xia, He, and Alfa (2017) for a detailed discussion with numerical examples on this issue for some queueing control problems.

Shipment consolidation is a logistics strategy whereby many small shipments are combined into a few larger loads to achieve economies of scale. Such systems are among the most natural examples of stochastic clearing systems. Although the main purpose of shipment consolidation is to minimize overall costs, that should not be at the expense of unsatisfactory customer service. By associating appropriate monetary values to the delays of orders, achieving an optimal balance between cost reductions and maintaining good service becomes the ultimate goal of that strategy. Shipment consolidation models have been investigated extensively (see Bookbinder & Higginson, 2002; Çetinkaya, Tekin, & Lee, 2008; Higginson & Bookbinder, 1994, 1995; Mutlu, Çetinkaya, & Bookbinder, 2010). Bookbinder, Cai, and He (2011) modeled the order arrival process by a discrete time *BMAP*. An efficient computational procedure was developed for evaluating classical dispatch policies (ie, [total] quantity-threshold policy, age-threshold policy, and the hybrid policy) against a set of performance measures. More recently, Cai, He, and Bookbinder (2014) proposed to use a tree structured Markov chain to record information about the consolidation process, specifically the quantities and waiting times of individual orders. A heuristic algorithm was developed to determine the parameter of a special set of dispatch policies, and the algorithm was proved to yield the overall optimal policy under certain conditions. The clearing model introduced in this paper is more general than most of the models considered in the literature. The closest are the transhipment models considered Bookbinder et al. (2011) and Cai et al. (2014), in which the focus is on steady-state analysis, not the optimal policy. In Table 1, we compare our clearing model assumptions to those in the literature.

The present paper generalizes the cost structure to include *nonlinear* functions for both the delay costs and the clearing costs. More importantly, this paper focuses on finding the optimal clearing policy, and the computational methods developed in this paper can be used for evaluating any given clearing policy as well.

The concepts of stochastic clearing systems have also been applied to queueing systems. For example, Boxma, Perry, and Stadje (2001) studied the clearing models for M/G/1 queues, in which events called "disasters" occur at certain random times, causing an instantaneous removal of the entire residual workload from the system. In a similar line of research, Dudin and Karolik (2001) and Inoue and Takine (2014) considered queueing systems with a Markovian arrival process and potential exposure to disasters. Those authors calculated quantities related to the embedded and arbitrary time queue length distributions, waiting time distributions, as well as the average output rate and loss probability.

Our literature review shows that the early publications on stochastic clearing systems involved a general problem framework. Since then, research has focused on particular application areas, mainly inventory control, shipment consolidation problems, and queueing system management. However, all these models have assumed that the cost to hold the accumulated inputs is charged at a *constant* rate, proportional to the total quantity on hand at any time. That assumption is sufficient in some areas, but not for others, especially those such as shipment consolidation and shuttle bus dispatch, where the longer an input stays in the system, the more expensive the cost rate is perceived to be. From the modeling perspective, the present literature on stochastic clearing systems is ill-equipped to reflect some delicate real situations, in which the input process is correlated and costs depend on the status

TABLE 1 Model assumption comparison

	This paper	Bookbinder et al. (Bookbinder et al., 2011)	Cai et al. (Cai et al., 2014)	Literature
Input	*DBMAP*	*DBMAP*	*DBMAP*	Geometric
Cost	General delay cost; general clearing cost	General delay cost; constant clearing cost	General delay cost; general clearing cost	Linear delay cost; convex clearing cost
System state	Individual orders (tree structure)	Aggregate quantities (eg, quantity, age)	Individual orders (tree structure)	Aggregate quantities

of individual inputs. In this paper, we try to fill the gap by eliminating the deficiency of the input process and cost structure in the current literature, which is the main contribution of the present research.

3 | THE MODEL OF INTEREST

We define a typical stochastic clearing system in which inputs of random quantities are received, accumulated, and cleared. We assume that the planning horizon is discrete and divided into *N periods* or *N decision epochs*, where N can be finite or infinite. At the beginning of each period, a clearing decision is made according to a clearing policy, and that decision is executed immediately. Within each period, inputs of a random quantity arrive and become part of the total quantity outstanding. In the rest of this section, we define four components of the clearing system explicitly: (i) the input process; (ii) system state space; (iii) decision rules and clearing policies; and (iv) cost structure.

3.1 | The input process

Inputs arrive according to a *discrete time batch Markovian arrival process* (*DBMAP*) (see Neuts, 1979). Let $\{i_t, t = 0,1,2, \ldots\}$ be a discrete time homogeneous Markov chain with states $\{1, 2, \ldots, M\}$ and transition matrix D. In the system, the probability that an input of quantity q_t arrives in period t depends on the state of the Markov chain. Specifically, we assume that $\{(q_t, i_t), t = 1, 2, \ldots\}$ is a Markovian arrival process with matrices $\{D_0, D_1, \ldots, D_Q\}$ such that, in state i, $(D_q)_{i,j}$, for $q = 0, 1, \ldots, Q$, is the probability that $q_t = q$ units of input arrive while the underlying process i_t changes from state i at the beginning of the current period t to state j at the beginning of the next period. Here Q is the maximum possible quantity of an input. It is easy to see that $D = D_0 + D_1 + \cdots + D_Q$. The following are two special cases of *DBMAP*.

Example 3.1.1 The well-known *compound renewal process* is a *DBMAP* with $M = 1$.

Example 3.1.2 *Markov modulated arrival process* is a *DBMAP* with $D_q = diag(f_1(q), \ldots, f_M(q))D$, where $f_i(q) = Pr\{q_t = q | i_t = i\}$, for $i = 1, 2, \ldots, M$, and "*diag*" means a diagonal matrix.

The state of underlying process of the input process provides information on the residual arrival time and the amount of the next input. In general, optimal clearing policies depend on the state, and yet information on the state is not available in practice. For applications, heuristic policies based on the optimal clearing policies can be introduced, which can perform significantly better than classical clearing policies (eg, the quantity, age, or hybrid policy). Again, we refer to Xia et al. (2017) for a numerical discussion on the issue.

3.2 | System state space and vector dominance

For stochastic clearing systems, a system state variable is usually introduced to track the total quantity of all outstanding inputs. Our model extends the existing ones by simultaneously recording the *delays* of individual inputs, that is, the time elapsed since each input was received, and the quantities of individual inputs. To do so, we use a *sequence* (ie, *string*) of nonnegative integers.

Suppose that the current period is t. Let $x_{[j]}$ be the (remaining) quantity of the input that arrived in period $t-j$, where j is the delay time of the input, and $j = 1, 2, \ldots, t$. We call $x_{[j]}$ the *remaining* quantity, since part of the input that arrived in period $t-j$ may have been cleared before period t. We define $x_t = [x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}]$, where l is the age of the oldest input that is still not completely cleared, $\{x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}\}$ are quantities of inputs still in the system at the beginning of period t: $x_{[l]}$ is the quantity of the oldest input (that has not been cleared completely), $x_{[l-1]}$ the second oldest, …, and $x_{[1]}$ the youngest that just arrived in period $t-1$. Then x_t captures the state of the system before the clearing decision is made in period t. We shall refer to x_t as the *pre-clearing system content* in period t. For example, if $x_8 = [4,2,3]$, there are three inputs in the system at the beginning of period eight; the oldest input that arrived in period five has quantity 4; the second oldest input that arrived in period six has quantity 2; and the youngest input that arrived in period seven has quantity 3.

Analogous to x_t, we define the *post-clearing system content* as y_t, which records outstanding inputs remaining in the system immediately after the clearing decision is carried out and before any new input arrives in period t. The differences between x_t and y_t are the outstanding quantities that are cleared in period t, which we shall denote as w_t. For the above example, in period eight, suppose that we decide to clear the input that arrived in period five and 2 units of the input that arrived in period seven; then we have $w_8 = [4,0,2]$ and $y_8 = [0,2,1]$. Apparently, we must have $x_8 = w_8 + y_8$, or more generally $x_t = w_t + y_t$. Suppose further that an input of 25 units arrives in period eight, then we have $x_9 = [y_8, 25] = [0,2,1,25]$.

If the system is empty at the beginning of period t, we set $x_t = \emptyset$ or $x_t = [0, \ldots, 0]$. Under the context of our model, sequences $[0, \ldots, 0, x_{[l]}, \ldots, x_{[1]}]$ and $[x_{[l]}, \ldots, x_{[1]}]$ contain essentially the same information and can thus be used interchangeably. With that said, the state space of x_t can be defined by

$$\Phi = \{\emptyset\} \cup \bigcup_{l=1}^{\infty} \{[x_{[l]}, \ldots, x_{[1]}] : x_{[l]} > 0]\}. \tag{1}$$

Together with the states of the underlying Markov chain of the input process, we define the set of *potential system states* as $\Omega = \Phi \times \{1, 2, \ldots, M\}$ for (x_t, i_t). Recall that i_t is the state of the input process. For our analysis, the following operations associated with sequences in Φ are used.

Definition 3.1 For any pair of sequences $x = [x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}]$ and $y = [y_{[k]}, y_{[k-1]},$

$\ldots, y_{[1]}]$ in Φ, we define the following operators:

i. sequence concatenation: $x \frown y = [x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}, y_{[k]}, y_{[k-1]}, \ldots, y_{[1]}]$.
ii. sequence addition: $x + y = [x_{[l]}, \ldots, x_{[k+1]}, x_{[k]} + y_{[k]}, \ldots, x_{[1]} + y_{[1]}]$, if $l \geq k$.
iii. sequence sum: $|x| = \sum_{j=1}^{l} x_{[j]}$;
iv. sequence length: $L(x) = l$. (Note: $x_{[l]} > 0$.)

