SIAM J. DISCRETE MATH.
Vol. 28, No. 4, pp. 1725–1796

FULLY POLYNOMIAL TIME APPROXIMATION SCHEMES FOR STOCHASTIC DYNAMIC PROGRAMS*

NIR HALMAN†, DIEGO KLABJAN‡, CHUNG-LUN LI§, JAMES ORLIN¶, AND DAVID SIMCHI-LEVI¶

Abstract. We present a framework for obtaining fully polynomial time approximation schemes (FPTASs) for stochastic univariate dynamic programs with either convex or monotone single-period cost functions. This framework is developed through the establishment of two sets of computational rules, namely, the calculus of K-approximation functions and the calculus of K-approximation sets. Using our framework, we provide the first FPTASs for several NP-hard problems in various fields of research such as knapsack models, logistics, operations management, economics, and mathematical finance. Extensions of our framework via the use of the newly established computational rules are also discussed.

Key words. fully polynomial time approximation schemes, stochastic dynamic programming, K-approximation

AMS subject classifications. 68Q25, 68W25, 90B05, 90B06, 90C15, 90C39, 90C40, 90C56, 90C59

DOI. 10.1137/130925153

1. Introduction.

Dynamic programming. Dynamic programming (DP) is an algorithmic technique used for solving sequential, or multistage, decision problems and is a fundamental tool in combinatorial optimization (see, e.g., [39], [3, sect. 2.5], and [75, Chap. 8]). A stochastic discrete time finite horizon dynamic program (DP, to be distinguished from dynamic programming by context) aims to find an optimal policy over a finite time horizon that minimizes the expected cost. In each time period, the state of the system is observed before an action is taken. Based on exogenous stochastic information, the state, and the action, the system transitions into a new state at the beginning of the next time period, while a single-period cost is incurred at the same time.

We can formally model this by means of the optimality equation (or Bellman equation). Let $z_t(I_t)$ be the cost-to-go function (or value function). The value $z_t(I_t)$ is simply the cost of an optimal policy from time period t to the end of the time

*Received by the editors June 17, 2013; accepted for publication (in revised form) June 23, 2014; published electronically October 14, 2014. The first author's research was supported in part by the European Community's Seventh Framework Programme FP7/2007-2013 (grant agreement 247757), the Recanati Fund of the School of Business Administration, the Hebrew University of Jerusalem, and NSF contract CMMI-0758069. The first, second, and fifth authors' research was supported in part by NSF contracts DMI-0085683 and DMI-0245352 and by NASA interplanetary supply chain management and logistics architecture. The third author's research was supported in part by the Research Grants Council of Hong Kong under grant PolyU5228/08E. The fourth and fifth authors' research was supported in part by NSF contract CMMI-0758069. The fourth author's research was also supported in part by ONR grant N00014-98-1-0317. An extended abstract of this paper appeared in the *Proceedings of the* 19*th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2008.
http://www.siam.org/journals/sidma/28-4/92515.html
†Hebrew University, Jerusalem, Israel (halman@huji.ac.il). The research was conducted while the author was a postdoctorate associate at MIT.
‡Northwestern University, Evanston, IL 60208 (d-klabjan@northwestern.edu).
§The Hong Kong Polytechnic University, Hong Kong, China (chung-lun.li@polyu.edu.hk).
¶Massachusetts Institute of Technology, Cambridge, MA 02139 (jorlin@mit.edu, dslevi@mit.edu).

horizon, given that at the beginning of time period t the state is I_t. The optimality equation reads

$$z_t(I_t) = \min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t}\Big\{ g_t(I_t, x_t, D_t) + z_{t+1}(f_t(I_t, x_t, D_t)) \Big\}. \tag{1.1}$$

Here, x_t is the action, $\mathcal{A}_t(I_t)$ is the action set, and D_t is a vector of random variables corresponding to the stochastic exogenous information flow. The random variables are assumed to be independent, but they are not necessarily identically distributed. The system dynamics are represented by a transition function f_t, and the cost incurred in period t is g_t. In our context, I_t and x_t are one-dimensional, while D_t is a fixed-dimensional vector.

Monotone/convex DP. We study three special cases of such DPs. In the first case, for every t, the cost function g_t is nonincreasing in I_t and monotone in x_t, while the transition function f_t is nondecreasing in I_t and monotone in x_t. We call this the *nonincreasing case*. The second case, in which the conditions are analogous to the nonincreasing case, is called the *nondecreasing case*. We refer to these first two cases as the *monotone case*. In the third case, the transition function f_t is linear in I_t and x_t, while the cost function g_t has a convex structure (we give a formal definition in section 3), and we call this the *convex case*.

Fully polynomial time approximation schemes. When facing NP-hard problems, DPs will encounter difficulty in generating optimal solutions efficiently. One good resolution to this issue would be to employ fully polynomial time approximation schemes (FPTASs), which can efficiently generate solutions that are arbitrarily close to the optima. For any given tolerance ϵ, an FPTAS generates a solution with a relative error guaranteed to be no more than ϵ, while the running time of the algorithm is polynomial in $1/\epsilon$ and in the size of the problem. The essence of FPTASs is to apply a discrete approximation to the cost function so that the DP can be executed in polynomial time. It would be critical to design an algorithm and an approximation for the DP in such a way that small errors at one stage do not turn into large errors at subsequent stages.

Literature review. The earliest work conducted on FPTASs can be traced back to the mid 1970s, starting with the classic work of Ibarra and Kim [41], Horowitz and Sahni [40], and Sahni [66] on scheduling and knapsack problems. Since then, the most common techniques for constructing FPTASs have been dominance (i.e., omitting states or actions of the DP which are dominated, or approximately dominated, by other states or actions) and scaling/rounding the data (see, e.g., [39] and [3, sect. 2.5]). Although many FPTASs can be easily constructed once the key ideas from Ibarra and Kim [41] and Sahni [66] are understood, other FPTASs would require great care in algorithm design and analysis. In fact, the existence of FPTASs for some optimization problems are nontrivial. In particular, to the best of our knowledge, no FPTAS has been reported for stochastic optimization problems prior to 2006. (See the recent works of [33] and [68], where the latter deals with stochastic linear and integer programming rather than stochastic DP.)

Woeginger [79] made a key observation that many FPTASs were designed by modifying DPs, and he designed a *framework for deriving FPTASs for deterministic DPs* satisfying certain regularity conditions. His framework encompassed results from a dozen optimization problems, including the knapsack problem, for which the first FPTAS was developed in the seminal work of Ibarra and Kim [41]. However, Woeginger [79] did not address some deterministic problems that were known to have FPTASs, including treelike variants of the knapsack problem, problems involving convex or monotone functions, and stochastic optimization problems.

We note that many #P-complete problems exhibit fully polynomial randomized approximation schemes (FPRASs), for example, counting Hamiltonian cycles in dense graphs [18], counting knapsack solutions [17], counting Eulerian orientations of a directed graph [57], counting perfect matchings in a bipartite graph [43], or computing the permanent [44]. To the best of our knowledge, deterministic FPTASs for #P-hard problems known up to 2008 and published in the literature can only be found in the recent works of [77], [4], [5], and [28], which were developed by applying methods from statistical physics. Our FPTAS, which uses different methods, would provide another unique example of a (deterministic) FPTAS for #P-hard problems.

Our results. In this paper we adopt Woeginger's goal of transforming DPs into FPTASs. We develop a novel methodology that is both general and easy to use for deriving FPTASs for stochastic DPs. An important ingredient in this methodology is the development of a set of conditions such that if a DP satisfies it, then the DP admits an FPTAS. Our framework extends that of Woeginger in several key aspects. In particular, it applies to stochastic optimization problems, and it permits functions defined over a large interval of integers. Nevertheless, it is not yet a proper generalization of Woeginger's framework, and several problems with FPTASs that fit into Woeginger's framework do not fit into ours. Figure 1 depicts the interrelations between Woeginger's framework and ours. The figure is not drawn to scale; in fact, about 80% of the problems presented in [79] fall into our framework.

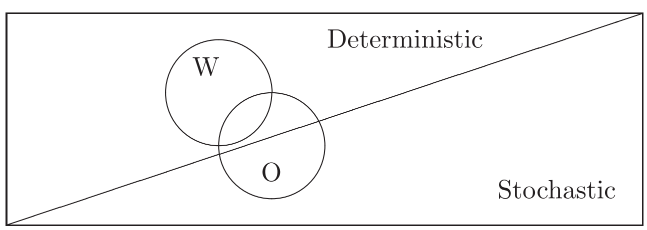


FIG. 1. *NP-hard optimization problems and the frameworks of Woeginger (W) and of ours (O).*

We show that our newly developed set of the conditions is satisfied by several basic problems in inventory control, economics, theoretical computer science, and finance. Moreover, all the FPTASs for stochastic optimization problems are new. (Recall that previous to 2006 there were no FPTASs for stochastic optimization problems.) We show that in many aspects this set of conditions is also necessary by giving inapproximability results whenever some parts of the sufficient conditions are not satisfied.

Our approach. In a previous work, we have studied a single-item stochastic dynamic inventory control problem [33]. In that work, we introduced the notions of K-approximation sets and K-approximation functions and tailored them to the specific functions involved in a certain formulation of the inventory control problem. Using this novel technique (which is different from dominance and/or scaling), we provided an ad hoc FPTAS for the inventory control problem.

Our current work also makes use of the notions of K-approximation sets and functions but aims at developing a general framework for FPTAS construction. To achieve this, we provide two sets of general computational rules for manipulating K-approximation functions and K-approximation sets, which we call *calculus of K-approximation functions* and *calculus of K-approximation sets*, respectively. While the calculus of K-approximation functions bounds the approximation ratio of the resulting functions (see the last column of Table 1), the calculus of K-approximation sets consists of a set of permissible operations on functions such that the resulting functions can be approximated without performing any additional queries to the original functions.

Assuming $\tilde{\varphi}_i$ and W_i are K_i-approximation function and K_i-approximation set, respectively, of a given function φ_i, $i = 1, 2$, and α, β are nonnegative real numbers, Table 1 summarizes which operations on which functions would admit an approximation set without further querying the functions involved. For instance, $\min\{\tilde{\varphi}_1, \tilde{\varphi}_2\}$ serves as a $\max\{K_1, K_2\}$-approximation function of $\min\{\varphi_1, \varphi_2\}$, while $W_1 \cup W_2$ serves as a $\max\{K_1, K_2\}$-approximation set for $\min\{\varphi_1, \varphi_2\}$, whenever φ_1, φ_2 are monotone functions. The calculus serves as a simple (and often automated) tool for analyzing the error propagations for monotone, unimodal, and convex functions. While some of the rules in the calculus are straightforward, others (e.g., Propositions 5.3, 7.1, and 7.2) are far more subtle (including some of the rules restricted to convex functions) and demand more thorough analysis.

TABLE 1
The calculus of K-approximation sets and functions.

Operation (nickname)	φ_i unimodal	φ_i monotone	φ_i convex	Apx. set	Apx. ratio (Prop. 5.1)
$\varphi(\psi)$ (composition)	Prop. 6.1 (3)	Prop. 6.1 (3)	Prop. 6.1 (3)	$\psi^{-1}(W_1)$	K_1
$\alpha + \beta\varphi_1$ (linearity)	Prop. 6.1 (4)	Prop. 6.1 (4)	Prop. 6.1 (4)	W_1	K_1
$\max\{\varphi_1, \varphi_2\}$ (maximization)	Prop. 6.1 (5)	Prop. 6.1 (5)	Prop. 6.1 (5)	$W_1 \cup W_2$	$\max\{K_1, K_2\}$
$\min\{\varphi_1, \varphi_2\}$ (minimization)	-	Prop. 6.2 (2)	-	$W_1 \cup W_2$	$\max\{K_1, K_2\}$
$\varphi_1 + \varphi_2$ (summation)	-	Prop. 6.2 (1)	Prop. 6.3	$W_1 \cup W_2$	$\max\{K_1, K_2\}$

Comparison to [33]. We now highlight the main difference in contributions between our previous work [33], where we introduced the concepts of K-approximation sets and functions, and the current paper. In [33] the main assumption is that the single-period cost functions are integer-valued convex functions over a contiguous interval of integers. Using this assumption and the notions of K-approximation sets and functions, we designed an FPTAS to a specific inventory control problem. The focus of the current work is on a more general setting of general monotone functions (which are not necessarily integer-valued convex). Indeed, eight of the applications presented in Table 2 satisfy the monotone assumption.

Our contribution. The contribution of this paper is sevenfold. First, we model the DP as a nicely structured function and carefully develop a theory (calculus) for the error propagation of functions. This theory is very convenient to work with and, therefore, is fruitful for further research. Second, in addition to developing the calculus, we develop a sufficient set of conditions that guarantee the existence of an FPTAS.

TABLE 2
Applications of the new framework.

	Problem	Previous results	New results	Case
1	Stochastic ordered adaptive knapsack problem	NP-hardness; polynomial-time algorithm which gives a solution whose value is at least the optimal value and violates the knapsack constraint by no more than $\epsilon \times 100\%$ [15]	FPTAS	M
2	Nonlinear knapsack problem	NP-hardness; FPTASs for various special cases [51, 38, 65, 10, 45]	FPTAS for the general model with monotone objective and monotone packing constraint	M
3	Dynamic capacity expansion	NP-hardness; pseudopolynomial time algorithm for the special case with cost function $\pi_{t,i}(x_{t,i}) = x_{t,i}\pi_i\gamma^{t-1}$ [67]	FPTAS	M
4	Time-cost trade-off machine scheduling	NP-hardness; FPTAS for the case with a maximization objective and a linear tradeoff function [13]	FPTAS for the minimization model with a general monotone tradeoff function	M
5	Single-item stochastic inventory control	#P-hardness; FPTAS [33]	Problem fits into our framework; approximated limit policy	C
6	Single-item stochastic batch dispatch	Heuristics without provably bounded error for the special case of time-independent costs [61]	FPTAS; #P-hardness proof	M
7	Single-resource revenue management	Pseudopolynomial time algorithm [73]	FPTAS; #P-hardness proof	M
8	Lifetime consumption of risky capital	DP formulation for the model with discounted utility function and stationary growth rate distribution [64]	FPTAS for the general model with time-varying utility function and growth rate distribution; #P-hardness proof	M
9	Stochastic growth model	DP formulation for a model under different assumptions [1]	FPTAS; #P-hardness proof	M
10	Cash management problem	Pseudopolynomial time algorithm [16]; heuristic that converges to the optimum [60]	FPTAS; #P-hardness proof; approximated limit policy	C

Third, we illustrate the generality and applicability of our framework by providing FPTASs to 10 different optimization problems as summarized in Table 2. (No FPTAS has been reported in the literature for any of these problems except for problem 5.) Fourth, we give new hardness results to five different optimization problems as summarized in Table 2. Fifth, we meticulously study the limits of our framework and show that it cannot be relaxed to deal with general nonindependent random variables (Corollary 10.2) and that the condition dealing with convex DP essentially cannot be relaxed (Theorem 9.2). Sixth, we show how to construct approximations to monotone functions that cannot be accessed directly and are instead accessed by nonmonotone functions that approximate them (section 4.2). This is a key ingredient in the development of our FPTAS for monotone DP and may be of independent interest. Seventh, we use the notion of discrete convexity (formally defined in the next section), together with some additional technical conditions to prove that function z_t in (1.1) is convex (Proposition 9.1). This is a key ingredient in the development of our FPTAS for convex DP and may also be of independent interest.

Applications. Our newly developed framework has numerous applications. To demonstrate the applicability of our framework, we present 10 examples of such applications. A summary of the previous results and the new results of these 10 problems is presented in Table 2. Formal definitions and detailed explanations of how these problems fit into our framework are available in Appendix A. Problems 1–3 are variants of the classical 0/1 knapsack problem, problems 4–7 are related to logistics and operations management, and problems 8–10 are in the areas of economics and mathematical finance. Some of these problems are deterministic, and some are stochastic. Note that no FPTAS has been reported in the literature for any of these problems except problem 5.