For $x = [x_{[l]}, x_{[l-1]}, \ldots, x_{[1]}]$ and $y = [y_{[l]}, y_{[l-1]}, \ldots, y_{[1]}]$, $x \leq y$ if and only if $x_{[j]} \leq y_{[j]}$, for all $j = 1, 2, \ldots, l$. Furthermore, we define partial order $\leq_d$, that is, *vector dominance*, as: $x \leq_d y$ if $\sum_{i=1}^{j} x_{[l+1-i]} \leq \sum_{i=1}^{j} y_{[l+1-i]}$, for $j = 1, \ldots, l$. For example, if $x = (1,1,1.2)$ and $y = (2,0.5,1)$, we have $x \leq_d y$ but not $x \leq y$. For two sequences in Φ with different lengths, we attach zeros to the left of the shorter one to make the two sequences equal in length. In this way, we allow the comparison of any pair of sequences in Φ. If $x \geq y$, we define $z = x - y$, for which z satisfies $x = z + y$. We also note that $x \leq y$ implies that $x \leq_d y$.

3.3 | Decision rules and clearing policies

At the beginning of each period, we must decide whether to continue accumulating inputs, clear some outstanding inputs, or clear everything in the system. Let $\mathcal{A}_t$ be the set of all available actions at the beginning of period t. Since we can clear any amount of outstanding inputs, actions to be chosen in period t depend on the pre-clearing system content x_t as well as the state of the input process i_t. We refer to the decision rule in period t as the *clearing rule*, which specifies the clearing action that must take place in period t. Thus, a clearing rule r_t is a mapping from the system state space Ω to the available action set $\mathcal{A}_t$, that is, $r_t : \Omega \to \mathcal{A}_t$, where $r_t(x_t, i_t) = a_t \in \mathcal{A}_t$ implies that, if the system state is (x_t, i_t) at the beginning of period t, action a_t must be taken. We denote the set of all such clearing rules by $\mathcal{R}_t$, for $t = 1, 2, \ldots, N$.

We call an action a partial clearing if only some outstanding inputs are cleared. If an action clears all existing outstanding inputs, we call it an *on-off control* (a.k.a. *bang-bang control*). In the case of on-off control, we define $\mathcal{A}_t = \{0, 1\}$ such that $a_t = 0$ means to continue to accumulate in period t and $a_t = 1$ means to clear the system immediately. Consequently, $r_t(x, i) = 0$ for $(x, i) \in \Omega$ (or $r_t(x, i) = 1$) means continue to accumulate (or clear the system), given that system state is (x, i) at the beginning of period t. Here are a few examples of clearing rules.

Example 3.3.1 The classical quantity threshold rule: $r_t(x_t, i_t) = 0$, if $|x_t| < 5$; 1, otherwise. This clearing rule is of the on-off type, and depends only on the total quantity of all existing inputs.

Example 3.3.2 The classical age threshold rule: $r_t(x_t, i_t) = 0$, if $L(x_t) < 4$; 1, otherwise. This clearing rule is also of the on-off type, and depends only on the age of the oldest (nonzero) input.

Example 3.3.3 Hybrid rule (w, T): For $w = 3$ and $T = 3$, $r_t(x_t, i_t) = 0$, if $|x_t| < 3$ and $L(x_t) < 3$; 1, otherwise. This clearing rule is of the on-off type, and depends on both the total quantity and the age of the oldest input.

A *clearing policy* π specifies the clearing rules to be used at all decision epochs. Such a policy consists of a series of clearing rules, that is, $\pi = (r_1, r_2, \ldots, r_N)$, where $r_t \in \mathcal{R}_t$, for all $t = 1, 2, \ldots, N$.

3.4 | Cost structure and the expected Total costs

Two types of costs are relevant to our analysis of stochastic clearing systems: a *delay penalty cost*, a.k.a. the disutility of waiting, or the holding cost in many practical applications, is incurred during every period that an input remains in the system; and a *fixed clearing cost* that is incurred when an actual clearing takes place.

We assume that the delay penalty cost in each period is calculated based on the post-clearing system content y_t. Denote the delay penalty cost by H_t for period t. Then H_t is a function defined on Φ (the state space of y_t). For most clearing systems considered in the literature, H_t is a linear function of the total accumulated quantity, for example, $H_t = 0.1|y_t|$. In our model, since y_t contains other information in addition to the accumulated quantity, more choice of H_t can be introduced and used in the analysis of clearing systems. Here are a few examples of the delay penalty cost. Let $y_t = [y_{[l]}, \ldots, y_{[1]}]$.

Example 3.4.1 $H_t(y_t) = 0.1|y_t|^2$ (a nonlinear function of the total accumulated quantity).

Example 3.4.2 $H_t(y_t) = 0.1(L(y_t))^2$ (a nonlinear function of the longest delay).

Example 3.4.3 $H_t(y_t) = 1.5\sum_{j=1}^{l} j^2 \cdot y_{[j]} + 2.5 \sum_{j=1}^{l} (j \cdot y_{[j]})^2$, if $|y_t| \leq L$, and $H_t(y_t) = \infty$, otherwise, where L is a positive constant (a nonlinear function in both the quantity and delay of each input with a clearing capacity constraint).

For arbitrary t, we assume that the fixed clearing cost in that period is given by the function $K_t(w_t) = \hat{k}_t \cdot \delta_{\{|w_t|>0\}}$, where $w_t = x_t - y_t$ (which is the content cleared in period t) and $\delta_{\{\cdot\}}$

is the indicator function. Here are a few examples of the fixed clearing cost.

Example 3.4.4 $K_t(w_t) = \hat{k} \cdot \delta_{\{|w_t|>0\}}$ (a conventional fixed cost function).

Example 3.4.5 $K_t(w_t) = (k_1 + k_2 \max\{0, N-t\} + k_3/t)\delta_{\{|w_t|>0\}}$ (a nonlinear function, decreasing in time).

We assume that the entire system has to be cleared at a fixed cost by the end of period N (or at the beginning of period $N+1$). We denote by $C^{\pi,\alpha}_{<t_1,t_2>}(x, i)$ the expected total cost incurred in periods t_1, t_1+1, …, t_2, where (x, i) is the initial system state in period t_1, $<t_1, t_2>$ is the time horizon, π is the clearing policy, and α is the discount factor ($0 < \alpha \leq 1$). Utilizing the above notation, the *expected total discounted cost* over the entire planning horizon can be calculated by

$$C^{\pi,\alpha}_{<1,N>}(x, i) = \mathbb{E}\left[\sum_{t=1}^{N} \alpha^{t-1}(H_t(y_t) + K_t(w_t))|(x_1, i_1) = (x, i)\right]. \quad (2)$$

In Sections 4 and 5, we develop algorithms to compute the clearing policy π that minimizes the above cost function.

4 | OPTIMAL POLICIES OVER A FINITE HORIZON

In this section, we study the stochastic clearing problem over a finite planning horizon (that is: $N < \infty$). The objective is to find the clearing policy that minimizes the expected total cost of Equation (2) with discount factor $\alpha = 1$. For that purpose, we first formulate and analyze our problem as a *MDP* (Puterman, 2005).

Given a clearing policy $\pi = (r_1, r_2, \ldots, r_N)$, the cost incurred in period t is given by

$$u^{\pi}_t(x, i) = H_t(y) + K_t(x - y), \quad (3)$$

where y is the post-clearing system content, which is determined by the clearing rule $r_t(x, i)$, if the initial system state is (x, i) in period t. Subject to the clearing policy π, the process $\{(x_t, i_t), t = 1, 2, \ldots\}$ is a Markov chain. At an arbitrary decision epoch n, the expected total cost starting from period n until the end of the planning horizon can be rewritten as $C^{\pi}_{<n,N>}(x, i) = \mathbb{E}\left[\sum_{t=n}^{N} u^{\pi}_t(x_t, i_t) \mid (x_n, i_n) = (x, i)\right]$, for all $n = 1, 2, \ldots, N$. For any given initial system state (x, i) in period n, the Markovian property of $\{(x_t, i_t), t = 1, 2, \ldots\}$ allows us to write $C^{\pi}_{<n,N>}(x, i)$ recursively as

$$C^{\pi}_{<n,N>}(x, i) = u^{\pi}_n(x, i) + \mathbb{E}\left[C^{\pi}_{<n+1,N>}(x_{n+1}, i_{n+1}) \mid (x_n, i_n) = (x, i)\right]. \quad (4)$$

Recall that everything has to be cleared at constant cost by the end of the planning horizon. Thus, we have $C^{\pi}_{<N+1,N>}(x, i) = K_{N+1}(x)$.

Denote by $\pi^* = (r^*_1, r^*_2, \ldots, r^*_N)$, the clearing policy that minimizes the expected total cost during the entire planning horizon, given that the initial system state is (x, i). That is, $C^{\pi^*}_{<1,N>}(x, i) \leq C^{\pi}_{<1,N>}(x, i)$ for any policy π. By the principle of optimality, the following optimality equations hold for the optimal clearing policy π^*:

$$C^{\pi^*}_{<t,N>}(x, i) = \inf_{y \leq x}\left\{K_t(x - y) + H_t(y) + \mathbb{E}\left[C^{\pi^*}_{<t+1,N>}(y \frown q_t, i_{t+1}) \mid i_t = i\right]\right\}, \quad (5)$$

for $t = 1, 2, \ldots, N$. Note that y is the post-clearing system content, q_t is the input in period t, and $x_{t+1} = y \frown q_t$ in the above equation. We define the *value function* for the problem over the time interval $<t, N>$, with initial system state $(x_t, i_t) = (x, i)$, to be $V_t(x, i) = \inf_{\pi \in R_1 \times R_2 \times \cdots \times R_N} C^{\pi}_{<t,N>}(x, i)$. Then Equation (5) can be written as: $V_{N+1}(x, i) = K_{N+1}(x)$, and

$$V_t(x, i) = \inf_{y \leq x}\left\{K_t(x - y) + H_t(y) + \mathbb{E}\left[V_{t+1}\left(y \frown q_t, i_{t+1}\right)|i_t = i\right]\right\}, \quad \text{for } t = 1, 2, \ldots, N. \quad (6)$$

Since the cost functions depend on both the quantities and the elapsed times of individual inputs, the structure of the optimal clearing policy can be complicated. For instance, the optimal clearing decision for states $x_1 = [5,0,1]$ and $x_2 = [1,0,5]$ can be different, even though they have the same accumulated quantity. The optimal clearing policy can be found by solving Equation (6) recursively. However, the state space for searching the optimal policy can be too big, due to its tree structure.

In the rest of this section, upon imposing some realistic conditions on the cost functions, we identify the structure of the optimal policy (Section 4.1) and develop an algorithm for computing the optimal policy (Section 4.2).

4.1 | Characterization of the optimal policy

In this subsection, we establish the existence and some properties of the optimal clearing policy if the cost functions satisfy:

Condition 4.1

i. $\hat{k}_t \geq \hat{k}_{t+1} \geq \cdots \geq \hat{k}_{N+1} = 0$, for $t = 1, 2, \ldots, N$.
ii. $0 = H_t(\emptyset) \leq H_t(x') \leq H_t(x)$, for all $x' \leq_d x \in \Phi$ and $t = 1, 2, \ldots, N$. (Recall that "$\emptyset$" means an empty system.)

Condition 4.1(i) assumes that the clearing cost is nonincreasing in time, which is consistent with the fact that, as the system evolves, it becomes more efficient in operation so that the fixed costs are nonincreasing. Condition 4.1(ii) assumes that the delay penalty costs are increasing in both the quantities and delay times (ages) of outstanding inputs, which is realistic in practice. Those are an important contribution of the paper, which is supported by examples in inventory management and shipment consolidation as we have discussed in Section 1.

Defining

$$W_t(y, i) = H_t(y) + \mathbb{E}[V_{t+1}(y \frown q_t, i_{t+1}) | i_t = i], \quad \text{for } t = 1, 2, \dots, N, \tag{7}$$

we can rewrite the dynamic program (6) as $V_{N+1}(x, i) = 0$, and

$$V_t(x, i) = \inf_{y \le x}\{K_t(x - y) + W_t(y, i)\}, \quad \text{for } t = 1, 2, \dots, N. \tag{8}$$

Under Condition 4.1, it can be shown that there exists an on-off type optimal clearing policy.

Theorem 4.1 *Under Condition* 4.1, *there exists an on-off optimal clearing policy, that is, if a clearing is triggered then all accumulated inputs must be cleared, such that*

$$V_t(x, i) = \min\{K_t(x) + W_t(\emptyset, i), W_t(x, i)\}, \quad \text{for } t = 1, 2, \dots, N. \tag{9}$$

Although the on-off feature of the optimal policy has been found, the actual representation of that policy can be very cumbersome. However, we next show that, subject to the following Condition 4.2 on the delay penalty function, the optimal policy can actually be a *state-dependent threshold policy*, which is defined as

Definition 4.1 A state-dependent threshold clearing policy in a finite horizon is denoted as $\pi^{\tau} = (r_1^{\tau}, \dots, r_N^{\tau})$. For period t, the clearing rule r_t^{τ} is determined by a set of parameters $\{\tau_{t,1}, \dots, \tau_{t,M}\}$, for which we have

$$r_t^{\tau}(x, i) = \begin{cases} 0, & \text{if } H_t(x) \le \tau_{t,i}; \\ 1, & \text{otherwise}, \end{cases} \tag{10}$$

for all $(x, i) \in \Omega$. (Note that Ω is defined right after Equation (1).)

Such a policy is parameterized by a single number (for each period t and $i = 1, 2, \dots, M$), as opposed to depending on the set of all possible states.

Condition 4.2 $H_t(x) \le H_t(y)$ implies $H_{t+1}(x \frown q) \le H_{t+1}(y \frown q)$, for $t = 1, 2, \dots, N$, $x, y \in \Phi$, and $0 \le q \le Q$. (Note that Φ is defined in Equation (1).)

We would like to point out that (a) since $x \le_d y$ is not required in Condition 4.2, that condition is not a consequence of part (ii) in Condition 4.1; and (b) Condition 4.2 is realistic in practice. Under Conditions 4.1 and 4.2, it can be shown that there exists a state-dependent threshold type optimal clearing policy.

Theorem 4.2 *If Conditions* 4.1 *and* 4.2 *are satisfied, then there exists a state-dependent threshold optimal clearing policy denoted as* π^{τ^*}. *Specifically, for each underlying state* $i = 1, 2, \dots, M$, *there exists a non-empty sequence* $x_t^*(i)$ *such that*

$$x_t^*(i) = \arg\max_{x \in \Phi}\{H_t(x) : W_t(x, i) \le W_t(\emptyset, i) + K_t(x)\}, \tag{11}$$

and $\tau_{t,i}^* = H_t(x_t^*(i))$, *for all* $i = 1, 2, \dots, M$ and $t = 1, 2, \dots, N$.

We note that the sequence $x_t^*(i)$ is used only for computing $\tau_{t,i}^*$. Parameters of the optimal clearing policy, that is, $\{\tau_{t,1}^*, \dots, \tau_{t,M}^*\}$, are independent of the current state x. It is easy to see that, under Conditions 4.1 and 4.2, if $r_t^{\tau^*}(x, i) = 1$ and $x \le_d y$, we must have if $r_t^{\tau^*}(y, i) = 1$. This implies that, for any state x in the tree-structured state space, if the optimal decision is to clear all, then the optimal decision for any off-spring of x is to clear all.

Denote by e_k a vector with all elements being zero except for the kth element, which is one. The following property provides interesting insight into the structure of the optimal policy and is used in developing algorithms for computing the optimal policy.

Corollary 4.3 *Under Conditions* 4.1 *and* 4.2, *if* $r_t^{\tau^*}(x, i) = 1$, *we must have (i)* $r_t^{\tau^*}(x+e_k, i) = 1$; *(ii)* $r_t^{\tau^*}(x \frown q, i) = 1$, *for all* q, x, *and* i; *and (iii)* $r_t^{\tau^*}(x + e_j - e_k, i) = 1$, *for* $j < k$.

Proof By part (ii) of Condition 4.1, $H_t(x) \le H_t(x \frown q)$. By Theorem 4.2, (i) and ii) follow. Since $x \le_d x + e_j - e_k$, for $j < k$, by Condition 4.1, $H_t(x) \le H_t(x + e_j - e_k)$. By Theorem 4.2, (iii) follows. ∎

4.2 | Computing the optimal clearing policy

Due to the complexity of the state space Ω, the search for such an optimal policy is not straightforward and is challenging, especially for systems with a large state space. Fortunately, under Conditions 4.1 and 4.2, Theorems 4.1 and 4.2 imply that the optimal policy is of the on-off type and of the state-dependent threshold type. To find the optimal policy, our idea is to organize system states into a tree structure (see Figure 1), and explore as many tree branches as possible. At the same time, we find thresholds to stop the search process along individual branches, which avoids going too far down

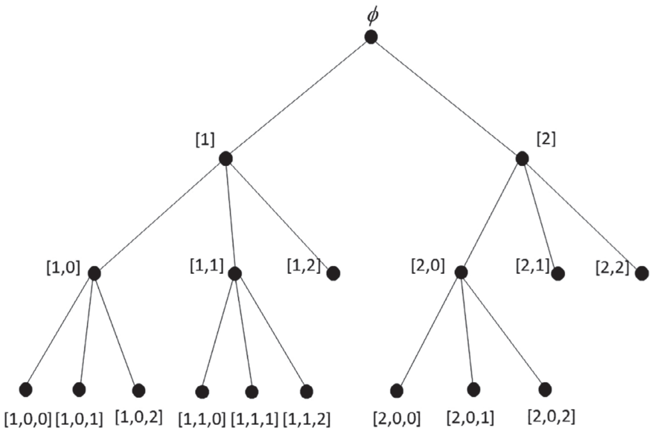

FIGURE 1 A sample tree of $\Psi_{(r_t)}$ with $Q = 2$

the tree. In this section, we apply the branch-and-bound idea to develop a search procedure in the tree-structured system to find the optimal policy. Due to Theorem 4.1, we now can limit the admissible policies to the set of on-off control policies.

First, we develop a method for the construction of the system state space. At the beginning of each period (before clearing), the system state is (x_t, i_t), for $t = 1, 2, \ldots$. That *pre-clearing system state process* (x_t, i_t) can be modeled as a GI/M/1-type Markov chain with a tree structure (see Figure 1). By definition, the receiving state is affected by the clearing rule r_{t-1} in the previous period. To determine which pre-clearing system contents can ever be attained in at least one underlying state in period t, we write

$$\Psi_{(r_{t-1})} = \bigcup_{q \in \{0, \ldots, Q\}} \left\{ x \frown q : x \in \Phi : \prod_{i=1}^{M} r_{t-1}(x, i) = 0 \right\}. \quad (12)$$

For computational purposes, we collect the underlying state space as $\Psi_{(r_{t-1})} \times \{1, 2, \ldots, M\}$, for $t = 1, 2, \ldots, N$. We thus group the system states with the same system contents, and use $\Psi_{(r_{t-1})}$ as the "level" set to identify the system content without needing to specify the underlying states.