1. *Stochastic ordered adaptive knapsack problem* [15]. A number of items are to be considered sequentially for placing into a knapsack. Each item i has a deterministic profit π_i and a stochastic volume v_i in which the distribution is known in advance. The actual volume of an item is unknown until we instantiate the item by attempting to place it into the knapsack, and we have to decide whether to select the item for packing. The packing process will be terminated once the knapsack capacity is exceeded. The objective is to maximize the expected total profit of the packed items.

2. *Nonlinear knapsack problem* [38, 45]. This problem is similar to the classical integer knapsack problem, in which a quantity of each given item is selected and packed into the knapsack. However, instead of having fixed volumes and profits per unit, a general nondecreasing volume function and a general nondecreasing profit function are given; that is, placing x units of item i into the knapsack will result in a profit of $\pi_i(x)$ and take up a volume of $v_i(x)$. The objective is to maximize the total profit without exceeding the knapsack's capacity.

3. *Dynamic capacity expansion* [67]. This problem is best viewed as a multi-period minimization integer knapsack problem. Given a sequence of demands $c_1, \ldots, c_T$ and a set of items $\{1, \ldots, n\}$ of various volumes v_i and time-dependent cost functions $\pi_{t,i}$ $(i = 1, \ldots, n;\ t = 1, \ldots, T)$, we would like to determine a combination of quantities of each of these items to be placed in a knapsack in each time period. The objective is to satisfy the accumulated demand at minimum cost.

4. *Time-cost trade-off machine scheduling* [13]. There is a single machine and a given set of jobs. The processing time of a job is a nonincreasing function of the amount of monetary resources allocated to it. Each job is given a due date, and a late penalty will be incurred if the job completes after its due date. The objective is to determine the job processing times and to schedule the jobs on the machine in such a way that the sum of the total late penalty and the total resource consumption is minimized.

5. *Single-item stochastic inventory control* [33]. This is a stochastic version of the classical single-item dynamic lot sizing problem. The planning horizon consists of a finite number of time periods. In each time period, the decision maker has to determine the procurement quantity of the item. Demand is stochastic and time-dependent. Any leftover at the end of a time period will be carried forward to the next period and incur an inventory holding cost. All shortages are backordered.

The procurement cost function, holding cost function, and backlogging cost function are convex, and the objective is to minimize the expected total cost.

6. *Single-item stochastic batch dispatch* [61]. Consider running a dispatch station over a finite time horizon, where a vehicle with a finite capacity is available to dispatch goods in batches. In each time period, goods arrive randomly based on a time-dependent distribution known in advance. The decision in each time period is whether we should send off the vehicle and, if yes, how many units of the goods should be carried by the vehicle. Dispatching the vehicle will incur a fixed cost as well as a per-unit cost of the dispatched goods, while the goods left at the dispatch station will incur a per-unit holding cost.

7. *Single-resource revenue management* [73, Chap. 2]. There is a single resource with a given limited capacity C (e.g., an airplane with seat capacity C for a specific flight). There are T customer classes, in which class t has a revenue contribution of r_t per arrival. All customers in class t arrive in time period t, and the number of such customers is distributed randomly based on a random variable D_t with a known distribution. We assume no cancellations or no-shows, no overbookings, and independent customer arrivals. The problem is to find acceptance policies to maximize the expected total revenue.

8. *Lifetime consumption of risky capital* [64]. Consider an individual managing her capital over a finite time horizon. In each time period, she can consume some of her capital, and the subsequent utility is derived from her consumption based on an underlying utility function. The remaining capital yields a stochastic return. In addition, she receives an income at the end of the period. The problem is to determine an optimal consumption strategy which maximizes her expected total utility.

9. *Stochastic growth model* [1, Chap. 5]. This is a variant of "lifetime consumption of risky capital." In each time period, a household decides how much of its capital it should consume, and utility is derived from its consumption. The rest of the capital can be used to produce output via a production process. There is a deterministic depreciation of the remaining capital, but fluctuations in capital are created by random shocks to the production process. The objective is to maximize the expected total utility throughout the time horizon.

10. *Cash management problem* [16, pp. 154–155]. A person needs to manage the cash flow of a mutual fund. At the beginning of each time period, the cash balance can be changed by either selling or buying stocks. At the end of each time period, the net value of deposits/withdrawals is observed, and consequently the cash balance of the mutual fund is determined. If the balance is negative, the fund will borrow money from the bank. If the balance is positive, a cost will be incurred, as the fund's money could have been invested elsewhere. The problem is to determine a policy that minimizes the total cost.

Organization of the paper. In section 2, notation is defined, and an overview of discrete convex functions is presented. Section 3 describes our framework and states the sufficient conditions needed for the framework. In section 4, we explain how K-approximation sets and functions can be built succinctly and efficiently. The development of calculus of K-approximation functions and calculus of K-approximation sets is presented in sections 5 and 6, respectively. In section 7, a theory linking K-approximation sets and functions to DP is introduced. Based on this theory, our main results for monotone DPs and convex DPs (i.e., the FPTAS together with its analysis) are discussed in sections 8 and 9, respectively. Several variants of our framework dealing with maximization problems, random vectors, correlated stochastic events, implicit descriptions of stochastic events, profit maximization, nonexact evaluation of the cost functions, etc., and an analysis of the structure of an optimal policy for convex DPs are provided in section 10. Concluding remarks are made in section 11. A detailed description of the applications of our model is available in Appendix A. Proofs of the propositions in sections 4, 5, 6, and 10 are provided in Appendix B. All proofs of computational intractability are presented in Appendix C.

2. Preliminaries.

2.1. General notation. Let $\mathbb{R}, \mathbb{Z}, \mathbb{N}, \mathbb{Q}$ denote the set of real numbers, integers, positive integers, and rational numbers, respectively. Let $D \subset \mathbb{R}$ be a finite set of real numbers. Let $D^{\min}$ and $D^{\max}$ denote the minimal and maximal elements in D, respectively. For $x < D^{\max}$, let $\text{next}(x, D) = \min\{y \in D \mid y > x\}$. For $x > D^{\min}$, let $\text{prev}(y, D) = \max\{y \in D \mid y < x\}$. For any pair of integers $A \leq B$, let $[A, \ldots, B] = \{A, A+1, \ldots, B\}$ denote the set of integers between A and B. We call $[A, \ldots, B]$ a *contiguous interval*. Let X be a set, and let $Y(x)$ be a set for every $x \in X$. We denote by $X \otimes Y$ the set $\bigcup_{x \in X}(\{x\} \times Y(x))$; see Figure 2.

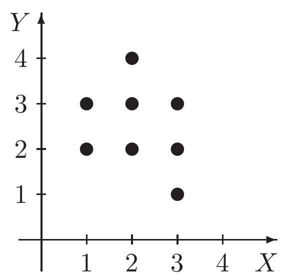


FIG. 2. $X \otimes Y$ *for* $X = \{1, 2, 3\}$, $Y(1) = \{2, 3\}$, $Y(2) = \{2, 3, 4\}$, *and* $Y(3) = \{1, 2, 3\}$.

For any $x \in \mathbb{R}$, let $x^+ = \max\{0, x\}$ and $x^- = \max\{0, -x\}$. For any $X \subseteq \mathbb{R}$, let X^+ denote the set of nonnegative numbers in X, i.e., $X^+ = \{x \in X \mid x \geq 0\}$. For example, $\mathbb{R}^+$ denotes the set of all nonnegative real numbers. For any $x \in \mathbb{R}$, let $\lceil x \rceil$ denote the smallest integer no less than x, and let $\lfloor x \rfloor$ denote the largest integer no greater than x. For every Boolean expression X, let δ_X be 1 if X is true and 0 otherwise. The base two logarithm of z is denoted by $\log z$.

Consider any real-valued function $\varphi : D \to \mathbb{R}$. Let $\arg\min \varphi = \arg\min\{\varphi(x) \mid x \in D\}$ ($\arg\max \varphi = \arg\max\{\varphi(x) \mid x \in D\}$) be any $x \in D$, where $\varphi(x)$ is minimized (maximized). Note that if function φ has multiple minimizers (maximizers), we may arbitrarily select any minimizer (maximizer) of φ as $\arg\min \varphi(x)$ ($\arg\max \varphi(x)$). Let $\varphi^{\max} = \max_{x \in D} |\varphi(x)|$. If $\varphi \not\equiv 0$, then let $\varphi^{\min} = \min\{|\varphi(x)| \mid x \in D \text{ and } \varphi(x) \neq 0\}$. We denote by t_φ the time needed to evaluate φ on a single point in its domain. Function φ is said to be *unimodal over* D if there exists $x^* \in D$ such that φ is nonincreasing over $D \cap \{x \mid x \leq x^*\}$ and nondecreasing over $D \cap \{x \mid x \geq x^*\}$. Note that monotone functions and convex functions are special cases of unimodal functions. Note also that in our context the "mode" of a unimodal function is a minimum point of the function.

Consider any multiparameter real-valued function $\varphi(x_1, \ldots, x_k)$. For $1 \leq i \leq k$, we say that φ *is monotone in* x_i (or equivalently "$\varphi(x_1, \ldots, x_{i-1}, \cdot, x_{i+1}, \ldots, x_k)$ is monotone") if either φ is nondecreasing in x_i for any fixed values of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$, or φ is nonincreasing in x_i for any fixed values of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$.

2.2. Notation for convex DPs. For any subset $E \subseteq D$, we define the *piecewise-linear extension of* φ *induced by* E as the continuous function over the domain $\{x \in D \mid E^{\min} \leq x \leq E^{\max}\}$ obtained by making φ linear between successive values of E. We define the *convex extension of* φ *induced by* E as the continuous function over the domain $\{x \in D \mid E^{\min} \leq x \leq E^{\max}\}$ obtained by making φ equal to the lower envelop of the convex hull of $\{(x, \varphi(x)) \mid x \in E\}$. Note that the convex extension of φ induced by E is a piecewise-linear function and is the greatest convex function that does not lie above φ over the points in E. For any subset $D' \subseteq D$, a function $\varphi : D \to \mathbb{R}$ is said to be *convex over* D' if its piecewise-linear extension induced by D' is convex.

We now turn to a discussion of convex functions over two-dimensional discrete domains. We consider the following important example.

Example 2.1. Let $\varphi : \mathbb{R}^2 \to \mathbb{R}$ be defined as $\varphi(x, y) = (x - 2y)^2$. Clearly, φ is convex over $\mathbb{R}^2$. Define $\psi_1, \psi_2 : \mathbb{R} \to \mathbb{R}$ such that

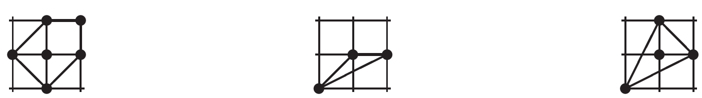
$$\psi_1(x) = \min_{y \in \mathbb{R}} \varphi(x, y)$$

and

$$\psi_2(x) = \min_{y \in \mathbb{Z}} \varphi(x, y). \tag{2.1}$$

Note that $\psi_1 \equiv 0$. Hence, it is convex over $\mathbb{R}$. On the other hand, because ψ_2 is 0 for even x's and is 1 for odd x's, it is not convex over $\mathbb{R}$.

This example shows that if we want ψ_2 to be convex over $\mathbb{R}$, we will need to impose a stronger condition on φ than just requiring φ to be convex on $\mathbb{R}^2$. To achieve this, we first explain the meaning of *integrally convex sets* introduced by Murota [59]. Let X be a contiguous interval, and let $Y(x)$ be a nonempty contiguous interval for every $x \in X$. The set $X \otimes Y = \bigcup_{x \in X} (\{x\} \times Y(x)) \subset \mathbb{Z}^2$ is said to be *integrally convex* if there exists a convex (but not necessarily bounded) polyhedron C_{XY} such that $X \otimes Y = C_{XY} \cap \mathbb{Z}^2$ and that the slopes of C_{XY}'s edges are in the set $\{-\infty, -1, 0, 1, \infty\}$; see Figure 3, which is adopted from [59].



Integrally convex　　Not integrally convex　　Not integrally convex

FIG. 3. *Concept of integrally convex sets.*

As will be seen in the proof of Proposition 9.1, a sufficient condition for ψ_2 in (2.1) to be convex is that φ is defined over an integrally convex set and that it can be expressed as $\varphi(x, y) = \varphi_1(x) + \varphi_2(y) + \omega(\tau(x, y))$, where φ_1, φ_2, ω are univariate convex functions and $\tau(x, y) = ax + by + c$ for some $a, c \in \mathbb{Z}$ and $b \in \{-1, 0, 1\}$. This is a key observation for developing our convex DP model.

We remark that in discrete optimization, discrete analogues of convexity, or "discrete convexity" for short, have been considered. Miller investigated a class of discrete functions, called "discrete convex functions," of which local optimality implies global optimality [58]. Favati and Tardella considered a certain special way of extending functions defined over the integer lattice to piecewise-linear functions defined over the real space, and they introduced the concept of "integrally convex functions" [23]. Murota introduced the concepts of "L-convexity" and "M-convexity," in which L stands for "lattice" and M stands for "matroid" [59]. L- and M-convex functions possess several desirable properties as discrete convex functions, including extendability to ordinary (continuous) convex functions, duality theorems, and conjugacy between L- and M-convex functions, etc. An alternative sufficient condition for ψ_2 in (2.1) to be convex is that function φ is "integrally convex" as defined in [23]. Because all the convex problems we solve satisfy the above mentioned sufficient condition, the alternative sufficient condition that φ should be integrally convex will not be discussed in this paper.

3. Model statement. In this section, a basic model of decision making under stochastic uncertainty over a finite number of time periods is reviewed. We consider the following formulation for a finite horizon stochastic DP, as defined in Bertsekas [7]. The model has two principal features: (i) an underlying *discrete time dynamic system*, and (ii) a *cost function that is additive over time*. The system dynamics are of the form

$$I_{t+1} = f_t(I_t, x_t, D_t), \quad t = 1, \ldots, T, \tag{3.1}$$

where t is the discrete time index, I_t is the state of the system, x_t is the action or decision to be selected in time period t, D_t is a discrete random variable, and T is the number of time periods. The cost function, denoted by $g_t(I_t, x_t, D_t)$, is additive in the sense that the cost incurred in time period t is accumulated over time. Let I_1 be the initial state of the system. Given a realization d_t of D_t, for $t = 1, \ldots, T$, the total cost is

$$g_{T+1}(I_{T+1}) + \sum_{t=1}^{T} g_t(I_t, x_t, d_t),$$

where $g_{T+1}(I_{T+1})$ is the terminal cost incurred at the end of the process. The problem is to determine

$$z^*(I_1) = \min_{x_1,\ldots,x_T} E\left\{ g_{T+1}(I_{T+1}) + \sum_{t=1}^{T} g_t(I_t, x_t, D_t) \right\}, \tag{3.2}$$

where the expectation is taken with respect to the joint distribution of the random variables involved. The optimization is over the actions $x_1, \ldots, x_T$. Here, x_t is selected with the knowledge of the current state I_t but before the realization of D_t takes place.

The state I_t is an element of a given *state space* $\mathcal{S}_t$, the action x_t is constrained to take values in a given *action space* $\mathcal{A}_t(I_t)$, and the discrete random variable D_t takes values in a given set $\mathcal{D}_t$. The state space and the action space are one-dimensional. Note that the domain of functions g_t and f_t is $(\mathcal{S}_t \otimes \mathcal{A}_t) \times \mathcal{D}_t$. The following theorem states the well-known DP recursion for this model.

THEOREM 3.1 (the DP recursion [6]). *For every initial state I_1, the optimal cost $z^*(I_1)$ of the DP is equal to $z_1(I_1)$, where the function z_1 is given by the last step of the following recursion, which proceeds backward from period T to period 1:*

$$z_{T+1}(I_{T+1}) = g_{T+1}(I_{T+1}), \tag{3.3}$$

$$z_t(I_t) = \min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t} \left\{ g_t(I_t, x_t, D_t) + z_{t+1}(f_t(I_t, x_t, D_t)) \right\}, \quad t = 1, \dots, T, \tag{3.4}$$

where the expectation is taken with respect to the probability distribution of D_t.