For the clearing rule in Example 3.3.3, and with $Q = 2$ for the input process, the subtree corresponding to $\Psi_{(r_t)}$ is illustrated in Figure 1. Recall that the policy is of on-off type and clearing takes place if $|x| \geq 3$ or $L(x) = 3$ for state x. For leaf nodes {[1,0,0], [1,0,1], [1,1,0], [2,0,0]} in Figure 1, we have $L(x) = 3$; for leaf nodes {[1,1,1], [1,1,2], [2,0,2]}, $L(x) = 3$ and $|x| \geq 3$, and for {[1, 2], [2, 1], [2, 2]}, $|x| \geq 3$. For those states, the system state becomes $\emptyset$ right after the next clearing decision, and it becomes $\emptyset$, [1], or [2] at the end of the time period. Due to the on-off property of the clearing rule, the process x_t can traverse down at most one level in the tree, but can go back to the root node $\emptyset$ from all leaf nodes. That is why the process (x_t, i_t) is called a GI/M/1-type Markov chain.

Now we discuss how particular properties of clearing rules affect the structure and size of the system state space. First, let us call a clearing rule "logical" if it satisfies.

Condition 4.3 (Logical condition) At any decision epoch $t = 1, 2, \ldots, N$, and underlying state $i = 1, 2, \ldots, M$,

i. $r_t(\emptyset, i) = 0$ with certainty;
ii. for any $x, y \in \Phi$, if $r_t(x, i) = 1$, then $r_t(x \frown y, i) = 1$;
iii. for any $x \leq y \in \Phi$, if $r_t(x, i) = 1$, then $r_t(y, i) = 1$.

Proposition 4.4 *Under Conditions* 4.1 *and* 4.2, *the clearing rule* r_t *in period* t *satisfies Condition* 4.3, *and* $\Psi_{(r_t)}$ *can be mapped to a connected subtree of* Φ.

Proof That r_t is logical is an immediate consequence of Corollary 4.3. Next, we prove that the tree for $\Psi_{(r_t)}$ is actually a connected subtree for Φ. Part (i) of Condition 4.3 implies that $\emptyset$, the root node for Φ, can always be the root node for $\Psi_{(r_t)}$. Part (ii) of Condition 4.3 suggests that starting from $\emptyset$ and proceeding on any downward path of Φ, if a node x is not in $\Psi_{(r_t)}$, then no successor of x, that is, $x \frown y$, is in $\Psi_{(r_t)}$ either. In other words, there is a "cut-off" point on each downward path of Φ, and the subtree can be built by excluding the nodes beyond those cut-off points. Consequently, we have a subtree in which every node is reachable from, and can reach, state $\emptyset$. This completes the proof. ■

Theoretically, the system state space can be infinitely large if none of the clearing criteria can ever be attained. However, in practice, a clearing policy needs to be "feasible" so that the system is cleared in finite time. For any period t, we define the maximum cumulative quantity and the oldest input age allowed by clearing rule r_t, respectively, as follows:

$$\begin{aligned} \overline{Q}_{(r_t)} &= \max_{(x,i) \in \Omega} \{|x| \, : \, r_t(x, i) = 0\}; \\ \overline{L}_{(r_t)} &= \max_{(x,i) \in \Omega} \{L(x) \, : \, r_t(x, i) = 0\}. \end{aligned} \quad (13)$$

Here is the condition for a clearing rule to be feasible.

Condition 4.4 (Feasibility condition) For a given clearing rule r_t in period t, $\overline{Q}_{(r_t)}$ and $\overline{L}_{(r_t)}$ are finite.

Proposition 4.5 *Under Conditions* 4.1 *and* 4.2, *the clearing rule* r_t *in period t satisfies Condition* 4.4, *that is, the set* $\Psi_{(r_t)}$ *is finite, and the size of that set grows in the order of* $\mathcal{O}(Q^{\overline{L}_{(r_t)}})$.

Proof By Theorems 4.1 and 4.2, it is easy to see that Condition 4.4 is satisfied. If $\overline{L}_{(r_t)} < \infty$, then based on Equation (13), all sequences $x \in \Psi_{(r_t)}$ must have finite length. Each entry in a sequence is discrete and finite according to our assumptions of discrete and finite input quantities. Since we can make all sequences into equal length by attaching zeros to the left, the total number of such sequences grows in the order of $\mathcal{O}(Q^{\overline{L}_{(r_t)}})$. ■

For discrete time and discrete quantity models, the finite tree structure of $\Psi_{(r_t)}$ allows us to use the *breadth-first-traversal* (BFT) procedure from graph theory to construct and navigate the trees. The procedure for constructing $\Psi_{(r_t)}$ is described below.

BFT procedure: Constructing $\Psi_{(r_t)}$

Step 1 Initialize a list V, and store $\emptyset$ and numbers 1, 2, …, Q in V.

Step 2 Initialize a list U, and store numbers 1, 2, …, Q in U.

Step 3 If U is empty, go to Step 5; otherwise, read and delete the next entry from U according to the first-in-first-out (FIFO) rule and denote it as x.

Step 4 For any underlying state $i = 1, 2, \dots, M$, if $r_t(x, i) = 0$, enter $x \frown q$ for each $q = 0, 1, \dots, Q$ into both lists V and U; go back to Step 3.

Step 5 Output the sequences in V as $\Psi_{(r_t)}$.

Recall that the underlying process $\{i_t, t = 1, 2, \dots\}$ is an irreducible Markov chain, and the input process $\{(q_t, i_t), t = 1, 2, \dots\}$ is also Markovian. For an on-off control clearing rule r_t, we have $y_t = x_t \cdot (1 - r_t(x_t, i_t))$, $w_t = x_t - y_t = x_t \cdot r_t(x_t, i_t)$, and $x_{t+1} = x_t \cdot (1 - r_t(x_t, i_t)) \frown q_t$, which only depend on (x_t, i_t), q_t, and r_t. Recall that $\{(x_t, i_t), t = 1, 2, \dots\}$ is a discrete time Markov chain, whose transitions are time-dependent if π is not stationary. There are two types of transitions for that Markov chain, for all $x \in \Phi$, $i, j = 1, 2, \dots, M$ and $q = 0, 1, \dots, Q$:

i. $(x, i) \to (x \frown q, j)$: starting from system state (x, i), no clearing is required at the beginning of the current period, and a quantity q is received during the period;

ii. $(x, i) \to (q, j)$: starting from system state (x, i), the system is cleared at the beginning of the current period, and quantity q is received during the period.

The *one-step transition probabilities* between (x_t, i_t) and $(x_{t+1}, i_{t+1}) = (x', j)$ are determined by r_t and can be written as:

$$P^{(x,r_t)}_{(x,i),(x',j)} = \begin{cases} a^{(x,r_t)}_{(x,i),(x\frown q,j)} = (1 - r_t(x, i))[D_q]_{i,j}, & \text{if } x' = x \frown q; \\ b^{(x,r_t)}_{(x,i),(q,j)} = r_t(x, i)[D_q]_{i,j}, & \text{if } x' = q; \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $a^{(x,r_t)}_{(x,i),(x\frown q,j)}$ corresponds to the transition probabilities in period t without clearing, and $b^{(x,r_t)}_{(x,i),(q,j)}$ corresponds to the transition probabilities in period t with clearing.

We can collect the underlying states and express the transition probabilities in matrix form. More specifically, if $x' = x \frown q$, we have

$$P^{(x,r_t)}_{x,x'} = A^{(x,r_t)}_{x,x\frown q} = \begin{pmatrix} 1 - r_t(x, 1) & & \\ & \ddots & \\ & & 1 - r_t(x, M) \end{pmatrix} D_q; \tag{15}$$

while if $x' = q$, we have

$$P^{(x,r_t)}_{x,x'} = B^{(x,r_t)}_{x,q} = \begin{pmatrix} r_t(x, 1) & & \\ & \ddots & \\ & & r_t(x, M) \end{pmatrix} D_q; \tag{16}$$

otherwise, $P^{(x,r_t)}_{x,x'} = 0$. Together, we have $\sum_{q=0}^{Q} (A^{(x,r_t)}_{x,x\frown q} + B^{(x,r_t)}_{x,q}) = D$. Note that $P^{(x,r_t)}_{x,x'}$ are the block matrices in $P^{(x,r_t)}$, which is the transition probability matrix for the Markov chain $\{(x_t, i_t), t = 1, 2, \dots\}$.

With the transition probabilities obtained, the expected total cost function $C^{\pi}_{<n,N>}(x, i)$ defined in Equation (4) can be computed as

$$C^{\pi}_{<n,N>}(x, i) = u^{\pi}_n(x, i) + \sum_{(x',j)\in\Psi_{(r_n)}\times\{1,2,\dots,M\}} P^{(x,r_n)}_{(x,i),(x',j)} C^{\pi}_{<n+1,N>}(x', j), \tag{17}$$

where

$$u^{\pi}_t(x, i) = H_t((1 - r_t(x, i))x) + K_t(r_t(x, i)x), \tag{18}$$

for the on-off control function $r_t(x, i)$.

For a given policy π, the expected total cost $C^{\pi}_{<1,N>}(x, i)$ can be calculated by using a backward scheme that employs Equations (14), (17), and (18).

Finally, based on Theorem 4.1 and the tree structure of the system state space, Algorithm I utilizes the value iteration method for MDP to construct the optimal policy for cases satisfying Condition 4.1:

Algorithm 1. Finite horizon optimal policy algorithm

I.1 If $H_N(x) > \hat{k}_N$, let $r_N(x, i) = 1$; otherwise, let $r_N(x, i) = 0$. Use the BFT Procedure to create $\Psi_{(r_N)}$.

I.2 For each $(x, i) \in \Psi_{(r_N)} \times \{1, 2, \dots, M\}$, set $V^{N+1}(x, i) = K_{N+1}(x)$. Initialize counter $n = N$.

I.3 If $H_{n-1}(x) > \hat{k}_{n-1}$, let $r_{n-1}(x, i) = 1$; otherwise, let $r_{n-1}(x, i) = 0$. Use the BFT Procedure to create $\Psi_{(r_{n-1})}$.

I.4 For each $x \in \Psi_{(r_{n-1})}$, $j = 1, 2, \dots, M$, and $q = 0, 1, \dots, Q$, create $y = x \frown q$ and look up $V^{n+1}(y, j)$. If not found, set $V^{n+1}(y, j) = K_{n+1}(y) + V^{n+1}(\emptyset, j)$.

I.5 Compute $W_n(x, i)$ and $W_n(\emptyset, i)$ by Equation (7). If $W_n(x, i) > W_n(\emptyset, i) + K_n(x)$, record $V_n(x, i) = W_n(\emptyset, i) + K_n(x)$ and assign $r^*_n(x, i) = 1$; otherwise, record $V_n(x, i) = W_n(x, i)$ and assign $r^*_n(x, i) = 0$. Set $n := n - 1$.

I.6 If $n = 0$, stop and report $r^*_t(x, i)$ and $V_t(x, i)$, for all $(x, i) \in \Psi_{r^*_t} \times \{1, 2, \dots, M\}$ and $t = 1, 2, \dots, N$; otherwise, go back to Step I.3.

We note that, if Condition 4.2 also holds, we can output $\tau^*_t(i)$ instead of $r^*_t(x, i)$.

5 | OPTIMAL POLICIES OVER AN INFINITE HORIZON

We now consider the stochastic clearing problem over an *infinite* planning horizon, with the expected total discounted cost as our objective function. We assume that the discount factor $\alpha < 1$ and the cost functions and decision rule are time homogenous, that is, $H_t(x) = H(x)$, $K_t(x) = K(x)$, and $r_t(x, i) = r(x, i)$, for $t = 1, 2, \ldots$. The set of all admissible decision rules is denoted as R (ie, $R_t = R$, for $t = 1, 2, \ldots$). A clearing policy satisfying $r_t = r$, for all $t = 1, 2, \ldots$, is called a *stationary* policy. Since we consider only stationary policies, we shall call any admissible decision rule r a clearing policy. In this section, we characterize the optimal clearing policy.

As in Section 4, the expected total discounted cost over the infinite planning horizon can be found by

$$C^{r,\alpha}_{<1,\infty>}(x,i) = \mathbb{E}\left[\sum_{t=1}^{\infty} \alpha^{t-1}(H(y_t) + K(w_t)) | (x_1, i_1) = (x, i)\right]. \quad (19)$$

We assume that the optimal clearing policy r^* must satisfy $C^{r^*,\alpha}_{<1,\infty>}(x, i) \leq C^{r,\alpha}_{<1,\infty>}(x, i)$, for all $r \in R$. Existing theorems on dynamic programming and MDP allow us to generalize the results for the finite horizon problem to an infinite horizon by extending the length of the planning horizon $N \to \infty$ (Beyer et al., 2010; Iglehart, 1963; Puterman, 2005). Assume that the cost functions are stationary over time. We can drop the subscript t from Equation (5) and express the optimality equations for the infinite horizon problem as

$$C^{r,\alpha}_{<1,\infty>}(x,i) = \inf_{y \leq x}\left\{K(x-y) + H(y) + \alpha\mathbb{E}[C^{r,\alpha}_{<1,\infty>}(y \frown q, j) \mid i]\right\}. \quad (20)$$

Note that, abusing notation a bit, we use q and j as random variables for the input quantity per period and the state of the underlying Markov chain, respectively.

Condition 5.1, whose interpretation for the cost functions is similar to that of Condition 4.1, is required to establish our infinite-horizon results.

Condition 5.1 $0 = H(\emptyset) \leq H(x') \leq H(x)$, for all $x' \leq_d x \in \Phi$.

To prove the existence of an optimal clearing policy, we use successive approximations of the infinite horizon problem by "extending" the finite-horizon problem. First, let us examine the "first-n-period truncation" of the infinite horizon problem. Denote the expected total discounted cost for the truncated problem as

$$C^{r,\alpha}_{<1,n>}(x,i) = \mathbb{E}\left[\sum_{t=1}^{n} \alpha^{t-1} u^r(x_t, i_t)\right], \quad (21)$$

for all $(x, i) \in \Omega$. Next, define the value function for the truncated problem as $V^\infty_n(x, i) = \inf_{r \in R} C^{r,\alpha}_{<1,n>}(x, i)$. Naturally, the value function for this n-stage finite horizon problem exists, and can be computed recursively as $V^\infty_0(x, i) = 0$, and

$$V^\infty_n(x,i) = \min_{y \leq x}\left\{K(x-y) + H(y) + \alpha\mathbb{E}[V^\infty_{n-1}(y \frown q, j) \mid i]\right\}. \quad (22)$$

Now we define the value function of the infinite horizon problem as $V(x, i) = \lim_{n\to\infty} \inf_{r \in R} C^{r,\alpha}_{<1,n>}(x, i)$. Since $\inf_{r\in R} C^{r,\alpha}_{<1,n>}(x, i) = V^\infty_n(x, i)$, if we can prove that $V^\infty_n(x, i)$ converges as well, then we know that the value function for the infinite horizon problem exists, for which we find

$$V(x,i) = \lim_{n\to\infty} V^\infty_n(x,i) = \lim_{n\to\infty} C^{r^*,\alpha}_{<1,n>}(x,i) = C^{r^*,\alpha}_{<1,\infty>}(x,i). \quad (23)$$

Under Condition 5.1, it can be shown that $V^\infty_n(x, i) \leq V^\infty_{n+1}(x, i)$, for all $(x, i) \in \Omega$ and $n = 0, 1, \ldots$.

Next, we shall establish an upper bound on $V^\infty_n(x, i)$. One such possible bound is for the case that the maximum input quantity is received in each period, and the system is also cleared that often. A clearing cost of $\hat{k}$ is thus incurred every period; no delay penalty is ever charged, since inputs are cleared at the earliest possible time. For this case, if $0 \leq \alpha < 1$, it is true that

$$\lim_{n\to\infty} \sup_{r\in R} C^{r,\alpha}_{<1,n>}(x,i) \leq \hat{k} + \frac{\alpha\hat{k}}{1-\alpha} = \frac{\hat{k}}{1-\alpha} < \infty.$$

With the established upper bound, the results obtained in Section 4.1 for the finite horizon problem continue to hold for the infinite horizon problem. We summarize them in the following theorems.

Theorem 5.1 *Under Condition* 5.1, $\lim_{n\to\infty} V^\infty_n(x, i)$ *exists, and* $V(x, i) = \lim_{n\to\infty} V^\infty_n(x, i)$ *is a solution to the optimality Equations* (20), *for all* $(x, i) \in \Omega$.

Theorem 5.2 *Under Condition* 5.1, *there exists a state-dependent threshold optimal clearing policy denoted by a nonnegative vector* τ^*. *More specifically, there exists a non-empty sequence* $x^*(i)$ *such that*

$$x^*(i) = \arg\max_{x\in\Phi}\{x : W(x,i) \leq K(x) + W(\emptyset, i)\};$$
$$\tau^*(i) = H(x^*(i)), \quad (24)$$

for all $i = 1, 2, \ldots, M$. *If* $r^*(x \frown q, i) = 0$, *then* $r^*(x, i) = 0$, *for all* x, q, *and* i.

The infinite horizon stochastic clearing problem can be solved as an infinite horizon discounted MDP. We can use value iteration or policy iteration to calculate its value function and optimal policy parameters. Similar to the finite horizon case, an algorithm can be developed for computing the optimal clearing policy. Details are omitted.

6 | NUMERICAL EXAMPLES

In this section, we analyze three examples to demonstrate the added value of the stochastic clearing model, especially the time dependent cost structure, introduced in this paper. In Example 6.1, we compare models with linear and nonlinear delay penalty cost functions. The observation is that a model with a nonlinear structure may not be approximated well by models with a linear cost structure. In Example 6.2, we compare the optimal policy to conventional hybrid policies. The observation is that the (best) conventional hybrid (time-and-quantity) policies may perform poorly. In Example 6.3, we consider a case with an infinite planning horizon and an input process with three states. The example demonstrates that the optimal actions and the corresponding expected total cost can be very sensitive to the state of the input process.

Example 6.1 We consider a clearing system with a finite planning horizon $N = 10$: (i) The input process is a compound renewal process with $M = 1$, $Q = 5$, $D_0 = 0.1$, $D_1 = 0.2$, $D_2 = 0.3$, $D_3 = 0.1$, $D_4 = 0.2$, and $D_5 = 0.1$; (ii) the delay-penalty cost function is $H_t(y_t) = \mu \sum_{j=1}^{l} j^2 y_{[j]}$ with $\mu = 1.5$; and (iii) the fixed clearing cost function is $K_t(w_t) = 10\delta_{\{|w_t|>0\}}$.

The delay-penalty costs per unit time of each individual input are increasing in time, which captures the urgency to clear those inputs that have been in the system for a longer interval. We use this example to demonstrate the necessity to use our model to find the optimal clearing policy. We conduct the analysis in three steps.

(a) First, we find the optimal clearing policy and the minimum expected total costs. Since Conditions 4.1 and 4.2 are satisfied, the optimal clearing policy π is of the on-off type. The optimal clearing rule and its corresponding expected cost for each initial state x at the beginning of period 1 are presented in Table 2. Note that for any state x not listed in Table 2, the optimal clearing rule is 1 (ie, to clear immediately), with an expected cost $V_1^{(1.5)}(x) = 76.6459$. The optimal actions for stages $n = 2, 3, \ldots, N = 10$ can be found similarly by setting the planning horizon N to be 9, 8, …, 2, and 1. Details are omitted.