Note that the DP recursion given in Theorem 3.1 yields an exact solution for $z_1(I_1)$ but may require a pseudopolynomial running time. For example, if $\mathcal{A}_t(I_t) \equiv \mathcal{A}$ and $\mathcal{S}_t \equiv \mathcal{S}$ for every t and I_t, then this DP has a running time of $O(T|\mathcal{A}||\mathcal{S}|)$, but $|\mathcal{A}|$ and $|\mathcal{S}|$ may be exponential in the (binary) input size.

We assume that the random variables are given explicitly in the following way. For each D_t, we are given n_t, the number of different values it admits with positive probability, and its support $\mathcal{D}_t := \{d_{t,1}, \dots, d_{t,n_t}\}$, where $d_{t,i} < d_{t,j}$ for $i < j$. We are also given positive integers $q_{t,1}, \dots, q_{t,n_t}$ such that

$$\text{Prob}(D_t = d_{t,i}) = \frac{q_{t,i}}{\sum_{j=1}^{n_t} q_{t,j}}.$$

For every $t = 1, \dots, T$ and $i = 1, \dots, n_t$, we denote $p_{t,i} = \text{Prob}(D_t = d_{t,i})$. Then, we have

(3.5)
$$E_{D_t} \left\{ g_t(I_t, x_t, D_t) + z_{t+1}(f_t(I_t, x_t, D_t)) \right\} = \sum_{j=1}^{n_t} p_{t,j} \Big[g_t(I_t, x_t, d_{t,j}) + z_{t+1}(f_t(I_t, x_t, d_{t,j})) \Big].$$

In our analysis, the following notation will be used:

$n^* = \max_t n_t$	=	maximum number of different values that D_t can take over the entire time horizon;
$D^* = \sum_{t=1}^{T} \lvert d_{t,n_t} \rvert$	=	maximum possible total value that the random variables can take over the entire time horizon;
$U_{\mathcal{S}} = \max_{t=1,\dots,T+1} \lvert\mathcal{S}_t\rvert$	=	maximal size of the state space;
$U_{\mathcal{A}} = \max_{t=1,\dots,T} \max_{I_t \in \mathcal{S}_t} \lvert\mathcal{A}_t(I_t)\rvert$	=	maximal size of the action space.

Let $g_t^{\max} = \max_{I \in \mathcal{S}_t,\, x \in \mathcal{A}_t(I),\, d \in \mathcal{D}_t} g_t(I, x, d)$ be the maximal cost value in time period t for $t = 1, \dots, T$. Let $g_{T+1}^{\max} = \max_{I \in \mathcal{S}_{T+1}} g_{T+1}(I)$. Let

$$g_t^{\min} = \min_{I \in \mathcal{S}_t,\, x \in \mathcal{A}_t(I),\, d \in \mathcal{D}_t} \{ g_t(I, x, d) \mid g_t(I, x, d) > 0 \}, \quad t = 1, \dots, T,$$

be the minimal positive cost value in time period t. Let $g_{T+1}^{\min} = \min_{I \in \mathcal{S}_{T+1}} \{ g_{T+1}(I) \mid g_{T+1}(I) > 0 \}$. (*Note*: For $t = 1, \dots, T+1$, if $g_t \equiv 0$, then $g_t^{\min} = +\infty$.) Let

$$U_g = \frac{\max_{t=1,\dots,T+1} g_t^{\max}}{\min_{t=1,\dots,T+1} g_t^{\min}}.$$

In order to derive an FPTAS for our DP, the following conditions are needed.

CONDITION 1. *$\mathcal{S}_{T+1}, \mathcal{S}_t, \mathcal{A}_t(I_t) \subset \mathbb{Z}$ for $I_t \in \mathcal{S}_t$ and $t = 1, \dots, T$. For any set X among these sets, $\log \max_{x \in X}(|x| + 1)$ is bounded polynomially by the (binary) input size, and the kth largest element in X can be identified in constant time for any $1 \le k \le |X|$. For every $t = 1, \dots, T$, the number of different values the random variable D_t admits with positive probability is a given integer n_t, and its probability distribution* *function is given as n_t ordered pairs $(d_{t,i}, p_{t_i})$, where $p_{t,i} = \mathrm{Prob}(D_t = d_{t,i}) \in \mathbb{Q}$ for $i = 1, \dots, n_t$. Moreover, $\mathcal{D}_t \subset \mathbb{Q}$ for $t = 1, \dots, T$.*

CONDITION 2. *For every $t = 1, \dots, T+1$, functions f_t, g_t are either given explicitly (i.e., as explicit formulae) or are accessed via oracle calls. Moreover, the values of g_t are nonnegative rational numbers that are polynomially bounded by the (binary) size of the input.*[1]

CONDITION 3. *At least one of the following properties holds:*

(i) (Nondecreasing DP) *Function g_{T+1} is nondecreasing. For $t = 1, \dots, T$, function f_t is nondecreasing in its first variable and monotone in its second variable, and g_t is monotone in its second variable. Moreover, for each $t = 1, \dots, T$, either z_t is nondecreasing, or g_t is nondecreasing in its first variable and $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \geq I'$.*

(ii) (Nonincreasing DP) *Function g_{T+1} is nonincreasing. For $t = 1, \dots, T$, function f_t is nondecreasing in its first variable and monotone in its second variable, and g_t is monotone in its second variable. Moreover, for each $t = 1, \dots, T$, either z_t is nonincreasing, or g_t is nonincreasing in its first variable and $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \leq I'$.*

(iii) (Convex DP) *The terminal state space $\mathcal{S}_{T+1}$ is a contiguous interval. For $t = 1, \dots, T$, the state space $\mathcal{S}_t$ and the action space $\mathcal{A}_t(I)$, $\forall I \in \mathcal{S}_t$, are both contiguous intervals, and $\mathcal{D}_t \subset \mathbb{Z}$. Function g_{T+1} is a convex function over $\mathcal{S}_{T+1}$. For $t = 1, \dots, T$, the set $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex, function g_t can be expressed as $g_t(I, x, d) = g_t^I(I, d) + g_t^x(x, d) + u_t(f_t(I, x, d))$, and function f_t can be expressed as $f_t(I, x, d) = a(d)I + b(d)x + c(d)$, where $g_t^I(\cdot, d), g_t^x(\cdot, d), u_t(\cdot)$ are univariate nonnegative convex functions, $a(d) \in \mathbb{Z}$, $b(d) \in \{-1, 0, 1\}$, and $c(d) \in \mathbb{Z}$.*

The input data of the problem consists of the number of time periods T, the initial state I_1, and the explicit description of the random variables as described in Condition 1. We call DP formulation (3.3)–(3.4) *monotone* whenever it satisfies either Condition 3(i) or Condition 3(ii) and *convex* whenever it satisfies Condition 3(iii).

Condition 1 requires the kth largest element in each of the state and action spaces to be obtainable in constant time. However, the monotone DP model and the convex DP model remain valid if this requirement is relaxed by allowing the kth largest element in each of the state and action spaces to be obtainable in time polylogarithmic in the size of the space. Note that whenever the state and action spaces are contiguous intervals, one can find the kth largest element in constant time. This is indeed the situation in the convex case and in most applications of the monotone case. Condition 1 also requires that $\mathcal{D}_t \subset \mathbb{Q}$. However, the monotone DP model and the convex DP model remain valid if $\mathcal{D}_t \subset \mathbb{Q}^\ell$, where ℓ is a positive integer constant (see section 10.2 for details). Note that Condition 1 implies that $\log U_{\mathcal{S}}$ and $\log U_{\mathcal{A}}$ are polynomially bounded by the input size, and Condition 2 implies that $\log U_g$ is polynomially bounded by the input size.

At first glance, one may think that Condition 3, with its three cases, is quite cumbersome. It is due to our effort to formulate it in a general way. As shown in the 10 examples described in section 1 (with the details provided in Appendix A), each of the three cases has applications. Condition 3(iii) is somewhat restrictive. Unfortunately, as shown in Theorem 9.2, the condition "$b \in \{-1, 0, 1\}$" and the

[1] After the paper was accepted for publication, the authors noted that the requirement that the values of g_t are polynomially bounded by the input size can be omitted if one includes U_g in the input size.

condition that "$\mathcal{S}_t \otimes \mathcal{A}_t$ is an integrally convex set" are both needed to ensure that the convex DP model admits an FPTAS. (We note in passing that the condition "$\mathcal{S}_t \otimes \mathcal{A}_t$ is an integrally convex set" implies that both $\mathcal{S}_t$ and $\mathcal{A}_t$ are contiguous intervals.)

We aim to provide an FPTAS for generating an approximated value of $z_1(I_1)$. Note that even in the very restrictive case where the number of states in the system is a constant, computing the optimal solution by the DP recursion in Theorem 3.1 can take up to $\sum_{t=1}^{T} \max_I |\mathcal{A}_t(I)|$ evaluations of g_t. When the action spaces are "large," this number can be exponential in the input size. Woeginger designed a framework for deriving an FPTAS for deterministic DPs [79]. Among various assumptions, he requires the cardinality of the action space to be bounded by a polynomial over the binary input size (Condition C.4(ii) in [79]). Our work does not require this assumption. Hence, our framework, when applied to deterministic DPs, is not a special case of Woeginger's framework. The main result of this paper, proved in sections 4–9, is stated in the following theorem.

THEOREM 3.2. *Every stochastic minimization DP satisfying Conditions* 1–3 *admits an FPTAS.*

Our result also applies to maximization problems, where the DP recursion (3.4) has a "max" function instead of a "min" function. In order to achieve this, Condition 3 for maximization problems is reformulated as follows.

CONDITION 4. *At least one of the following properties holds:*

(i) (Nondecreasing DP) *Function g_{T+1} is nondecreasing. For $t = 1,\ldots,T$, function f_t is nondecreasing in its first variable and monotone in its second variable, and g_t is monotone in its second variable. Moreover, for each $t = 1,\ldots,T$, either z_t is nondecreasing, or g_t is nondecreasing in its first variable and $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \le I'$.*

(ii) (Nonincreasing DP) *Function g_{T+1} is nonincreasing. For $t = 1,\ldots,T$, function f_t is nondecreasing in its first variable and monotone in its second variable, and g_t is monotone in its second variable. Moreover, for each $t = 1,\ldots,T$, either z_t is nonincreasing or g_t is nonincreasing in its first variable and $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \ge I'$.*

(iii) (Concave DP) *The terminal state space $\mathcal{S}_{T+1}$ is a contiguous interval. For $t = 1,\ldots,T$, the state space $\mathcal{S}_t$ and the action space $\mathcal{A}_t(I)$, $\forall I \in \mathcal{S}_t$, are both contiguous intervals, and $\mathcal{D}_t \subset \mathbb{Z}$. Function g_{T+1} is a concave function over $\mathcal{S}_{T+1}$. For $t = 1,\ldots,T$, the set $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex, function g_t can be expressed as $g_t(I,x,d) = g_t^I(I,d) + g_t^x(x,d) + u_t(f_t(I,x,d))$, and function f_t can be expressed as $f_t(I,x,d) = a(d)I + b(d)x + c(d)$, where $g_t^I(\cdot,d), g_t^x(\cdot,d), u_t(\cdot)$ are univariate nonnegative concave functions, $a(d) \in \mathbb{Z}$, $b(d) \in \{-1,0,1\}$, and $c(d) \in \mathbb{Z}$.*

THEOREM 3.3. *Every stochastic maximization DP satisfying Conditions* 1, 2, *and* 4 *admits an FPTAS.*

In the analysis presented in sections 4–9, we focus on minimization problems. A discussion of the validity of Theorem 3.3 is provided in section 10.1. Details of the 10 problems mentioned in the introduction and specifically how each problem is cast as either monotone or convex DP are provided in Appendix A.

4. *K*-approximation sets and functions.

Suppose $\varphi : D \to \mathbb{R}^+$ is an arbitrary implicit function over a finite domain D, and φ is accessed via an oracle in t_φ time units. We want to preprocess a representation for it such that any further evaluation of $\varphi(\cdot)$ will be done by this representation instead of querying $\varphi(\cdot)$ directly. Of course, by querying all values $\varphi(x)$ in $x \in D$ and storing them in a sorted array of the form $\{(x, \varphi(x)) \mid x \in D\}$, we can obtain in $O(|D| t_\varphi)$ time a representation of size $O(|D|)$ which can return the value $\varphi(x)$ for any x in $O(\log |D|)$ time. However, whenever either $|D|$ or t_φ is large, we would like to have a representation that takes less space or fewer queries to construct. We say that a representation is *succinct* if its size is polylogarithmic in $|D|$ and $\frac{\varphi^{\max}}{\varphi^{\min}}$ and that a representation is *efficient* if it can be built in time polylogarithmic in both of these terms. (*Note*: Recall that $\varphi^{\min} = \min\left\{|\varphi(x)| \;\middle|\; x \in D \text{ and } \varphi(x) \neq 0\right\}$. Hence, if $\varphi \not\equiv 0$, then $\varphi^{\min} > 0$, and the ratio $\frac{\varphi^{\max}}{\varphi^{\min}}$ is well-defined. If $\varphi \equiv 0$, then $\varphi^{\min}$ is undefined, and we will refer to $\frac{\varphi^{\max}}{\varphi^{\min}}$ as 1.)

Of course, not all functions admit efficient succinct representations. In fact, even polynomial functions do not admit efficient succinct representations in general. However, as shown below, if the given function φ is a unimodal function (e.g., monotone functions and convex/concave functions), then we can build a step function $\hat{\varphi}$ (see Definitions 4.4 and 4.2 below) to approximate it, and this step function admits an efficient succinct representation.

DEFINITION 4.1. *Let $K \geq 1$ and $r, \tilde{r} \geq 0$. We say that $\tilde{r}$ is a K-*approximation value of r *(or more briefly, a K-approximation of r) if $r \leq \tilde{r} \leq Kr$. Let $\varphi, \tilde{\varphi} : D \to \mathbb{R}^+$ be real-valued functions over a finite set D. We say that φ is* succinct *if it admits a representation in space polylogarithmic in $|D| + \frac{\varphi^{\max}}{\varphi^{\min}}$. Function $\tilde{\varphi} : D \to \mathbb{R}^+$ is said to be a K-*approximation function of φ *(or more briefly, a K-approximation of φ) if $\varphi(x) \leq \tilde{\varphi}(x) \leq K\varphi(x)$ for all $x \in D$ (i.e., if $\tilde{\varphi}(x)$ is a K-approximation value of $\varphi(x)$ for all $x \in D$). Function $\tilde{\varphi}$ is called a* succinct K-approximation of φ *if it is a succinct function and is a K-approximation of φ. Such a function $\tilde{\varphi}$ is said to be* efficient *if it can be constructed in time polylogarithmic in $|D| + \frac{\varphi^{\max}}{\varphi^{\min}}$.*

Remark 1. If φ is a well-structured function, e.g., a monotone function, a K-approximation of it is not necessarily so. In this section we show how to construct K-approximation functions that do maintain the structure of the function they approximate.

In order to get succinct approximations, we consider only succinct subsets of the domain. (Of course, this can be done only by sacrificing accuracy.)