(b) Now, we redefine the delay-penalty cost function as $\widehat{H}_t(y_t) = \mu \sum_{j=1}^{l} j y_{[j]}$. This means that the delay penalty cost is charged at a constant rate (ie, a linear function in the delay time of each input), which is the conventional accounting scheme for delay penalty costs. We denote by $\widehat{\pi}^{(\mu)}$ the optimal clearing policy. For $\mu = 1.5$, we find the optimal clearing policy and in turn the expected total cost $\widehat{V}_1^{(\mu)}(x)$. Results are shown in Table 3. Note that for all other states x (absent from this table), the optimal clearing rule is to clear immediately and the corresponding expected total cost is 75.7539.

(c) Now, we apply the clearing policy $\widehat{\pi}^{(\mu)}$ to the original model and similarly find its expected total cost $\widetilde{V}_1^{(\mu)}(x)$. Since $\widehat{\pi}^{(\mu)}$ may not be optimal in the original model, we must have $\widetilde{V}_1^{(\mu)}(x) \geq V_1^{(\mu)}(x)$ for all x and μ. The difference between the $\widetilde{V}_1^{(\mu)}(x)$ and $V_1^{(\mu)}(x)$ measures the modeling error caused by using the linear accounting scheme for the delay penalty cost. For a fair comparison, we let μ go from 0.5 to 4.5 and compute the respective $V_1^{(\mu)}(x)$ and $\widetilde{V}_1^{(\mu)}(x)$. For example, we show the results for $\widetilde{V}_1^{(\mu)}(\emptyset) - V_1^{(\mu)}(\emptyset)$ in Table 4. When μ decreases, the delay penalty cost decreases. On the other hand, the difference between $V_1^{(\mu)}(\emptyset)$ and $\widetilde{V}_1^{(\mu)}(\emptyset)$ tends to be larger. Thus, it is more important to use our model for such cases. This example demonstrates that the performance of the clearing policy $\widehat{\pi}^{(\mu)}$ found by using the $\widehat{H}_t(x)$ can be significantly worse than the actual optimal clearing policy π. This example also justifies the added complexity in the cost functions, and the usefulness of the methods and algorithms developed in this paper.

Example 6.2 We consider a general clearing system with $N = 8$. The input process is a Markov modulated arrival process given as

$$D_0 = \begin{pmatrix} 0.09 & 0.01 \\ 0.06 & 0.54 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 0.27 & 0.03 \\ 0.03 & 0.27 \end{pmatrix},$$

$$D_2 = \begin{pmatrix} 0.54 & 0.06 \\ 0.01 & 0.09 \end{pmatrix}.$$

The delay penalty cost is given as $H_t(y_t) = 0.1 \sum_{j=1}^{l} (j \cdot y_{[j]})^2$ with $l = |y_t|$. The fixed clearing cost is $K_t(w_t) = (2 + 0.5\max\{0, \ 4 - t\}) \cdot \delta_{\{|w_t|>0\}}$.

TABLE 2 Summary of optimal clearing rule in period 1 for Example 6.1 (original model)

x	∅	[1]	[2]	[3]	[4]	[5]	[1,0]	[1,1]
$r_1^*(x)$	0	0	0	0	0	1	0	1
$V_1^{(1.5)}(x)$	56.8383	61.9062	63.4939	64.9939	66.4939	66.8383	66.4939	66.8383

TABLE 3 Summary of optimal clearing rule in period 1 for Example 6.1 (modified model)

x	∅	[1]	[2]	[3]	[4]	[1,0]	[1,1]	[1,2]	[2,0]
$\widehat{r}_1^*(x)$	0	0	0	0	0	0	0	0	0
$\widehat{V}_1^{(1.5)}(x)$	55.8954	60.0074	62.3867	63.9369	65.4369	62.1320	63.9369	65.4369	65.4369

TABLE 4 $\widetilde{V}_1^{(\mu)}(\emptyset) - V_1(\emptyset)$ for ¯ = 0.5,1.0, … ,4.0, and 4.5 for Example 6.1

μ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
$V_1^{(\mu)}(\emptyset)$	42.5899	50.7137	56.8383	61.4904	64.8961	67.5316	70.0477	72.4172	73.7083
$\widetilde{V}_1^{(\mu)}(\emptyset)$	46.8378	52.7534	58.4119	62.3613	65.3802	68.1498	70.6092	73.1517	73.8417
$\widetilde{V}_1^{(\mu)}(\emptyset) - V_1^{(\mu)}(\emptyset)$	4.2479	2.0397	1.5736	0.8709	0.4841	0.6182	0.5615	0.7345	0.1334

TABLE 5 Summary of optimal clearing rule and costs in period 1 for Example 6.2

x	$r_1^*(x, 1)$	$r_1^*(x, 2)$	$V_1(x, 1)$	$V_1(x, 2)$
Ø	0	0	8.1006	5.7630
[1]	0	0	8.9449	7.4782
[2]	0	0	10.1333	8.5407
[1, 0]	0	0	9.7820	8.3735
[1, 1]	0	0	10.1680	8.6207
[1, 2]	0	0	10.5333	8.9407
[2, 0]	0	1	11.3333	9.2630
[2, 1]	0	1	11.4333	9.2630
[2, 2]	1	1	11.6006	9.2630
[1,0,0]	0	0	10.6333	9.0407
[1,0,1]	0	0	10.7333	9.1407
[1,0,2]	0	1	11.0333	9.2630
[1,1,0]	0	1	11.0333	9.2630
[1,1,1]	0	1	11.1333	9.2630
[1,1,2]	0	1	11.4333	9.2630
All other states	1	1	11.6006	9.2630

Conditions 4.1 and 4.2 are satisfied, and the optimal clearing policy is of the on-off type. The optimal clearing policy and the related value functions for period 1 are summarized in Table 5. Note that for all other states x, the optimal clearing rule is to clear immediately, and $V_1(x, 1) = 11.6006$ and $V_1(x, 2) = 9.2630$. As shown in Table 5, the optimal action depends on not only the outstanding input x, but also the state of the underlying process of the inputs. The resulting expected total costs can vary significantly for different process states. This justifies the need to use the Markovian arrival process for the input process.

Next, we run the same model with the hybrid policy (see Example 3.3.3) with parameters (w, T) and find the expected total cost $\widetilde{V}^{(w,T)}(x, i)$ for initial state (x, i). Apparently, $\widetilde{V}^{(w,T)}(x, i)$ is greater than $V_1(x, i)$ for all (x, i) and any valid (w, T). For example, $\widetilde{V}^{(w,T)}(\emptyset, 1)$ is greater than $V_1(\emptyset, 1) = 6.3309$ for any valid (w, T). We present the difference $\widetilde{V}^{(w,T)}(\emptyset, 1) - V_1(\emptyset, 1)$ in Table 6. The results show that properly chosen (w, T) can perform very close to the optimal policy for a specific x. However, it might be challenging to find a proper hybrid policy (w, T) for *all* states x. Our extended numerical experiments demonstrate that even the best hybrid policy may perform poorly. Thus, finding the (nonhybrid type) optimal policy becomes an important practical issue for stochastic clearing systems.

Example 6.2 has demonstrated that the optimal action depends on the state of the underlying Markov chain of the input process. In the next example, we study a case with an infinite planning horizon and an input process with three states.

Example 6.3 We consider a clearing system with $N = \infty$, the discount factor $\alpha = 0.95$, the delay penalty cost is $H_t(y_t) = 0.5 \sum_{j=1}^{l} j(y_{[j]})^2$, the fixed clearing cost is $K(w_t) = 10 \cdot \delta_{\{|w_t|>0\}}$, and the input process is,

$$D_0 = \begin{pmatrix} 0.1 & 0.0 & 0.2 \\ 0.0 & 0.2 & 0.1 \\ 0.1 & 0.0 & 0.7 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix},$$

$$D_2 = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.6 & 0.0 \\ 0.0 & 0.1 & 0.0 \end{pmatrix},$$

$$D_3 = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}, \quad D_4 = \begin{pmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 \end{pmatrix},$$

$$D_5 = \begin{pmatrix} 0.7 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

The optimal policy and the corresponding expected total costs are shown in Table 7. Note that for all other states x, we have $V(x, 1) = 103.2129$ and $V(x, 2) = 98.5976$, $V(x, 3) = 92.7427$, and the optimal clearing rule is to clear immediately. It is seen that the optimal policy depends on the state $i = 1,2,3$, and the corresponding expected total costs are significantly different. We would like to point out that, for this example, the system state space is very large and the computation time is significantly longer than that of Examples 6.1 and 6.2.

The state space has $n = 2850$ states to search for the optimal clearing policy. The maximum quantity in each state is about 10 (ie, $|x| \leq 10$). The time required to search for the optimal policy is much longer than that for smaller cases, which indicates that the search time increases significantly as the state space increases. In fact, the state space would have been much larger, if we had chosen parameters in $H_t(x)$ and $K_t(x)$ differently.

The optimal policy given in Table 7 is identical to the optimal policies for the finite horizon cases with $N \geq 6$. This example demonstrates that results for the infinite horizon case could be used as approximations to the finite horizon cases, if the planning horizon N is sufficiently large and the discount rate α is close to one.

Now, we enforce capacity constraints on the model and assume that the clearing limit is $W = 4$, that is, as soon as the total input quantity $|x|$ reaches W, a clearing takes place

TABLE 6 $\widetilde{V}_1^{(w,T)}(\emptyset, 1) - V_1(\emptyset, 1)$ for Example 6.2

w/T	2	3	4	5	6	7	8
1	5.7050	5.7050	5.7050	5.7050	5.7050	5.7050	5.7050
2	5.7050	2.9330	2.7396	2.7506	2.7808	2.8033	2.8099
3	5.7050	0.4763	0.3870	0.5488	0.6327	0.6925	0.7073
4	5.7050	0.4763	0.3661	0.9154	1.1161	1.2142	1.2392
5	5.7050	0.4763	0.7595	2.2187	2.5655	2.6874	2.7199
6	5.7050	0.4763	0.7595	3.5544	4.2084	4.3968	4.4337

TABLE 7 Summary of optimal clearing rules and costs for Example 6.3

x	$r^*(x, 1)$	$r^*(x, 2)$	$r^*(x, 3)$	$V(x, 1)$	$V(x, 2)$	$V(x, 3)$
∅	0	0	0	93.2129	88.5976	82.7427
[1]	0	0	0	95.3336	91.2785	88.3316
[2]	0	0	0	97.7533	95.3509	92.3415
[3]	0	0	1	100.5629	98.0499	92.7427
[4]	1	1	1	103.2129	98.5976	92.7427
[5]	1	1	1	103.2129	98.5976	92.7427
[1, 0]	0	0	0	96.1493	92.4478	89.8045
[1, 1]	0	0	0	97.0858	94.1799	91.6220
[1, 2]	0	0	1	98.9062	96.5499	92.7427
[1, 4]	0	1	1	101.5629	98.5976	92.7427
[1, 5]	1	1	1	103.2129	98.5976	92.7427
[2, 0]	1	1	1	103.2129	98.5976	92.7427
[2, 1]	0	0	1	99.9537	97.5499	92.7427
[2, 2]	0	0	1	100.5487	98.0499	92.7427
[2, 3]	0	1	1	102.0629	98.5976	92.7427
All other states	1	1	1	103.2129	98.5976	92.7427

TABLE 8 Summary of optimal clearing rules and costs for Example 6.3: with capacity limit $W = 4$

x	$r^*(x, 1)$	$r^*(x, 2)$	$r^*(x, 3)$	$V(x, 1)$	$V(x, 2)$	$V(x, 3)$
∅	0	0	0	93.8133	90.1047	83.5010
[1]	0	0	0	97.1633	95.3244	91.9127
[2]	0	0	0	98.6633	96.8244	93.4127
All other states	1	1	1	103.8133	100.1047	93.5010

immediately. The state space for searching for the optimal clearing policy reduces to $n = 797$. The results are given in Table 8, and they indicate that the costs can be substantially higher, which is caused by the imposed capacity constraints.

7 | CONCLUSIONS AND DISCUSSION

This paper studied a complex stochastic clearing system. The main contributions of our research are (a) modeling of the delay penalty cost as an increasing function of both the quantities and delays of individual inputs; (b) identifying the on-off or threshold structure of the optimal policies; (c) developing efficient algorithms for computing the optimal policies; and (d) gaining insights on the advantages of the new optimal policies over the conventional hybrid policies. The optimal policies found in this paper are state-dependent threshold policies.

To extend the potential applications of our work, research can be done in several directions, including systems with (clearing) capacity constraints and/or variable (vs fixed) clearing costs. For theoretical development, one may consider continuous time and/or continuous quantities stochastic clearing systems.

ACKNOWLEDGMENTS

We would like to thank three reviewers and the associate editor for their constructive suggestions and insightful comments that helped us improve the quality of the paper significantly. We would also like to thank NSERC for financial support to this research through two discovery grants.

ORCID

Qi-Ming He https://orcid.org/0000-0003-2381-3242

REFERENCES

Asmussen, S., & Koole, G. (1993). Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, *30*, 365–372.

Beyer, D., Cheng, F., Sethi, S. P., & Taksar, M. (2010). Markov demand inventory models. New York, NY: Springer US.

Bookbinder, J. H., Cai, Q., & He, Q.-M. (2011). Shipment consolidation by private carrier: The discrete time and discrete quantity case. *Stochastic Models*, *27*(4), 664–686.

Bookbinder, J. H., & Higginson, J. K. (2002). Probabilistic modeling of freight consolidation by private carriage. *Transportation Research, Part E*, *38*(5), 305–318.

Boxma, O. J., Perry, D., & Stadje, W. (2001). Clearing models for M/G/1 queues. *Queueing Systems*, *38*(3), 287–306.

Cai, Q., He, Q.-M., & Bookbinder, J. H. (2014). A tree-structured Markovian model of the shipment consolidation process. *Stochastic Models*, *30*(4), 521–553.

Çetinkaya, S., Tekin, E., & Lee, C. Y. (2008). A stochastic model for joint inventory and outbound shipment decisions. *IIE Transactions*, *40*(3), 324–340.

Chen, F., & Song, J. S. (2001). Optimal policies for multiechelon inventory problems with Markov-modulated demand. *Operations Research*, *49*(2), 226–234.

Dudin, A. N., & Karolik, A. V. (2001). BMAP/SM/1 queue with Markovian input of disasters and non-instantaneous recovery. *Performance Evaluation*, *45*(1), 19–32.

Higginson, J. K., & Bookbinder, J. H. (1994). Policy recommendations for a shipment consolidation program. *Journal of Business Logistics*, *15*(1), 87–112.

Higginson, J. K., & Bookbinder, J. H. (1995). Markovian decision processes in shipment consolidation. *Transportation Science*, *29*(3), 242–255.

Iglehart, D. L. (1963). Optimality of (*s*, *S*) policies in the infinite horizon dynamic inventory problem. *Management Science*, *9*(2), 259–267.

Inoue, Y., & Takine, T. (2014). The FIFO single-server queue with disasters and multiple Markovian arrival streams. *Journal of Industrial and Management Optimization*, *10*(1), 57–87.

Kim, K., & Seila, A. F. (1993). A generalized cost model for stochastic clearing systems. *Computers & Operations Research*, *20*(1), 67–82.

Li, Q., & Yu, P. (2013). Multimodularity and its applications in three stochastic dynamic inventory problems. *Manufacturing and Service Operations Management*, *16*(3), 455–463.

Li, Q., Yu, P., & Wu, X. (2016). Managing perishable inventories in retailing: Replenishment clearance sales, and segregation. *Operations Research*, *64*(6), 1270–1284.

Mutlu, F., Çetinkaya, S., & Bookbinder, J. H. (2010). An analytical model for computing the optimal time-and-quantity-based policy for consolidated shipments. *IIE Transactions*, *42*(5), 367–377.

Neuts, M. F. (1979). A versatile Markovian point process. *Journal of Applied Probability*, *16*, 764–779.

Puterman, M. L. (2005). Markov decision processes: Discrete stochastic dynamic programming. Hoboken, NJ: John Wiley and Sons.

Scarf, H. (1960). *The optimality of* (S, s) *policies in the dynamic inventory problem*. In Mathematical methods in the social sciences. Stanford, CA: Stanford University Press.

Song, J. S., & Zipkin, P. (1993). Inventory control in a fluctuating demand environment. *Operations Research*, *41*(2), 351–370.

Stidham, S. (1974). Stochastic clearing systems. *Stochastic Processes and their Applications*, *2*(1), 85–113.

Stidham, S. (1977). Cost models for stochastic clearing systems. *Operations Research*, *25*(1), 100–127.

Stidham, S. (1986). Clearing systems and (*s*, *S*) inventory systems with nonlinear costs and positive lead times. *Operations Research*, *34*(2), 276–280.

Veinott, A. F., & Wagner, H. M. (1965). Computing optimal (*s*, *S*) inventory policies. *Management Science*, *11*(5), 525–552.

Xia, L., He, Q.-M., & Alfa, A. S. (2017). Optimal control of state-dependent service rates in a MAP/M/1 queue. *IEEE Transactions on Automatic Control*, *62*(10), 4965–4979.

Zheng, Y. S. (1991). A simple proof for optimality of (*s*, *S*) policies in infinite-horizon inventory systems. *Journal of Applied Probability*, *28*(4), 802–810.

How to cite this article: He Q-M, Bookbinder JH, Cai Q. Optimal policies for stochastic clearing systems with time-dependent delay penalties. *Naval Research Logistics* 2020;67:487–502. https://doi.org/10.1002/nav.21931

APPENDIXA

The proofs of Theorem 4.1, 4.2, 5.1, and 5.2 are collected in this Appendix.

Proof of Theorem 4.1 The theorem can be proved by characterizing the functions $V_t(x, i)$ and $W_t(x, i)$. We show that, under Condition 4.1, $x' \leq_d x$ implies

$$V_t(x', i) \leq V_t(x, i) \leq V_t(x', i) + \hat{k}_t; \tag{A1}$$

$$\begin{aligned} 0 \leq H_t(x) - H_t(x') &\leq W_t(x, i) - W_t(x', i) \\ &\leq H_t(x) - H_t(x') + \hat{k}_{t+1}, \end{aligned} \tag{A2}$$

for all $x, x' \in \Phi$, $i = 1, 2, \ldots, M$ and $t = 1, 2, \ldots, N$.