DEFINITION 4.2. *Let $K \geq 1$, and let $\varphi : D \to \mathbb{R}^+$ be a real-valued function over a finite domain of real numbers. We say that $W \subseteq D$ is a K-*approximation set of φ *if the following three conditions are satisfied:*

1. $D^{\min}, D^{\max} \in W$.
2. *(Boundedness) For every $x \in W \setminus \{D^{\max}\}$, either $next(x, D) \in W$ or $\max\{\varphi(x), \varphi(next(x, W))\} \leq K \min\{\varphi(x), \varphi(next(x, W))\}$.*
3. *(Locality) $\max\{\varphi(prev(x, W)), \varphi(next(x, W))\} \leq K\varphi(x)$ for every $x \in D \setminus W$.*

Remark 2. The notion of K-approximation sets was introduced in [33]. The original definition of K-approximation set in [32, 33] (called the weak K-approximation set in [32]) required that $\arg\min \varphi \in W$ and that W satisfies the first two conditions of Definition 4.2. In our new definition, we do not require W to include $\arg\min \varphi$. As indicated in Proposition 4.3, this new definition is a generalization of the original definition, and it provides us with stronger properties. For example, in section 6 we have Proposition 6.3 and property 5 of Proposition 6.1, which do not hold under the original definition of K-approximation set.

It is easy to check that for any unimodal function φ, if $\arg\min \varphi \in W$, then the first two conditions of Definition 4.2 automatically imply the third condition. Hence, we have the following proposition.

PROPOSITION 4.3. *Let $K \geq 1$, and let $\varphi : D \to \mathbb{R}^+$ be a unimodal function over a finite domain of real numbers. Let W be a subset of D that satisfies the first two conditions of Definition* 4.2. *If* $\arg\min \varphi \in W$, *then W is a K-approximation set of φ.*

Note that if φ is a monotone function, then by Proposition 4.3, any subset W of D that satisfies the first two conditions of Definition 4.2 is a K-approximation set of φ.

4.1. Direct access to φ. In this section we show that every unimodal (e.g., convex or monotone) function $\varphi : D \to \mathbb{R}$ with a given argmin admits a succinct approximation that preserves the convex/monotone/unimodal structure of φ. Suppose W is a subset of D that contains $D^{\min}, D^{\max}$. Having access to φ, we can construct the following approximation of φ.

DEFINITION 4.4. *Let $\varphi : D \to \mathbb{R}^+$ be a real-valued function over a finite domain of real numbers. Let $W \subseteq D$ be a set that contains $D^{\min}, D^{\max}$. The* approximation of φ induced by W *is*

$$\hat{\varphi}(x) = \begin{cases} \varphi(x) & \text{if } x \in W; \\ \max\{\varphi(prev(x, W)), \varphi(next(x, W))\} & \text{otherwise.} \end{cases}$$

In the next proposition we show that K-approximation sets are useful for getting K-approximation functions. The proof of this proposition is provided in Appendix B. We give more properties of K-approximation sets in section 6.

PROPOSITION 4.5 (approximation of a unimodal function with direct access). *Let $\varphi : D \to \mathbb{R}^+$ be a unimodal function over a finite domain of real numbers. For any $K \geq 1$, any K-approximation set W of φ, and any minimizer $x^*_W = \arg\min\{\varphi(x) \mid x \in W\}$, the following properties hold (where $\hat{\varphi}$ is the approximation of φ induced by W):*

1. *$\hat{\varphi}$ is a K-approximation of φ. In addition, if φ is stored in a sorted array $\{(x, \varphi(x)) \mid x \in W\}$, then for any $x \in D$, $\hat{\varphi}(x)$ can be determined in $O(\log |W|)$ time.*
2. *Let $W^- = \{prev(x, D) \mid x \in W \setminus \{D^{\min}\}\}$ and $W^+ = \{next(x, D) \mid x \in W \setminus \{D^{\max}\}\}$. Then, W is a K-approximation set of $\hat{\varphi}$, and $W \cup W^- \cup W^+$ is a* 1-*approximation set of $\hat{\varphi}$. If φ is nondecreasing, then $W \cup W^+$ is a* 1-*approximation set of $\hat{\varphi}$. If φ is nonincreasing, then $W \cup W^-$ is a* 1-*approximation set of $\hat{\varphi}$.*
3. *$\hat{\varphi}$ is a unimodal function minimized at x^*_W. If φ is monotone, then so is $\hat{\varphi}$. If φ is convex over D, then the convex extension of $\hat{\varphi}$ induced by W is a convex K-approximation of φ which is minimized at x^*_W.*

We say that a K-approximation set of $\varphi : D \to \mathbb{R}$ is *succinct* if its size is polylogarithmic in $|D| + \frac{\varphi^{\max}}{\varphi^{\min}}$. Clearly, if there exists a succinct 1-approximation set of φ, then φ is succinct. When no succinct 1-approximation set of φ is available, the focus would be to find succinct K-approximations of φ, for some small $K > 1$, through constructing succinct K-approximation sets for φ. In Algorithm 1 below, ApxSet is a procedure for constructing a K-approximation set for any given constant $K > 1$ and unimodal function φ which is minimized at a given x^*.

The next proposition in this section, when coupled with Proposition 4.5, tells us that for any given convex or monotone function φ, or any unimodal function φ with a given minimizer, we can efficiently build a succinct K-approximation function that preserves the same structure as φ. The proof of this proposition is provided in Appendix B.

Algorithm 1. Constructing a K-approximation set for a unimodal $\varphi : D \to \mathbb{R}$ which is minimized at x^*.

Function ApxSet(φ, D, x^*, K)
$x \leftarrow D^{\max}$
$W \leftarrow \{D^{\min}, D^{\max}\}$
while $x > D^{\min}$ **do**
  if $x > x^*$ **then** $x \leftarrow \min\{\text{prev}(x, D), \min\{y \in D \mid y \geq x^* \text{ and } K\varphi(y) \geq \varphi(x)\}\}$
  else $x \leftarrow \min\{\text{prev}(x, D), \min\{y \in D \mid K\varphi(x) \geq \varphi(y)\}\}$
  $W \leftarrow W \cup \{x\}$
end while
Return W

Proposition 4.6. *Let $\varphi : D \to \mathbb{R}^+$ be a unimodal function over a finite domain of real numbers. Let x^* be a minimizer of function φ. Let t_φ be an upper bound on the time needed to evaluate $\varphi(x)$ for any $x \in D$. Then, for every given parameters φ, D, x^*, and $K > 1$, function ApxSet computes a K-approximation set of φ in $O(t_\varphi(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}}) \log |D|)$ time. This K-approximation set contains x^* and has a cardinality of $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$.*

4.2. Approximated (indirect) access to φ. Sometimes, as happens when we deal with monotone DP, "direct" access to the function φ which we want to approximate is impossible, and only an access to function $\bar{\varphi}$ that L-approximates φ is available ($L > 1$). If $\bar{\varphi}$ itself is monotone, then we can build for it a K-approximation set W and, as we shall see in the next section, the approximation of $\bar{\varphi}$ induced by W is a monotone KL-approximation of φ. Suppose now that φ is nondecreasing but $\bar{\varphi}$ is not necessarily such. This begs the question: Is it still possible to efficiently build a succinct nondecreasing approximation function for φ? The answer is in the affirmative, though some extra work is involved. First, in Algorithm 2 below, we construct a subset $\bar{W}$ of the domain D of φ. We then define $\tilde{\varphi}$ to be the maximal nondecreasing function that is bounded from above by $\bar{\varphi}$ over $\bar{W}$. Proposition 4.7 below, which serves as a key proposition in the proof of the FPTAS for the monotone DP case, tells us that $\bar{W}$ is a K-approximation set of $\tilde{\varphi}$ and that $\tilde{\varphi}$ is a nondecreasing KL-approximation of φ. The proof of this proposition is given in Appendix B.

Algorithm 2. Constructing a subset of D for a function $\bar{\varphi}$ that approximates a nondecreasing function φ.

Function IndirectApxSet$(\bar{\varphi}, D, K)$
$x \leftarrow D^{\max}$ and $\bar{W} \leftarrow \{D^{\min}, D^{\max}\}$
while $x > D^{\min}$ and $K\bar{\varphi}(D^{\min}) < \bar{\varphi}(x)$ **do**
  $x \leftarrow x' \mid x' < x$ and $K\bar{\varphi}(x') < \bar{\varphi}(x)$ and $K\bar{\varphi}(\text{next}(x', D)) \geq \bar{\varphi}(x)$
  $\bar{W} \leftarrow \bar{W} \cup \{x, \text{next}(x, D)\}$
end while
Return $\bar{W}$

Proposition 4.7 (succinct approximation of a nondecreasing function via an L-approximation general oracle). *Let $\varphi : D \to \mathbb{R}^+$ be a nondecreasing function over a finite domain of real numbers. Let $\bar{\varphi}$ be an (unnecessarily nondecreasing) L-approximation function of φ ($L > 1$). Let $\bar{W}$ be the output of function IndirectApxSet for given parameters $\bar{\varphi}$, D, and $K > 1$. Let $\tilde{\varphi}$ be the maximal nondecreasing function* *that is bounded from above by $\bar{\varphi}$ over $\bar{W}$. Let $t_{\bar{\varphi}}$ be an upper bound on the time needed to evaluate $\bar{\varphi}(x)$ for any $x \in D$. Then, $\bar{W}$ is a K-approximation set of $\tilde{\varphi}$, $\tilde{\varphi}$ is a nondecreasing KL-approximation step function of φ, function IndirectApxSet computes $\bar{W}$ in $O(t_{\varphi}(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}}) \log |D|)$ time, and $|\bar{W}| = O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$.*

We conclude this section with the following example, which demonstrates the outcome of Algorithm 2 on the specific instance of φ and $\bar{\varphi}$ given in Table 3. This example also shows the resulted function $\tilde{\varphi}$ and illustrates that the approximation ratio of $\tilde{\varphi}$ may equal the worst-case bound guaranteed by Proposition 4.7.

Table 3
An example of a 4-approximation of φ built via a nonmonotone 2-approximation $\bar{\varphi}$ of φ.

Objects \ i	0	1	2	3	4	5	6	7	8	9	10
$\varphi(i)$	0	1	1	2	2	2	2	2	4	4	5
$\bar{\varphi}(i)$	0	2	1	4	3	3	4	3	6	5	8
$\bar{W}$	*	*	*	*		*	*				*
$\tilde{\varphi}$	0	3	3	3	3	3	4	8	8	8	8

Example 4.8. Let $L = K = 2$ and $D = [0, 10]$. Let φ and $\bar{\varphi}$ be as given in Table 3. The black dots in Figure 4 are the values of φ, and the small circles are the values of $\bar{\varphi}$. Note that $\bar{\varphi}$ is a nonmonotone 2-approximation of the nondecreasing function φ. The modification of $\bar{\varphi}$ to a maximal nondecreasing function bounded from above by $\bar{\varphi}$ (i.e., the solid step function with small circle endpoints in Figure 4) is a nondecreasing 2-approximation of φ, but since it is computed in linear time, it is too costly to compute. The big circles are the values of $\bar{\varphi}(x)$ for those $x \in \bar{W}$, where set $\bar{W}$ is constructed by Algorithm 2. (See also the proof of Proposition 4.7 for details.) Function $\tilde{\varphi}$ is the dashed step function with big circle endpoints in Figure 4. It is a nondecreasing 4-approximation of φ.

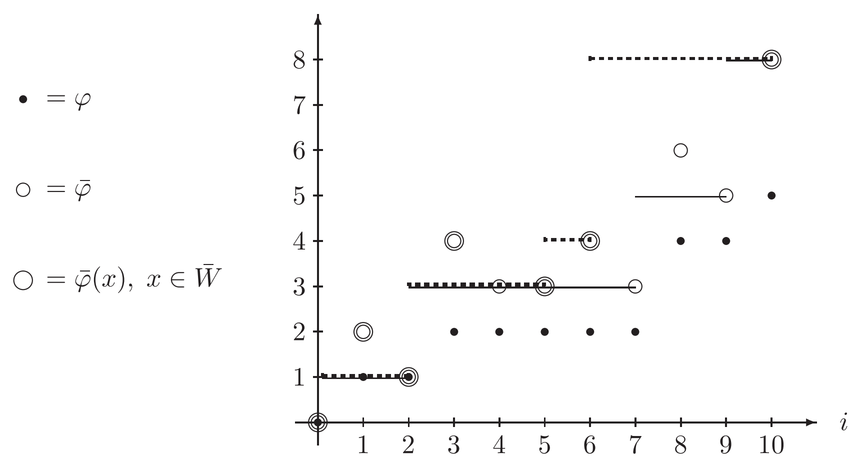


Fig. 4. *Constructing $\bar{W}$ and $\tilde{\varphi}$ in Example* 4.8.

5. Calculus of *K*-approximation functions. In this section, a set of computational rules for manipulating K-approximation functions is developed. The following proposition, referred as calculus of K-approximation functions, follows directly from the definition of K-approximation functions, and its proof is therefore omitted.

(Properties 2, 3, and 4 in the proposition are adopted from [33].)

PROPOSITION 5.1 (calculus of K-approximation functions). *For $i = 1, 2$, let $K_i \geq 1$, let $\varphi_i : D \to \mathbb{R}^+$ be an arbitrary function over any finite domain D, and let $\tilde{\varphi}_i : D \to \mathbb{R}^+$ be a K_i-approximation of φ_i. Let $\psi_1 : D' \to D$ be an arbitrary function over any finite domain D'. Let $\alpha, \beta \in \mathbb{R}^+$. The following properties hold:*

1. *φ_1 is a 1-approximation of itself.*
2. *(Linearity of approximation) $\alpha + \beta\tilde{\varphi}_1$ is a K_1-approximation of $\alpha + \beta\varphi_1$.*
3. *(Summation of approximation) $\tilde{\varphi}_1 + \tilde{\varphi}_2$ is a $\max\{K_1, K_2\}$-approximation of $\varphi_1 + \varphi_2$.*
4. *(Composition of approximation) $\tilde{\varphi}_1(\psi_1)$ is a K_1-approximation of $\varphi_1(\psi_1)$.*
5. *(Maximization of approximation) $\max\{\tilde{\varphi}_1, \tilde{\varphi}_2\}$ is a $\max\{K_1, K_2\}$-approximation of $\max\{\varphi_1, \varphi_2\}$.*
6. *(Minimization of approximation) $\min\{\tilde{\varphi}_1, \tilde{\varphi}_2\}$ is a $\max\{K_1, K_2\}$-approximation of $\min\{\varphi_1, \varphi_2\}$.*
7. *(Approximation of approximation) If $\varphi_2 = \tilde{\varphi}_1$, then $\tilde{\varphi}_2$ is a K_1K_2-approximation of φ_1.*

Table 4 summarizes the places where the calculus of K-approximation functions is used.

TABLE 4
Places where the calculus of K-approximation functions is used.

Rule	Prop. 5.2	Prop. 5.3	Prop. 6.4	Prop. 7.1	Prop. 7.2	Thm. 9.3	Sect. 10.1
Linearity				✓	✓	✓	
Summation	✓						
Composition	✓	✓					
Maximization							✓
Minimization	✓						
Approximation			✓				

The following two propositions will be useful in section 7 when K-approximation sets and functions are linked with DP. Proofs of these propositions are available in Appendix B.

PROPOSITION 5.2 (minimization of summation of composition). *Let $n \in \mathbb{N}$, let $K_i \geq 1$ for $i = 1, \ldots, n$, let D be any finite domain, and let $C(x)$ be any finite domain for every $x \in D$. Let $\varphi_i : D \to \mathbb{R}^+$, let $\tilde{\varphi}_i$ be a K_i-approximation of φ_i, and let $\psi_i : D \otimes C \to D$ for $i = 1, \ldots, n$. Let $\varphi, \tilde{\varphi} : D \to \mathbb{R}^+$ such that*

$$\varphi(x) = \min_{y \in C(x)} \left\{ \sum_{i=1}^{n} \varphi_i(\psi_i(x, y)) \right\} \quad \text{and} \quad \tilde{\varphi}(x) = \min_{y \in C(x)} \left\{ \sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y)) \right\}.$$

Then, $\tilde{\varphi}$ is a $\max\{K_1, \ldots, K_n\}$-approximation of φ.

Since the cardinality of $C(x)$ may be "big," applying Proposition 5.2 and calculating the minimum over all the elements of $C(x)$ may take time exponential in the input size. For this reason, we would like to "approximate" $C(x)$ succinctly in such a way that performing the minimization of $\sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y))$ under this approximated set, instead of under the entire $C(x)$, will result in an efficient K-approximation of φ for some "reasonable" K. This would be possible whenever the functions to be approximated are monotone.

PROPOSITION 5.3. *For $i = 1, \ldots, n$, let $K_i, L_i \geq 1$, let $\varphi_i : D \to \mathbb{R}^+$ be a function with a finite domain $D \subset \mathbb{R}$, let $\tilde{\varphi}_i : D \to \mathbb{R}$ be an L_i-approximation of φ_i, and let*

$\psi_i : D \otimes C \to D$ be a function such that for any fixed $x \in D$, $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is monotone over a finite linearly ordered domain $C(x)$. Let m be an integer such that $1 \le m \le n$. For any $i = 1,\ldots,m$ and any $x \in D$, let $W_i(x) \subseteq C(x)$ be a K_i-approximation set of $\tilde{\varphi}_i(\psi_i(x,\cdot))$. Let $\varphi, \tilde{\varphi} : D \to \mathbb{R}^+$ such that

$$\varphi(x) = \min_{y \in C(x)} \left\{ \sum_{i=1}^{n} \varphi_i(\psi_i(x,y)) \right\} \quad \textit{and} \quad \tilde{\varphi}(x) = \min_{y \in \bigcup_{i=1}^{m} W_i(x)} \left\{ \sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x,y)) \right\}.$$

Suppose for every $x \in D$, the function $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is monotone in one direction (e.g., nondecreasing) for $i = 1,\ldots,m$ and is monotone in the other direction (e.g., nonincreasing) for $i = m+1,\ldots,n$. Then, $\tilde{\varphi}$ is a $\max\{K_1L_1,\ldots,K_mL_m, L_{m+1},\ldots,L_n\}$-approximation of φ.

Remark 3. Note that while Proposition 5.3 provides an upper bound to the approximation ratio of $\tilde{\varphi}$, function $\tilde{\varphi}$ is not necessarily monotone. However, scanning $\tilde{\varphi}$ (i.e., reading its values) in a linear way (e.g., sequentially from $D^{\min}$ to $D^{\max}$) and using the monotonicity of the original function φ, one can build a monotone K-approximation function for φ. This approach will be adopted in section 8.

6. Calculus of *K*-approximation sets. In this section, a set of computational rules based on the notion of K-approximation sets and functions is developed. Unlike the calculus of K-approximation functions, which focuses on the *range* of the functions, the calculus of K-approximation sets focuses on the *domain* of the functions. Detailed proofs of the following propositions are available in Appendix B.

PROPOSITION 6.1 (calculus of K-approximation sets of unimodal functions). *Let $K_1, K_2 \ge 1$. Let $\varphi_1 : D \to \mathbb{R}^+$ and $\varphi_2 : D \to \mathbb{R}^+$ be unimodal functions with a finite domain D of real numbers. Let W_i be a K_i-approximation set of φ_i for $i = 1,2$. Let $\psi : D' \to D$ be a monotone function with a finite domain D' of real numbers. Denote $\psi^{-1}(W_i) = \{ \max\{x \mid \psi(x) \le y\}, \min\{x \mid \psi(x) \ge y\} \mid y \in W_i \}$ if ψ is nondecreasing and $\psi^{-1}(W_i) = \{ \max\{x \mid \psi(x) \ge y\}, \min\{x \mid \psi(x) \le y\} \mid y \in W_i \}$ if ψ is nonincreasing. Let $\alpha, \beta \in \mathbb{R}^+$. The following properties hold:*

1. *D is a 1-approximation set of φ_1.*
2. *(Monotonicity of approximation sets) Every superset W' of W_1, where $W' \subseteq D$, is a K_1-approximation set of φ_1.*
3. *(Composition of approximation sets) $\psi^{-1}(W_1)$ is a K_1-approximation set of $\varphi_1(\psi)$.*
4. *(Linearity of approximation sets) W_1 is a K_1-approximation set of $\alpha + \beta\varphi_1$.*
5. *(Maximization of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\max\{\varphi_1, \varphi_2\}$.*

If the functions involved are monotone in addition to being unimodal, then three more rules hold.

PROPOSITION 6.2 (calculus of K-approximation sets of monotone functions). *Let $K_1, K_2 \ge 1$. Let $\varphi_1 : D \to \mathbb{R}^+$ and $\varphi_2 : D \to \mathbb{R}^+$ be monotone functions of the same direction (i.e., either both are nondecreasing or both are nonincreasing) with a finite domain D of real numbers. Let W_i be a K_i-approximation set of φ_i for $i = 1,2$. The following properties hold:*

1. *(Summation of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\varphi_1 + \varphi_2$.*
2. *(Minimization of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\min\{\varphi_1, \varphi_2\}$.*

3. *(Approximation of approximation sets) If φ_1 is a K_2-approximation of the restriction of φ_2 over W_1, then $\hat{\varphi}_1$ (i.e., the approximation of φ_1 induced by W_1) is a K_1K_2-approximation of φ_2.*

If the functions involved are convex in addition to being unimodal, then one more rule holds.

PROPOSITION 6.3 (calculus of K-approximation sets of convex functions). *Let $K_1, K_2 \geq 1$. Let $\varphi_1 : D \rightarrow \mathbb{Z}^+$ and $\varphi_2 : D \rightarrow \mathbb{Z}^+$ be convex over a finite domain D of real numbers. Let W_i be a K_i-approximation set of φ_i for $i = 1, 2$. Then,*

(Summation of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\varphi_1 + \varphi_2$.

Note that the calculus of K-approximation sets of unimodal functions includes neither summation of approximation sets nor minimization of approximation sets since unimodal functions are not closed under either summation or minimization. Moreover, the calculus of K-approximation sets of convex functions does not include minimization of approximation sets since the minimum of two convex functions is not necessarily convex or even unimodal. Table 5 summarizes the places where the calculus of K-approximation sets is used.

TABLE 5
Places where the calculus of K-approximation sets is used.

Rule	Prop. 6.1	Prop. 6.2	Prop. 6.3	Prop. 7.1	Thm. 8.2	Thm. 9.3
Monotonicity	✓	✓	✓			
Composition				✓	✓	
Linearity				✓		✓
Approximation					✓	

The last proposition in this section is as follows.

PROPOSITION 6.4. *Let $K', L' \geq 1$, and let φ be a convex function. Let $\tilde{\varphi}$ be a convex L'-approximation function of φ. Let W be a K'-approximation set of $\tilde{\varphi}$ and $\hat{\tilde{\varphi}}$ be the approximation of $\tilde{\varphi}$ induced by W. Then, the convex extension of $\hat{\tilde{\varphi}}$ induced by W is a convex $K'L'$-approximation of φ.*

Note that this proposition is valid because (i) by property 3 of Proposition 4.5, $\hat{\tilde{\varphi}}$ is a convex K'-approximation of $\tilde{\varphi}$; and (ii) by approximation of approximation (Proposition 5.1), $\hat{\tilde{\varphi}}$ is a $K'L'$-approximation of φ.

7. From K-approximation sets and functions to DP.

In this section, two propositions linking the notions of K-approximation sets and functions with DP are presented. The first proposition deals with monotone DPs and is applied when z_{t+1} is guaranteed to be a monotone function. Note that in (7.1) below, the function $E_{D_t}\{\tilde{g}_t(I_t, \cdot, D_t)\} + E_{D_t}\{\tilde{z}_{t+1}(f_t(I_t, \cdot, D_t))\}$ is not necessarily convex, and therefore we cannot use binary search to determine its minimum point. In order to find an efficient approximation, the minimization in (7.1) is over a set $W^{-1}(I_t)$, which is of size polylogarithmic in the size of the action space $\mathcal{A}_t(I_t)$.

PROPOSITION 7.1. *Suppose the DP formulation* (3.3)–(3.4) *is monotone (so either Condition* 3(i) *or Condition* 3(ii) *is satisfied). Let K', L', L'', t, and I_t be fixed values, where $K', L' \geq 1$, $1 \leq L'' \leq K'L'$, $I_t \in \mathcal{S}_t$, and $t \in [1, \ldots, T]$. Let g_t be as stated in Conditions* 3(i) *and* 3(ii). *Let $\tilde{z}_{t+1}$ be a monotone L'-approximation of z_{t+1} and W be a K'-approximation set of $\tilde{z}_{t+1}$. Let $W^{-1}(I_t) = \bigcup_{i=1}^{n_t} f_{t,i}^{-1}(I_t, W)$, where $f_{t,i}^{-1}(I_t, W) = \big\{ \max\{x_t \mid f_t(I_t, x_t, d_{t,i}) \leq w\}, \min\{x_t \mid f_t(I_t, x_t, d_{t,i}) \geq w\} \bigm| w \in W \big\}$ if f_t is nondecreasing in its second variable, and $f_{t,i}^{-1}(I_t, W) = \big\{ \max\{x_t \mid$*

$f_t(I_t,x_t,d_{t,i})\geq w\}$, $\min\{x_t \mid f_t(I_t,x_t,d_{t,i})\leq w\} \bigm| w\in W\}$ if f_t is nonincreasing in its second variable. Let $\tilde{g}_t(I_t,\cdot,D_t)$ be a monotone L''-approximation of $g_t(I_t,\cdot,D_t)$. Let

$$\bar{z}_t(I_t)=\min_{x_t\in W^{-1}(I_t)} E_{D_t}\{\tilde{g}_t(I_t,x_t,D_t)+\tilde{z}_{t+1}(f_t(I_t,x_t,D_t))\}. \tag{7.1}$$

Then, $\bar{z}_t(I_t)$ is a $K'L'$-approximation value of $z_t(I_t)$, and it can be determined in $O(n_t(t_{\tilde{g}_t}+t_{f_t}+t_{\tilde{z}_{t+1}})|W^{-1}(I_t)|)$ time if the elements of $W^{-1}(I_t)$ are given.

Proof. Since f_t is monotone in its second variable and $\tilde{z}_{t+1}$ is monotone, the function $E_{D_t}\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))$ is monotone. Also, since $\tilde{g}_t$ is monotone in its second variable, the function $E_{D_t}\tilde{g}_t(I_t,\cdot,D_t)$ is monotone. We consider two different cases.

Case 1. $E_{D_t}\tilde{g}_t(I_t,\cdot,D_t)$ and $E_{D_t}\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))$ are monotone in the same direction. We consider the situation where these two functions are nondecreasing. (The analysis for the nonincreasing case follows a similar argument.) Under this situation, the minimum of the expression $E_{D_t}\{\tilde{g}_t(I_t,\cdot,D_t)+\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))\}$ is attained when x_t is the smallest element in $\mathcal{A}_t(I_t)$ (which is also an element of $W^{-1}(I_t)$). By composition of approximation (Proposition 5.1), $\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))$ is an L'-approximation of $z_{t+1}(f_t(I_t,\cdot,D_t))$. By linearity of approximation and summation of approximation (Proposition 5.1), $E_{D_t}\{\tilde{g}_t(I_t,\cdot,D_t)+\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))\}$ is a $\max\{L',L''\}$-approximation of $E_{D_t}\{g_t(I_t,\cdot,D_t)+z_{t+1}(f_t(I_t,\cdot,D_t))\}$. Hence, $\bar{z}_t(I_t)$ is a $\max\{L',L''\}$-approximation of $z_t(I_t)$. This implies that $\bar{z}_t(I_t)$ is a $K'L'$-approximation of $z_t(I_t)$. In this case, $\bar{z}_t(I_t)$ can be determined in $O(n_t(t_{\tilde{g}_t}+t_{f_t}+t_{\tilde{z}_{t+1}}))$ time.

Case 2. $E_{D_t}\tilde{g}_t(I_t,\cdot,D_t)$ and $E_{D_t}\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))$ are monotone in the opposite direction. In this case, we apply Proposition 5.3 with the following parameter setting. Let $D=\mathcal{S}_t$, $C(\cdot)=\mathcal{A}_t(\cdot)$, $n=2n_t$, $m=n_t$, $x=I_t$, $y=x_t$, $\varphi(\cdot)=z_t(\cdot)$, and $\tilde{\varphi}(\cdot)=\bar{z}_t(\cdot)$. For $i=1,\ldots,n_t$, let $\varphi_i(\cdot)=p_{t,i}z_{t+1}(\cdot)$, $\tilde{\varphi}_i(\cdot)=p_{t,i}\tilde{z}_{t+1}(\cdot)$, $\psi_i(x,y)=f_t(x,y,d_{t,i})$, $K_i=K'$, $L_i=L'$, and $W_i(x)=f_{t,i}^{-1}(x,W)$. For $i=n_t+1,\ldots,2n_t$, let $\varphi_i(\cdot)=p_{t,i-n_t}g_t(I_t,\cdot,d_{t,i-n_t})$, $\tilde{\varphi}_i(\cdot)=p_{t,i-n_t}\tilde{g}_t(I_t,\cdot,d_{t,i-n_t})$, $\psi_i(x,y)=y$, and $L_i=L''$.

Because $\tilde{z}_{t+1}$ is an L'-approximation of z_{t+1}, by linearity of approximation (Proposition 5.1), $\tilde{\varphi}_i(\cdot)$ is an L'-approximation (i.e., L_i-approximation) of $\varphi_i(\cdot)$ for $i=1,\ldots,n_t$. Similarly, $\tilde{\varphi}_i(\cdot)$ is an L''-approximation (i.e., L_i-approximation) of $\varphi_i(\cdot)$ for $i=n_t+1,\ldots,2n_t$. Because $\tilde{g}_t(I_t,\cdot,d_{t,i})$, $\tilde{z}_{t+1}$, and $\psi_i(I_t,\cdot)$ are monotone, the function $\tilde{\varphi}_i(\psi_i(I_t,\cdot))$ is monotone for $i=1,\ldots,2n_t$. In addition, $\tilde{\varphi}_i(\psi_i(I_t,\cdot))$ is monotone in one direction for $i=1,\ldots,n_t$ and is monotone in the other direction for $i=n_t+1,\ldots,2n_t$. Because $f_t(I_t,x_t,d_{t,i})$ is monotone in x_t, and because W is a K'-approximation set of $\tilde{z}_{t+1}$, by composition of approximation sets (Proposition 6.1), $f_{t,i}^{-1}(I_t,W)$ is a K'-approximation set of $\tilde{z}_{t+1}(f_t(I_t,\cdot,d_{t,i}))$ for $i=1,\ldots,n_t$. By linearity of approximation sets (Proposition 6.1), $f_{t,i}^{-1}(I_t,W)$ is a K'-approximation set of $\tilde{\varphi}_i(\psi_i(I_t,\cdot))$ for $i=1,\ldots,n_t$. Thus, by Proposition 5.3, $\tilde{\varphi}$ is a $\max\{K_1L_1,\ldots,K_mL_m,L_{m+1},\ldots,L_n\}$-approximation of φ. Hence, $\bar{z}_t$ is a $\max\{K'L',L''\}$-approximation (i.e., a $K'L'$-approximation) of z_t.

In (7.1), the minimum of the function can be obtained in $|W^{-1}(I_t)|$ steps by scanning all the elements of $W^{-1}(I_t)$. Each of these steps involves n_t queries to $\tilde{g}_t$, f_t, and $\tilde{z}_{t+1}$ and requires $O(n_t(t_{\tilde{g}_t}+t_{f_t}+t_{\tilde{z}_{t+1}}))$ time. ∎

Remark 4. As will be shown in Proposition 8.1, function z_t in the DP formulation (3.3)–(3.4) is guaranteed to be monotone. Therefore, finding a monotone approximation for it makes sense. This is exactly the approach we take in section 8 by using Proposition 4.7.

The next proposition deals with convex DPs and is applied when both g_t and z_{t+1} are guaranteed to be convex. Note that the minimization in (7.2), which is taken over the entire action space $\mathcal{A}_t(I_t)$, is performed efficiently by exploiting the convexity of $E_{D_t}\{\tilde{g}_t(I_t,\cdot,D_t)\}$ and $E_{D_t}\{\tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))\}$.