We first observe that $x' \leq_d x$ implies that $H_t(x') \leq H_t(x)$ according to Condition 4.1 (ii), for $t = 1, 2, \ldots, N$. We then proceed with a proof by induction on t. Note that for any $i = 1, 2, \ldots, M$, we have $H_N(x) - H_N(x') \geq 0$ and $V_{N+1}(x, i) - V_{N+1}(x', i) = K_{N+1}(x) - K_{N+1}(x') \geq 0$. Since all inputs have to be cleared by period $N+1$, we have

$$\begin{aligned} W_N(x, i) - W_N(x', i) &= H_N(x) - H_N(x') \\ &+ \mathbb{E}\left[V_{N+1}(x \frown q_N, i_{N+1}) \right.\\ &\left. - V_{N+1}\left(x' \frown q_N, i_{N+1}\right) | i_N = i\right]. \end{aligned}$$

Then $W_N(x, i) - W_N(x', i) \geq 0$. Noting that $\emptyset \leq_d x$ for any state x, we have $W_N(x, i) - W_N(\emptyset, i) \geq 0$.

From Equation (8), we have

$$\begin{aligned}V_N(x',i) &= \inf_{y'\le x'}\left\{K_N(x'-y')+W_N\left(y',i\right)\right\}\\ &= \min\left\{\inf_{y'\le x',y'\ne x'}\{\hat{k}_N+W_N(y',i)\},W_N(x',i)\right\}\\ &= \min\left\{\hat{k}_N+W_N(\emptyset,i),\ \ W_N\left(x',i\right)\right\}\\ &\le \min\left\{K_N(x)+W_N(\emptyset,i),\ \ W_N(x,i)\right\}=V_N(x,i).\end{aligned}$$

Now suppose that Equations (A1) and (A2) both hold for periods $t+1, t+2, \dots,$ and N. We see that, for period t,

$$\begin{aligned}&\mathbb{E}\left[V_{t+1}(x\frown q_t,i_{t+1})-V_{t+1}(x'\frown q_t,i_{t+1})|i_t=i\right]\\ &\quad=\sum_{j=1}^{M}\sum_{q=0}^{Q}\left[V_{t+1}(x\frown q,j)-V_{t+1}(x'\frown q,j)\right][D_q]_{i,j}\ge 0,\end{aligned}$$

since $x'\le_d x$ implies $x'\frown q\le_d x\frown q$. By Equation (7) and Condition 4.1, we obtain $W_t(x,\ i)-W_t(x',i)\ge 0$, which leads to $W_t(\emptyset,i)=\inf_{x\in\Phi}W_t(x,\,i)$. Then we have

$$\begin{aligned}V_t(x',i) &= \inf_{y'\le x'}\left\{K_t(x'-y')+W_t\left(y',i\right)\right\}\\ &= \min\left\{\inf_{y'\le x',y'\ne x'}\{\hat{k}_t+W_t(y',i)\},W_t(x',i)\right\}\\ &= \min\left\{K_t(x')+W_t\left(\emptyset,i\right),\ \ W_t\left(x',i\right)\right\}\\ &\le \min\left\{K_t(x)+W_t(\emptyset,i),\ \ W_t(x,i)\right\}=V_t(x,i).\quad\text{(A3)}\end{aligned}$$

We also have

$$\begin{aligned}V_t(x,i) &= \min\{K_t(x)+W_t(\emptyset,i),\ \ W_t(x,i)\}\\ &\le W_t(\emptyset,i)+\hat{k}_t\\ &= \min\left\{W_t(\emptyset,i),\ \ W_t\left(\emptyset,i\right)\right\}+\hat{k}_t\\ &\le \min\left\{K_t(x')+W_t\left(\emptyset,i\right),\ \ W_t\left(x',i\right)\right\}+\hat{k}_t\\ &= V_t(x',i)+\hat{k}_t,\end{aligned}$$

for all $i=1,2,\dots,M$ and $t=1,2,\dots,N$, which proves Equation (A1).

Next, we prove Equation (A2) by induction on t. Recall that $V_{N+1}(x,\ i)-V_{N+1}(x',i)\ge 0$, and so

$$\begin{aligned}&W_N(x,i)-W_N(x',i)=H_N(x)-H_N(x')\\ &\quad+\mathbb{E}\left[V_{N+1}(x\frown q_N,i_{N+1})-V_{N+1}\left(x'\frown q_N,i_{N+1}\right)|i_N=i\right]\\ &\le H_N(x')-H_N(x)+\mathbb{E}\left[\hat{k}_{N+1}\delta_{|x\frown q_N|>0}|i_N=i\right]\\ &\le H_N(x')-H_N(x)+\hat{k}_{N+1}.\end{aligned}$$

Thus, inequality (A2) holds for $t=N$. Now, suppose that Equation (A2) holds for all $t+1$, $t+2, \dots, N$; we need to show that it holds for t. By the definition of $W_t(x,\,i)$, we have

$$\begin{aligned}&W_t(x,i)-W_t(x',i)=H_t(x)-H_t(x')\\ &\quad+\mathbb{E}\left[V_{t+1}(x\frown q_t,i_{t+1})-V_{t+1}\left(x'\frown q_t,i_{t+1}\right)|i_t=i\right]\\ &=H_t(x)-H_t(x')\\ &\quad+\sum_{j=1}^{M}\sum_{q=0}^{Q}\left[V_{t+1}(x\frown q,j)-V_{t+1}(x'\frown q,j)\right][D_q]_{i,j}.\end{aligned}$$

By the induction hypothesis and $x'\frown q\le_d x\frown q$, the two inequalities in Equation (A2) are obtained.

In the above proof of Equations (A1) and (A2), Theorem 4.1 has been proved as a by-product, as indicated in Equation (A3). ■

Proof of Theorem 4.2 Using the same induction proof as for Theorem 4.1, we can easily verify that, under Conditions 4.1 and 4.2, if $H_t(x)\le H_t(y)$ for some $x,y\in\Phi$ and $t=1,2,\dots,N$, then we must have, for all $i,j=1,2,\dots,M$,

i. $V_t(x,i)\le V_t(y,i)\le V_t(x,i)+\hat{k}_t$, and
ii. $0\le H_t(y)-H_t(x)\le W_t(y,i)-W_t(x,i)\le H_t(y)-H_t(x)+\hat{k}_{t+1}$.

The above results imply that, rather than having to check the actual system state against the sets of states with an "on" decision, optimal clearing decisions can be based on the delay penalty cost H_t. This leads to an optimal state-dependent *threshold clearing policy*, which means that for any state x in the tree-structured state space, if the optimal decision is to clear all, then the optimal decision for any off-spring of x is to clear all. First, it is easy to see that Equation (11) implies that $x_t^*(i)$ is the sequence with the highest delay penalty H_t among all sequences for which it is better to continue to accumulate in period t, for underlying state i. Then in any period $t=1,2,\dots,N$ and underlying state i, for any non-empty system content y such that $H_t(y)>H_t(x_t^*(i))$, the above (i) and (ii) imply that $W_t(y,i)>W_t(x_t^*(i),i)$. Consequently, Equation (11) shows that $W_t(y,i)>W_t(\emptyset,i)+K_t(y)$, which means clearing is preferred. On the other hand, for any non-empty system state $(z,\,i)$ such that $H_t(z)\le H_t(x_t^*(i))$, we have $W_t(z,i)\le W_t(x_t^*(i),i)$. This leads to $W_t(z,\ i)\le W_t(\emptyset,i)+K_t(z)$, which indicates that clearing is not necessary. Therefore, $\tau_{t,i}^*=H_t(x_t^*(i),i)$ is the threshold for the threshold policy. Theorem 4.2 is proved. ■

Proof of Theorem 5.1 We first show that the limit exists. The existence of a upper bound for V_n^∞ suggests that the sequence of functions V_n^∞ is point-wise non-decreasing in n and bounded from above, that is,

$$0=V_0^\infty\le V_1^\infty\le\cdots\le V_n^\infty\le\frac{\hat{k}}{1-\alpha}.$$

Therefore, by the monotone convergence theorem, V_n^∞ converges point-wisely to V. We now show that this limit is a solution to Equation (20). First we have

$$\begin{aligned} V_n^\infty(x,i) &\le V_{n+1}^\infty(x,i) \\ &= \min_{y\le x}\{K(x-y)+H(y)+\alpha\mathbb{E}[V_n^\infty(y\frown q,j)\mid i]\}. \end{aligned} \tag{A4}$$

Thus, as $n\to\infty$, both sides of the above inequality converge, so that

$$V(x,i)\le \min_{y\le x}\{K(x-y)+H(y)+\alpha\mathbb{E}[V(y\frown q,j)\mid i]\}.$$

On the other hand,

$$\begin{aligned} V_n^\infty(x,i) &\ge V_{n-1}^\infty(x,i) \\ &= \min_{y\le x}\{K(x-y)+H(y)+\alpha\mathbb{E}[V_{n-2}^\infty(y\frown q,j)\mid i]\}. \end{aligned}$$

Passing to the limit in the preceding inequality yields

$$V(x,i)\ge \min_{y\le x}\{K(x-y)+H(y)+\alpha\mathbb{E}[V(y\frown q,j)\mid i]\}. \tag{A5}$$

Combining Equations (A4) and (A5), Theorem 5.1 follows. ■

Proof of Theorem 5.2 Under Condition 5.1, if $x'\le_d x$, then

$$\begin{aligned} &V(x',i)\le V(x,i)\le V(x',i)+\hat{k}; \\ &V(x,i)=\min\{K(x)+W(\emptyset,i),W(x,i)\}, \end{aligned} \tag{A6}$$

where $W(y,i)=H(y)+\alpha\mathbb{E}[V(y\frown q,j)\mid i]$, for all $i=1,2,\ldots,M$ (note: q is a generic random variable for the input in a period). Equation (A6) implies that under Condition 5.1, the optimal clearing policy exists and must be an on-off control clearing policy. For the infinite case, we have $H_t(x)=H_{t+1}(x)=H(x)$. Applying the limit argument, we can further show that, under Condition 5.1, the optimal policy is of the threshold type and the threshold policy is given by Equation (24). This leads to Theorem 5.2. ■