PROPOSITION 7.2. *Suppose the DP formulation* (3.3)–(3.4) *is convex (so Condition* 3(iii) *is satisfied). Let K', K'', t, and I_t be fixed values, where $K'' \geq K' \geq 1$, $I_t \in \mathcal{S}_t$, and $t \in [1,\dots,T]$. Let $\tilde{g}_t(I_t,\cdot,d_{t,i})$ be a convex K'-approximation function of $g_t(I_t,\cdot,d_{t,i})$ for every $i = 1,\dots,n_t$. Let $\tilde{z}_{t+1}$ be a convex K''-approximation function of z_{t+1}. Let*

$$\bar{z}_t(I_t) = \min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t}\big\{\tilde{g}_t(I_t,x_t,D_t) + \tilde{z}_{t+1}(f_t(I_t,x_t,D_t))\big\}. \tag{7.2}$$

Then, $\bar{z}_t(I_t)$ is a K''-approximation value of $z_t(I_t)$ and can be determined in $O(n_t(t_{\tilde{g}_t} + t_{f_t} + t_{\tilde{z}_{t+1}})\log|\mathcal{A}_t(I_t)|)$ time.

Proof. We apply Proposition 5.2 with the following parameter setting. Let $D = \mathcal{S}_t$, $C(\cdot) = \mathcal{A}_t(\cdot)$, $n = 2n_t$, $x = I_t$, $y = x_t$, and $\varphi(\cdot) = z_t(\cdot)$. For $i = 1,\dots,n_t$, let $\varphi_i(\cdot) = p_{t,i}g_t(I_t,\cdot,d_{t,i})$, $\tilde{\varphi}_i(\cdot) = p_{t,i}\tilde{g}_t(I_t,\cdot,d_{t,i})$, $\psi_i(x,y) = y$, and $K_i = K'$. For $i = n_t+1,\dots,2n_t$, let $\varphi_i(\cdot) = p_{t,i-n_t}z_{t+1}(\cdot)$, $\tilde{\varphi}_i(\cdot) = p_{t,i-n_t}\tilde{z}_{t+1}(\cdot)$, $\psi_i(x,y) = f_t(x,y,d_{t,i-n_t})$, and $K_i = K''$. Because $\tilde{g}_t(I_t,\cdot,d_{t,i})$ is a K'-approximation of $g_t(I_t,\cdot,d_{t,i})$, by linearity of approximation (Proposition 5.1), $\tilde{\varphi}_i(\cdot)$ is a K'-approximation (i.e., K_i-approximation) of $\varphi_i(\cdot)$ for $i = 1,\dots,n_t$. Similarly, $\tilde{\varphi}_i(\cdot)$ is a K''-approximation (i.e., K_i-approximation) of $\varphi_i(\cdot)$ for $i = n_t+1,\dots,2n_t$. Hence, by Proposition 5.2, $\tilde{\varphi}$ is a $\max\{K_1,\dots,K_n\}$-approximation of φ; that is, $\bar{z}_t$ is a K''-approximation of z_t.

As for the computational time, note that for any fixed d_t, function $f_t(I_t,\cdot,d_t)$ is linear with slope in $\{-1,0,1\}$. Thus, $\tilde{z}_{t+1}(f_t(I_t,\cdot,d_t))$ is a convex function. Because a conical combination (i.e., linear combination with nonnegative coefficients) of convex functions is convex, $E_{D_t}\{\tilde{g}_t(I_t,\cdot,D_t) + \tilde{z}_{t+1}(f_t(I_t,\cdot,D_t))\}$ is a convex function, and therefore its minimum can be obtained in $O(\log|\mathcal{A}_t(I_t)|)$ steps by performing a binary search over the contiguous interval $\mathcal{A}_t(I_t)$. Each of these steps involves n_t queries to $\tilde{g}_t$, f_t, and $\tilde{z}_{t+1}$ and requires $O(n_t(t_{\tilde{g}_t} + t_{f_t} + t_{\tilde{z}_{t+1}}))$ time. ∎

Remark 5. As will be shown in Proposition 9.1, function $\bar{z}_t$ in (7.2) is guaranteed to be convex. Therefore, finding a convex approximation for it makes sense. This is exactly the approach we take in section 9.

From Propositions 7.1 and 7.2, we can see that approximating the stochastic DP recursion (3.4) is essentially as hard as approximating the deterministic counterpart of the problem (i.e., when the random variable is constant with probability 1), except for an additional complexity factor of n_t (i.e., the size of the support of the random variable). This situation is substantially different from the determination of an exact solution for the problem. For example, [33] showed that the single-item stochastic inventory control problem with discrete demand is #P-hard (see Appendix A.5), but it is known that the deterministic counterpart of this problem can be solved in polynomial time as a minimum convex cost network flow problem or as a linear program (see [25, section 4]).

8. An FPTAS for monotone DP.

In this section, we develop an FPTAS for nondecreasing DPs. The FPTAS for nonincreasing DPs is analogous. Our FPTAS is summarized in Algorithm 3. (Recall that function IndirectApxSet is summarized in Algorithm 2 in section 4.2.)

We give two remarks on Algorithm 3. The first remark is about step 5. In this step, the function $\bar{z}_t$ is given by (7.1) with the following setting. Function $\tilde{g}_t$ is set equal to g_t. The set $W_t^{-1}(I_t)$ equals $\bigcup_{i=1}^{n_t} f_{t,i}^{-1}(I_t, W)$, where $W = \bar{W}_{t+1} \cup \{\text{next}(x, \mathcal{S}_{t+1}) \mid$

ALGORITHM 3. FPTAS FOR NONDECREASING DP.

1: **Procedure FPTASNondecreasingDP**
2: $K \leftarrow 1 + \frac{\epsilon}{2(T+1)}$, $z_{T+1} \leftarrow g_{T+1}$, and $\bar{W}_{T+1} \leftarrow \mathbf{ApxSet}(z_{T+1}, \mathcal{S}_{T+1}, D^{\min}, K)$
3: Let $\tilde{z}_{T+1}$ be the approximation of z_{T+1} induced by $\bar{W}_{T+1}$
4: **for** $t := T$ **downto** 1 **do**
5: $\bar{W}_t \leftarrow \mathbf{IndirectApxSet}(\bar{z}_t, \mathcal{S}_t, K)$ /* $\bar{z}_t$ *is as defined in* (7.1)*; see details in the text* */
6: Let $\tilde{z}_t$ be the maximal nondecreasing function that is bounded from above by $\bar{z}_t$ on $\bar{W}_t$
7: **end for**

$x \in \bar{W}_{t+1} \setminus \{\mathcal{S}_{t+1}^{\max}\}\}$. ($\bar{W}_{t+1}$ is obtained from the previous iteration of the for-loop.) Note that $\bar{z}_t$ is not necessarily monotone (nor unimodal). Thus, a K-approximation set of it is undefined, and executing function ApxSet over φ cannot always be done efficiently. (Step 5 of Algorithm 1 does not necessarily run in logarithmic time when φ is not unimodal.) Hence, we call function IndirectApxSet instead.

The second remark is about step 6. This step determines the maximal nondecreasing function $\tilde{z}_t$ such that $\tilde{z}_t(x) \leq \bar{z}_t(x)$ for all $x \in \bar{W}_t$. Note that $\tilde{z}_t$ is a nondecreasing step function. Hence, this step can be performed easily as follows. We first set $\tilde{z}_t(\mathcal{S}_t^{\max}) \leftarrow \bar{z}_t(\mathcal{S}_t^{\max})$. Then, we scan $\bar{W}_t$ backward. For every pair of consecutive elements x, y in $\bar{W}_t$ $(x < y)$, we set $\tilde{z}_t(x) \leftarrow \min\{\bar{z}_t(x), \tilde{z}_t(y)\}$.

In order to prove that Algorithm 3 is indeed an FPTAS, it would be essential for certain property to remain invariant throughout the execution of the algorithm. This property is stated in the next proposition.

PROPOSITION 8.1 (monotone invariant). *If Condition* 3(i) *is satisfied, then for every* $t = 1, \ldots, T+1$, *function* z_t *in the DP formulation* (3.3)–(3.4) *is nondecreasing over* $\mathcal{S}_t$.

Proof. We use backward induction. We first consider the base case of $t = T+1$. Because $z_{T+1} \equiv g_{T+1}$ and g_{T+1} is nondecreasing, z_{T+1} is nondecreasing. Now, consider any $t = 1, \ldots, T$, and assume that z_{t+1} is nondecreasing. By condition 3(i), either z_t is nondecreasing or $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \geq I'$. Thus, it suffices to show that if $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \geq I'$, then z_t is nondecreasing. Because $f_t(\cdot, x_t, D_t)$ is nondecreasing, so is the composition function $z_{t+1}(f_t(\cdot, x_t, D_t))$. Because $g_t(\cdot, x_t, D_t)$ is nondecreasing, so is the sum $g_t(\cdot, x_t, D_t) + z_{t+1}(f_t(\cdot, x_t, D_t))$. This implies that $E_{D_t}\{g_t(\cdot, x_t, D_t) + z_{t+1}(f_t(\cdot, x_t, D_t))\}$ is nondecreasing. If $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \geq I'$, then $\mathcal{A}_t(\cdot)$ is nonincreasing (by set containment) over $\mathcal{S}_t$, which implies that the minimization $\min_{x_t \in \mathcal{A}_t(\cdot)} E_{D_t}\{g_t(\cdot, x_t, D_t) + z_{t+1}(f_t(\cdot, x_t, D_t))\}$ is nondecreasing over $\mathcal{S}_t$. Hence, z_t is nondecreasing. ∎

The main result of this section is stated in the next theorem.

THEOREM 8.2 (FPTAS for nondecreasing DP). *Consider a DP that satisfies Conditions* 1, 2, *and* 3(i), *and consider any* $0 < \epsilon < 1$. *For every initial state* I_1, $\tilde{z}_1(I_1)$ *is a* $(1+\epsilon)$*-approximation of the optimal cost* $z^*(I_1)$, *where* $\tilde{z}_1(I_1)$ *is given in step* 6 *in the last iteration of Algorithm* 3. *Moreover, Algorithm* 3 *runs in time polynomial in both* $\frac{1}{\epsilon}$ *and the (binary) input size.*

Proof. We first explain the correctness of Algorithm 3. Note that z_{T+1} is a nonnegative unimodal function whose value is minimized at $D^{\min}$. Thus, the call to function ApxSet in step 2 is valid. Note also that $\bar{z}_t$ is a nonnegative function, and function IndirectApxSet does not require its input function φ to be unimodal. Hence, the call to function IndirectApxSet in step 5 is valid as well.

Next, we prove that Algorithm 3 returns a $(1+\epsilon)$-approximation solution. To do so, we first show by induction that $\tilde{z}_t$ is a nondecreasing K^{T+2-t}-approximation step function of z_t and that $\bar{W}_t$ is a K-approximation set of $\tilde{z}_t$, for every $t = 1, \dots, T+1$. By Propositions 4.5 and 4.6, $\tilde{z}_{T+1}$ is a nondecreasing K-approximation of z_{T+1}, and $\bar{W}_{T+1}$ is a K-approximation set of $\tilde{z}_{T+1}$. Thus, the base case of $t = T+1$ is valid. The induction hypothesis is that $\tilde{z}_{t+1}$ is a nondecreasing K^{T+1-t}-approximation step function of z_{t+1}, and that $\bar{W}_{t+1}$ is a K-approximation set of $\tilde{z}_{t+1}$. We will show that $\tilde{z}_t$ is a nondecreasing K^{T+2-t}-approximation step function of z_t and that $\bar{W}_t$ is a K-approximation set of $\tilde{z}_t$.

The evaluation of $\bar{z}_t$ in step 5 of Algorithm 3 is performed by applying Proposition 7.1 with $K' = L'' = 1$, $L' = K^{T+1-t}$, and $W = \bar{W}_{t+1} \cup \{\mathrm{next}(x, \mathcal{S}_{t+1}) \mid x \in \bar{W}_{t+1} \setminus \{\mathcal{S}_{t+1}^{\max}\}\}$. In this way we get that $\bar{z}_t$ is an (unnecessarily monotone) K^{T+1-t}-approximation of z_t. (*Note:* The approximation of $\tilde{z}_{t+1}$ induced by $\bar{W}_{t+1}$ equals $\tilde{z}_{t+1}$, i.e., $\hat{\tilde{z}}_{t+1} \equiv \tilde{z}_{t+1}$; thus, by property 2 of Proposition 4.5, such a W is indeed a 1-approximation set of $\tilde{z}_{t+1}$.) Note that by the monotone invariant, z_t is a nondecreasing function. Therefore, by applying Proposition 4.7 with $\varphi = \bar{z}_t$, $D = \mathcal{S}_t$, $K = K$, and $L = K^{T+1-t}$ we get that $\bar{W}_t$ is a K-approximation set of $\tilde{z}_t$ and that $\tilde{z}_t$ is a nondecreasing K^{T+2-t}-approximation step function of z_t. This completes the proof by induction, and the result implies that $\tilde{z}_1$ is a K^{T+1}-approximation of z_1. Recall that $K = 1 + \frac{\epsilon}{2(T+1)}$. Hence, $z^*(I_1) \le z_1(I_1) \le \left[1 + \frac{\epsilon}{2(T+1)}\right]^{T+1} z^*(I_1)$. Because the inequality $(1 + \frac{x}{n})^n \le 1 + 2x$ holds for every $0 \le x \le 1$ and $n \in \mathbb{N}$, we have $z^*(I_1) \le z_1(I_1) \le (1+\epsilon) z^*(I_1)$ for any given $0 < \epsilon < 1$.

It remains to prove that the running time of Algorithm 3 is polynomial in both the input size and $\frac{1}{\epsilon}$. From Conditions 1 and 2, $\log U_{\mathcal{S}}$, $\log U_{\mathcal{A}}$, and $\log U_g$ are all polynomially bounded by the input size. For ease of exposition, we assume that the values of $U_{\mathcal{S}}$, $U_{\mathcal{A}}$, and U_g are at least 2 (so that their logarithmic values are at least 1). Clearly, the running time of Algorithm 3 is dominated by the for-loop, which has T iterations. In each iteration, the running time is dominated by the execution of function IndirectApxSet in step 5. By Proposition 4.7, each execution of function IndirectApxSet takes $O(t_{\bar{z}_t}(1 + \log_K(TU_g)) \log U_{\mathcal{S}}) = O(t_{\bar{z}_t} \log_K(TU_g) \log U_{\mathcal{S}})$ time. (Note that the maximum possible value of $\bar{z}_t$ is bounded from above by $K^{T+2-t}(T+2-t)U_g \le 2TU_g$, as $K^{T+1} \le 1 + \epsilon < 2$.) Note that (i) by Proposition 7.1, evaluating $\bar{z}_t$ takes $O(n_t(t_{g_t} + t_{f_t} + t_{\tilde{z}_{t+1}})|W^{-1}(I_t)|)$ time once $W^{-1}(I_t)$ is given; and (ii) by the monotonicity of $f_t(\cdot, \cdot, D_t)$ in its first two variables, the time needed to construct $W^{-1}(I_t)$ is $O(n_t |W| t_{f_t} \log U_{\mathcal{A}})$, once W is given. Thus, $t_{\bar{z}_t} = O\big(n_t(t_{g_t} + t_{f_t} + t_{\tilde{z}_{t+1}})|W^{-1}(I_t)| + n_t |W| t_{f_t} \log U_{\mathcal{A}}\big)$. Note that $|W^{-1}(I_t)| = O(n_t |W|)$. By Proposition 4.7, $|W| = O(1 + \log_K(TU_g)) = O(\log_K(TU_g))$. Note also that $\tilde{z}_{t+1}$ (which is obtained from the previous iteration of the for-loop) can be stored succinctly in a sorted array of size $|\bar{W}_{t+1}|$. Hence, $t_{\tilde{z}_{t+1}} = O(\log |\bar{W}_{t+1}|) = O(\log \log_K(TU_g))$. (Recall that we apply Proposition 7.1 with $|W| \le 2|\bar{W}_{t+1}| - 1$.) This implies that

$$t_{\bar{z}_t} = O\big(n_t^2(t_{g_t} + t_{f_t} + \log\log_K(TU_g)) \log_K(TU_g) + n_t t_{f_t} \log U_{\mathcal{A}} \log_K(TU_g)\big).$$

This in turn implies that each execution of function IndirectApxSet takes $O\big([n_t^2(t_{g_t} + t_{f_t} + \log\log_K(TU_g)) + n_t t_{f_t} \log U_{\mathcal{A}}] \log_K^2(TU_g) \log U_{\mathcal{S}}\big)$ time. Therefore, the running time of the entire algorithm is

$$O\Big(Tn^*\big[n^* t_g + (n^* + \log U_{\mathcal{A}}) t_f + n^* \log\log_K(TU_g)\big] \log_K^2(TU_g) \log U_{\mathcal{S}}\Big),$$

where $n^* = \max_t n_t$, $t_g = \max_t t_{g_t}$, and $t_f = \max_t t_{f_t}$. Because $0 < \epsilon < 1$, we have

$1 < K < 2$. It is easy to check that $O(\log_K(TU_g)) = O(\frac{\log(TU_g)}{K-1})$. Replacing K with $1 + \frac{\epsilon}{2(T+1)}$, we conclude that the running time of the algorithm is

(8.1)
$$O\left(\frac{T^3 n^*}{\epsilon^2}\left[n^* t_g + (n^* + \log U_{\mathcal{A}})t_f + n^* \log\left(\frac{T}{\epsilon}\log(TU_g)\right)\right]\log^2(TU_g)\log U_{\mathcal{S}}\right),$$

which is polynomial in both $\frac{1}{\epsilon}$ and the input size. ∎

Remark 6. The dependency of the running time of the algorithm on T is at most $(T\log T)^3$ and the dependency on ϵ is at most $\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}$. Note that if the transition functions $f_t(\cdot,\cdot,D_t)$, $t = 1,\dots,T$, are given explicitly and are strictly monotone in their second variable, then the construction of each $W^{-1}(I_t)$ can be speeded up to $O(n_t|W|t_{f_t})$ time, and the term $\log U_{\mathcal{A}}$ can be dropped from (8.1).

9. An FPTAS for convex DP.

In this section, we develop an FPTAS for any DP that satisfies Conditions 1, 2, and 3(iii). Our FPTAS is summarized in Algorithm 4. In order to prove that Algorithm 4 is indeed an FPTAS, it would be essential for certain property to remain invariant throughout the execution of the algorithm. This property is stated in the following proposition.

ALGORITHM 4. FPTAS FOR CONVEX DP.

Procedure FPTASConvexDP
$K \leftarrow 1 + \frac{\epsilon}{2(T+1)}$, $x^* \leftarrow \arg\min g_{T+1}$, $W_{T+1} \leftarrow$ **ApxSet**$(g_{T+1}, \mathcal{S}_{T+1}, x^*, K)$
Let $\check{z}_{T+1}$ be the convex extension of g_{T+1} induced by W_{T+1}
for $t := T$ **downto** 1 **do**
  $x^* \leftarrow \arg\min \bar{z}_t$ /* $\bar{z}_t$ *is as defined in* (7.2) *with* $\tilde{g}_t$ *set equal to* g_t */
  $W_t \leftarrow$ **ApxSet**$(\bar{z}_t, \mathcal{S}_t, x^*, K)$
  Let $\check{z}_t$ be the convex extension of $\bar{z}_t$ induced by W_t
end for

PROPOSITION 9.1 (convex invariant). *If Condition* 3(iii) *is satisfied, then function z_t in the DP formulation* (3.3)–(3.4) *is convex over $\mathcal{S}_t$ for every $t = 1,\dots,T+1$, and function $\bar{z}_t$ in* (7.2) *is convex over $\mathcal{S}_t$ for every $t = 1,\dots,T$.*

Proof. We first prove the convexity of function z_t. Our proof follows the lines of, but generalizes, the proof of Proposition 6.1 in [33]. We use backward induction. Consider first the base case of $t = T+1$. Note that $z_{T+1} \equiv g_{T+1}$. By Condition 3(iii), g_{T+1} is convex, and hence z_{T+1} is convex. For any $t = 1,\dots,T$, we assume by induction that z_{t+1} is convex and prove that z_t is also convex. By Condition 3(iii), (3.4) can be rewritten as

$$z_t(I) = E_{D_t}\{g_t^I(I,D_t)\} + \min_{x\in\mathcal{A}_t(I)} E_{D_t}\big\{g_t^x(x,D_t) + u_t(f_t(I,x,D_t)) + z_{t+1}(f_t(I,x_t,D_t))\big\}.$$

Define $q_t(\cdot) = E_{D_t}\{g_t^x(\cdot,D_t)\}$ and $y_{t,I}(\cdot) = E_{D_t}\{u_t(f_t(I,\cdot,D_t)) + z_{t+1}(f_t(I,\cdot,D_t))\}$ for all $I \in \mathcal{S}_t$. Note that q_t and $y_{t,I}$ are univariate functions over the contiguous interval $\mathcal{A}_t(I)$. Because u_t and z_{t+1} are convex functions and f_t is linear in its second variable, functions $u_t(f_t(I,\cdot,D_t))$ and $z_{t+1}(f_t(I,\cdot,D_t))$ are convex. In addition, because a conical combination of convex functions is a convex function, functions q_t, $y_{t,I}$, and $E_{D_t}\{g_t^I(\cdot,D_t)\}$ are convex. Since $E_{D_t}\{g_t^I(\cdot,D_t)\}$ is a convex function, it is sufficient to prove that the function

(9.1)
$$\zeta_t(I) = \min_{x\in\mathcal{A}_t(I)}\big\{q_t(x) + y_{t,I}(x)\big\}$$

is convex. It suffices to show that $2\zeta_t(I) \le \zeta_t(I+1) + \zeta_t(I-1)$ for all $I \in \mathcal{S}_t \setminus \{\mathcal{S}_t^{\min}, \mathcal{S}_t^{\max}\}$. Consider any $I \in \mathcal{S}_t \setminus \{\mathcal{S}_t^{\min}, \mathcal{S}_t^{\max}\}$. Let $x' \in \mathcal{A}_t(I-1)$ such that $\zeta_t(I-1) = q_t(x') + y_{t,I-1}(x')$, and let $x'' \in \mathcal{A}_t(I+1)$ such that $\zeta_t(I+1) = q_t(x'') + y_{t,I+1}(x'')$. Let $x_* = \lfloor \frac{x'+x''}{2} \rfloor$ and $x^* = \lceil \frac{x'+x''}{2} \rceil$. Because $\mathcal{S}_t$ is a contiguous interval, the convexity of q_t implies that $q_t(x_*) + q_t(x^*) \le q_t(x') + q_t(x'')$. We divide the analysis into three different cases depending on the coefficient b of x in the function $f_t(I, x, D)$. Suppose first that $b = 1$. The convexity of $y_{t,I}$ implies that (recall that a is the coefficient of I in the function $f_t(I, x, D)$)

$$y_{t,I}(x_*) + y_{t,I}(x^*) \le y_{t,I}(x'-a) + y_{t,I}(x''+a) = y_{t,I-1}(x') + y_{t,I+1}(x''). \tag{9.2}$$

If $x_*, x^* \in \mathcal{A}_t(I)$, then $\zeta_t(I) \le q_t(x_*) + y_{t,I}(x_*)$ and $\zeta_t(I) \le q_t(x^*) + y_{t,I}(x^*)$, which imply that $2\zeta_t(I) - [\zeta_t(I+1) + \zeta_t(I-1)] \le q_t(x_*) + q_t(x^*) - [q_t(x') + q_t(x'')] + y_{t,I}(x_*) + y_{t,I}(x^*) - [y_{t,I-1}(x') + y_{t,I+1}(x'')] \le 0$, or equivalently, $2\zeta_t(I) \le \zeta_t(I+1) + \zeta_t(I-1)$. Hence, it suffices in this case to prove that $x_*, x^* \in \mathcal{A}_t(I)$. If, on the other hand, $b = -1$, we replace (9.2) with

$$y_{t,I}(x_*) + y_{t,I}(x^*) \le y_{t,I}(x'+a) + y_{t,I}(x''-a) = y_{t,I-1}(x') + y_{t,I+1}(x''),$$

and, again, it suffices to prove that $x_*, x^* \in \mathcal{A}_t(I)$. Last, if $b = 0$, then $y_{t,I}(\cdot)$ is a constant, say, $y_{t,I}$, and we replace (9.1) with

$$\zeta_t(I) = y_{t,I} + \min_{x \in \mathcal{A}_t(I)} \{q_t(x)\},$$

so if $x_*, x^* \in \mathcal{A}_t(I)$, then $\zeta_t(I) \le q_t(x_*) + y_{t,I}$ and $\zeta_t(I) \le q_t(x^*) + y_{t,I}$, $y_{t,I} = y_{t,I-1} + a$, and $y_{t,I} = y_{t,I+1} - a$, which imply that $2\zeta_t(I) - [\zeta_t(I+1) + \zeta_t(I-1)] \le q_t(x_*) + q_t(x^*) - [q_t(x') + q_t(x'')] + 2y_{t,I} - [y_{t,I-1} + y_{t,I+1}] \le 0$, or equivalently, $2\zeta_t(I) \le \zeta_t(I+1) + \zeta_t(I-1)$. Hence, once again, it suffices to prove that $x_*, x^* \in \mathcal{A}_t(I)$.

To prove that $x_*, x^* \in \mathcal{A}_t(I)$ (i.e., $(I, x_*), (I, x^*) \in \mathcal{S}_t \otimes \mathcal{A}_t$), we consider the case where $x' \le x''$. The case where $x' > x''$ follows a similar argument and is omitted. Note that $(I-1, x'), (I+1, x'') \in \mathcal{S}_t \otimes \mathcal{A}_t$ and that $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex. Thus, it suffices to show that both (I, x_*) and (I, x^*) are elements of the *minimal* integrally convex set S that contains $(I-1, x')$ and $(I+1, x'')$. We divide the analysis into four different cases (see Figure 5), as follows.

Case 1. $x'' = x'$. In this case, $x_* = x^* = x' = x''$ and $S = \{(I-1, x'), (I, x'), (I+1, x')\}$. Hence, $(I, x_*), (I, x^*) \in S$.

Case 2. $x'' = x' + 1$. In this case, $x_* = x'$ and $x^* = x''$. Note that the edges of S must have slopes in $\{-\infty, -1, 0, 1, \infty\}$. Hence, $S = \{(I-1, x'), (I, x'), (I, x''), (I+1, x'')\}$, and therefore $(I, x_*), (I, x^*) \in S$.

Case 3. $x'' = x' + 2$. In this case, $x_* = x^* = \frac{x'+x''}{2}$ and $S = \{(I-1, x'), (I, \frac{x'+x''}{2}), (I+1, x'')\}$. Hence, $(I, x_*), (I, x^*) \in S$.

Case 4. $x'' \ge x' + 3$. Because the edges of S have slopes in $\{-\infty, -1, 0, 1, \infty\}$, set S is bounded from below by the line connecting the points $(I-1, x')$ and $(I+1, x'+2)$ and is bounded from above by the line connecting the points $(I-1, x''-2)$ and $(I+1, x'')$. Hence, points (I, x_*) and (I, x^*) must be included in S.

This completes the proof of convexity of function z_t.

To show the convexity of $\bar{z}_t$, we define $q_{t,I}(\cdot) = E_{D_t}\{\tilde{g}_t(I, \cdot, D_t)\}$ and $y_{t,I}(\cdot) = E_{D_t}\{\tilde{z}_{t+1}(f_t(I, \cdot, D_t))\}$ for all $I \in \mathcal{S}_t$. Then, $q_{t,I}$ and $y_{t,I}$ are convex functions, and

$$\bar{z}_t(I) = \min_{x \in \mathcal{A}_t(I)} \{q_{t,I}(x) + y_{t,I}(x)\}.$$

Using the same argument as in the above convexity proof for ζ_t, we get that $\bar{z}_t$ is convex. ∎

It should be noted that the convex invariant does not necessarily hold if we drop from Condition 3(iii) either the requirement that $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex or the requirement that the coefficient of the second variable of f_t is in $\{-1, 0, 1\}$. This is demonstrated in the following two examples:

$\mathcal{S}_t \otimes \mathcal{A}_t$ is not integrally convex. Consider the following single-period example: $T = 1$, $\mathcal{S}_1 = \{0, 1, 2\}$, $\mathcal{S}_2 = \{0, 1\}$, $\mathcal{A}_1(0) = \{0\}$, $\mathcal{A}_1(1) = \{1\}$, $\mathcal{A}_1(2) = \{1\}$, $g_1(I, x, d) = 2x$, $g_2(I) = |I|$, and $f_1(I, x, d) = I - x$. Note that in this example $\mathcal{S}_1 \otimes \mathcal{A}_1$ is depicted in the second diagram of Figure 3, where the bottom-left point is $(0, 0)$, and is not integrally convex. Note also that $z_1(0) = 0$, $z_1(1) = 2$, and $z_1(2) = 3$. Hence, z_1 is not convex.

The coefficient of the second variable of f_t is not in $\{-1, 0, 1\}$. Consider the following single-period example: $T = 1$, $\mathcal{S}_1 = [0, \ldots, 10]$, $\mathcal{S}_2 = [-8, \ldots, 10]$, $\mathcal{A}_1(I) = \{0, 1, 2\}$, $g_1 \equiv 0$, $g_2(I) = |I|$, and $f_1(I, x, d) = I - 4x$. Note that $\mathcal{S}_1 \otimes \mathcal{A}_1$ is integrally convex, as it is the intersection of a rectangle with the integer lattice. The global minima of z_1 are at 0, 4, and 8 (with value 0), while the global maxima are at 2, 6, and 10 (with value 2). Hence, z_1 is not convex.

These two examples show that if one of the major requirements of Condition 3(iii) does not hold, then the objective function z_1 is not necessarily convex. But is it still possible to design an FPTAS for the problem? The following theorem, which is proved in Appendix C, tells us that this is unlikely to happen.

THEOREM 9.2. *A convex DP where either $\mathcal{S}_t \otimes \mathcal{A}_t$ is not integrally convex, or $b \notin \{-1, 0, 1\}$, does not necessarily admit a constant factor approximation unless $P = NP$.*

The main result of this section is stated in the next theorem.

THEOREM 9.3 (FPTAS for convex DP). *Consider a DP that satisfies Conditions 1, 2, and 3(iii), and consider any $0 < \epsilon < 1$. For every initial state I_1, $\breve{z}_1(I_1)$ is a convex $(1 + \epsilon)$-approximation of the optimal cost $z^*(I_1)$, where $\breve{z}_1(I_1)$ is generated from step 7 in the last iteration of Algorithm 4. Moreover, Algorithm 4 runs in time polynomial in both $\frac{1}{\epsilon}$ and the (binary) input size.*

Proof. Note first that the convex invariant (Proposition 9.1) ensures that all the $\bar{z}_t$'s are convex functions. Hence, all calls to function ApxSet in step 6 are valid.

Next, we prove that Algorithm 4 returns a $(1 + \epsilon)$-approximation solution. To do so, we first show by induction that $\breve{z}_t$ is a convex K^{T+2-t}-approximation function of z_t for every $t = 1, \ldots, T + 1$. For the base case of $t = T + 1$, we apply Proposition 6.4 with $K' = K$, $L' = 1$, $\varphi = \tilde{\varphi} = z_{T+1}$, and $W = W_{T+1}$, and we get that $\breve{z}_{T+1}$ is convex K-approximation of z_{T+1}. Thus, the base case is valid. The induction hypothesis is that $\breve{z}_{t+1}$ is a convex K^{T+1-t}-approximation of z_{t+1}. We will show that $\breve{z}_t$ is a convex K^{T+2-t}-approximation of z_t. We apply Proposition 7.2 with $K' = 1$ (since $\tilde{g}_t \equiv g_t$) and $K'' = K^{T+1-t}$ (since $\breve{z}_{t+1}$ is a convex K^{T+1-t}-approximation of z_{t+1}). We get that $\bar{z}_t$ is a K^{T+1-t}-approximation function of z_t. By the convex

FIG. 5. *The four cases in the proof of Proposition 9.1.*

invariant (Proposition 9.1), $\bar{z}_t$ is a convex K^{T+1-t}-approximation of z_t. Applying Proposition 6.4 with $\varphi = z_t$, $\tilde{\varphi} = \bar{z}_t$, $K' = K$, $L' = K^{T+1-t}$, and $W = W_t$ (from step 6 of the algorithm, W_t is a K-approximation set of $\bar{z}_t$), we get that $\breve{z}_t$ is a convex K^{T+2-t}-approximation of z_t. This completes the proof by induction, and the result implies that $\breve{z}_1$ is a convex K^{T+1}-approximation of z_1. Recall that $K = 1 + \frac{\epsilon}{2(T+1)}$. Hence, $z^*(I_1) \le z_1(I_1) \le \left[1 + \frac{\epsilon}{2(T+1)}\right]^{T+1} z^*(I_1)$. Because the inequality $(1 + \frac{x}{n})^n \le 1 + 2x$ holds for every $0 \le x \le 1$ and $n \in \mathbb{N}$, we have $z^*(I_1) \le z_1(I_1) \le (1+\epsilon)z^*(I_1)$ for any given $0 < \epsilon < 1$.

It remains to prove that the running time of Algorithm 4 is polynomial in both the input size and $\frac{1}{\epsilon}$. From Conditions 1 and 2, $\log U_{\mathcal{S}}$, $\log U_{\mathcal{A}}$, and $\log U_g$ are all polynomially bounded by the input size. For ease of exposition, we assume that the values of $U_{\mathcal{S}}$, $U_{\mathcal{A}}$, and U_g are at least 2 (so that their logarithmic values are at least 1). Clearly, the running time of Algorithm 4 is dominated by the for-loop, which has T iterations. In each iteration, the running time of step 7 is dominated by that of steps 5 and 6. As mentioned above, $\bar{z}_t$ is a convex function. Therefore, binary search can be applied over the state space $U_{\mathcal{S}}$ to determine an argmin. Thus, step 5 takes $O(t_{\bar{z}_t} \log U_{\mathcal{S}})$ time. Note that the maximum possible value of $\bar{z}_t$ is bounded from above by $K^{T+2-t}(T+2-t)U_g \le 2TU_g$ (as $K^{T+1} \le 1+\epsilon < 2$). Subsequently, by Proposition 4.6, step 6 takes $O(t_{\bar{z}_t}[1+\log_K(TU_g)]\log U_{\mathcal{S}})$ time. Hence, each iteration of the for-loop takes $O(t_{\bar{z}_t}[1+\log_K(TU_g)]\log U_{\mathcal{S}}) = O(t_{\bar{z}_t}\log_K(TU_g)\log U_{\mathcal{S}})$ time. By Proposition 7.2, each evaluation of $\bar{z}_t(I_t)$ requires $O(n_t(t_{g_t} + t_{f_t} + t_{\breve{z}_{t+1}})\log U_{\mathcal{A}})$ time. Note that $\breve{z}_{t+1}$ can be stored in a sorted array of size no larger than $|W_t|$. Thus, by property 1 of Proposition 4.5 and Proposition 4.6, $\breve{z}_{t+1}$ can be evaluated in $O(\log(1+\log_K(TU_g)))$ time; that is, $t_{\breve{z}_{t+1}} = O(\log\log_K(TU_g))$. This implies that

$$t_{\bar{z}_t} = O\Big(n_t\big[t_{g_t} + t_{f_t} + \log\log_K(TU_g)\big]\log U_{\mathcal{A}}\Big).$$

Therefore, the running time of the entire algorithm is

$$O\Big(Tn^*\big[t_g + t_f + \log\log_K(TU_g)\big]\log_K(TU_g)\log U_{\mathcal{S}}\log U_{\mathcal{A}}\Big),$$

where $n^* = \max_t n_t$, $t_g = \max_t t_{g_t}$, and $t_f = \max_t t_{f_t}$. Because $0 < \epsilon < 1$, we have $1 < K < 2$. It is easy to check that $O(\log_K(TU_g)) = O(\frac{\log(TU_g)}{K-1})$. Replacing K with $1 + \frac{\epsilon}{2(T+1)}$, we conclude that the running time of the algorithm is

$$O\left(\frac{T^2 n^*}{\epsilon}\left[t_g + t_f + \log\left(\frac{T}{\epsilon}\log(TU_g)\right)\right]\log(TU_g)\log U_{\mathcal{S}}\log U_{\mathcal{A}}\right),$$

which is polynomial in both $\frac{1}{\epsilon}$ and the input size. □

Remark 7. The dependency of the running time of the algorithm on T is at most $(T\log T)^2$, and the dependency on ϵ is at most $\frac{1}{\epsilon}\log\frac{1}{\epsilon}$.

10. Extensions. Our framework for designing FPTASs for stochastic DPs can be extended well beyond the results stated in Theorems 3.2 and 3.3. However, there is a trade-off between the level of generalization of a framework for designing FPTASs and the complication of its construction and analysis. The goal of this paper, in this respect, is to develop a "reasonable" sufficient set of conditions that guarantees the existence of an FPTAS and to provide such an FPTAS. The conditions are satisfied by some basic problems in logistics, operations management, economics, and finance, as we have demonstrated with the 10 problems stated in the introduction. On the other hand, it is possible to extend the framework much further at the expense of additional complexity. In this section we show how to extend the framework in a few directions, including maximization problems, random vectors, correlated stochastic events, implicit descriptions of stochastic events, and profit approximation. Some of these extensions are moderately involved and are discussed in detail here. Other extensions, such as those in sections 10.6–10.9, are more involved and will be presented in full in other papers.

10.1. Maximization problems. Not surprisingly, our framework also applies to maximization problems as summarized in Theorem 3.3. We now give some detailed explanation of why this is so. So far, we have considered the one-sided approximation, where for every $K \geq 1$, we construct a function $\tilde{z}$ that K-approximates z, i.e., $z(x) \leq \tilde{z}(x) \leq Kz(x)$, for every x. If one draws the graph of z and $\tilde{z}$, then $\tilde{z}$ lies "above" z. To emphasize this point, we say that $\tilde{z}$ K-approximates z *from above*. For maximization problems, we would like to construct an approximation function $\tilde{z}$ so that the error remains one-sided but is of the other side. In other words, $\tilde{z}$ is a K-approximation of z *from below* if $\frac{z}{K} \leq \tilde{z} \leq z$. Clearly, if $\tilde{z}$ K-approximates z from above, then $\frac{\tilde{z}}{K}$ K-approximates z from below. Similarly, if $\tilde{z}$ K-approximates z from below, then $K\tilde{z}$ K-approximates z from above. It is possible, and rather straightforward, to extend the definitions and results in sections 4–9 to deal with maximization problems.

10.2. Random vectors. Until now we have assumed that $D_1, \ldots, D_T$ are independent one-dimensional random variables. It is not difficult to check that the analysis of our framework remains valid if $D_1, \ldots, D_T$ are nonnegative independent multi-dimensional random variables, i.e., *random vectors*. Extending $D_1, \ldots, D_T$ to random vectors enables our framework to have more applications. Consider, for example, a more general version of the stochastic ordered adaptive knapsack problem described in Appendix A.1, in which not only the volume v_t, but also the profit π_t, is a random variable. In this case, the input includes the probability distribution of $D_t = (v_t, \pi_t)$. (For every t, we allow v_t and π_t to be nonindependent.) The domains of the single-period cost function g_t and the transition function f_t are now four-dimensional, where $g_t(I_t, x_t, v_t, \pi_t) = x_t \pi_t \delta_{v_t \leq I_t}$ and $f_t(I_t, x_t, v_t, \pi_t) = (I_t - x_t v_t)^+$. This example can be further extended to include *random yields*, in which the order of an item for inclusion into the knapsack may not be fulfilled. Let γ_t be a binary random variable, which is equal to 1 when the inclusion of item t into the knapsack is successful, and is equal to 0 otherwise. Then, $D_t = (v_t, \pi_t, \gamma_t)$, and the domains of functions g_t and f_t are five-dimensional, where $g_t(I_t, x_t, v_t, \pi_t, \gamma_t) = \gamma_t x_t \pi_t \delta_{v_t \leq I_t}$ and $f_t(I_t, x_t, v_t, \pi_t, \gamma_t) = (I_t - \gamma_t x_t v_t)^+$.

An example of a binary random process in inventory control theory is given in [63]. *Random yield* models in logistics generalize the supply process in that the *proportion* of the order being executed is a random variable. (See the survey of [80] and the references therein.) Consider, for example, a random yield version of single-item stochastic batch dispatch problem studied in Appendix A.6, in which the single-period cost function g_t and the transition function f_t depend on both the random variable G_t counting the number of units of newly arriving goods and a rational random yield random variable O_t (i.e., the random vector is $D_t = (G_t, O_t)$). In this case, $g_t(I_t, x_t, G_t, O_t) = K_t \delta_{\lfloor O_t x_t \rfloor > 0} + c_t \lfloor O_t x_t \rfloor + h_{t-1} I_t$ and $f_t(I_t, x_t, G_t, O_t) = I_t - \lfloor O_t x_t \rfloor + G_t$.

Another example of random vectors is presented in Appendix A.7 when we deal with single-resource revenue management with stochastic customer arrivals and cancellations.

10.3. Nonindependent random vectors. Dealing with nonindependent random vectors would demand greater attention. Let us consider the well-studied model of "Markov-modulated demand" (or "world-driven demand"); see, for example, [12, 42, 46, 71] and [81, pp. 415–420]. There is an exogenous discrete-time Markov process $W = \{W_t\}$, called the *world*. The distribution of D_t now depends on the current value of W_t. This means that $D_1, \ldots, D_T$ are no longer independent. Random vector D_t is influenced by W_t, and the Markovian dependence among $W_1, \ldots, W_T$ induces dependence in $D_1, \ldots, D_T$. We also allow D_t and the next world state W_{t+1} to be driven by common events, so that D_t and W_{t+1} may be dependent. We assume, however, that these are the only sources of dependence; that is, conditional on W_t, the pair (D_t, W_{t+1}) is independent of all past events. For example, in a generalization of the cash management problem, the world may represent the economy [37]. In a generalization of the single-item dispatch problem in which the goods are ash of fireplaces, the world may represent the weather.

Let the world state consist of the n states $[1, \ldots, n]$ and be represented by a transition probability matrix $(W_{i,j})$. Note that the classical model with independent and identically distributed D_t is a special case of this model with the world being a single state. The basic model presented in section 3 is also a special case of this model with the number of states in the world being $n = T$ and the transition matrix $(W_{i,j})$ being a $T \times T$ stochastic matrix with $w_{i,i+1} = 1$ for $i = 1, \ldots, T-1$, $w_{T,T} = 1$, and $w_{i,j} = 0$ for all other i, j pairs.

In the Markov-modulated demand model, the domain of z_t is $[1, \ldots, n] \times \mathcal{S}_t$. Thus, instead of (3.2), we have

$$z^*(w_1, I_1) = \min_{x_1, \ldots, x_T} E\left\{ g_{T+1}(w_{T+1}, I_{T+1}) + \sum_{t=1}^{T} g_t(W_t, I_t, x_t, D_t) \right\},$$

where the expectation is taken with respect to the mutual probability distribution of W_t and D_t, and $W_1 = w_1$. Instead of (3.3)–(3.4), we have

$$z_{T+1}(w_{T+1}, I_{T+1}) = g_{T+1}(w_{T+1}, I_{T+1})$$

and $z_t(w_t, I_t) =$

$$\min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t | W_t = w_t} \left\{ g_t(w_t, I_t, x_t, D_t) + E_{W_{t+1} | W_t = w_t} z_{t+1}(W_{t+1}, f_t(w_t, I_t, x_t, D_t)) \right\}$$

for $t = 1, \ldots, T$. Instead of (3.5), we have

$$\begin{aligned} &E_{D | W_t = w_t} \big\{ g_t(w_t, I_t, x_t, D_t) + E_{W_{t+1} | W_t = w_t} z_{t+1}(W_{t+1}, f_t(w_t, I_t, x_t, D_t)) \big\} \\ &\quad = \textstyle\sum_{j=1}^{n_{w_t}} p_{w_t, j} \big[g_t(w_t, I_t, x_t, d_{w_t, j}) + \sum_{i=1}^{n} w_{w_t, i} z_{t+1}(i, f_t(w_t, I_t, x_t, d_{w_t, j})) \big]. \end{aligned}$$

For every fixed world-state w_t, we compute K-approximation sets and functions of $z_t(w_t, \cdot)$. Since the world transition probability matrix is given explicitly, the computation will take polynomial time in the (binary) input size.

Our framework can be easily generalized to provide FPTASs for any constant number of Markov-modulated processes, in which each process is modulated by a separate Markov chain. For example, if we have a Markov-modulated demand process W^D and a Markov-modulated supply process W^S, as studied in [27], then the domain of $z_t(w_t^D, w_t^S, I_t)$ becomes three-dimensional, and for every pair of states $(w_t^D, w_t^S) \in [1, \ldots, n^D] \times [1, \ldots, n^S]$, we compute K-approximating sets and functions for it.

It is also possible to handle another case of nonindependence. Here, there is no "world" state, but the random process $\{D_1, \ldots, D_T\}$ forms a Markov chain with transition matrix (P_{ij}). Then, the state of the environment d_t in period t is dependent only on the observed state d_{t-1} in period $t-1$.

Finally, we consider a non-Markov-modulated process, where the world state w_t at time t transitions to the next state by a deterministic transition function $h_t : \mathcal{W} \times \mathcal{S} \times \mathcal{A} \times D \rightarrow \mathcal{W}$ (i.e., being in world state w_t and inventory state I_t, performing action x_t, and having an instantiation d_t of the random variable D_t, the next world state is $h_t(w_t, I_t, x_t, d_t)$). If the number of world states is polynomially bounded by the input size, then this case also admits an FPTAS.

We conclude this subsection by considering monotone/convex DP with general nonindependent random variables.

THEOREM 10.1. *The stochastic ordered adaptive knapsack problem with nonindependent item volumes is APX-hard.*

The proof of Theorem 10.1 is provided in Appendix C. Because we have formulated the stochastic ordered adaptive knapsack problem as a maximization nondecreasing DP in section A.1, we have the following corollary.

COROLLARY 10.2. *The monotone/convex DP framework presented in this paper cannot be extended to deal with general nonindependent random variables unless $P = NP$.*

10.4. Structure of optimal policies. In this paper, we mainly deal with complexity and computational issues of our framework. A natural issue to explore is the structure of optimal and approximate policies for the problems in our framework. We start with two definitions. A continuous real-valued function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^+$ is said to be *V-shaped* in its variable x if it is linear with nonpositive slope for $x < 0$ and is linear with nonnegative slope for $x \geq 0$. A policy is said to be a *limit policy* (r, s), $-\infty \leq r \leq s \leq \infty$, if (i) whenever state I falls below r, it increases the state to r by adding $r - I$ units; (ii) whenever state I exceeds s, it decreases the state to s by removing $I - s$ units; and (iii) it does nothing when state I is between r and s.

PROPOSITION 10.3. *Suppose a given convex DP satisfies the following: The coefficients of function f_t satisfy $|a| = |b|$ for every d, and function g_t can be expressed as $g_t(I_t, x_t, D_t) = v_t(x_t, D_t) + u_t(f_t(I_t, x_t, D_t), D_t)$, where v_t is V-shaped in x_t, and u_t is convex in x_t. Then, this convex DP admits an optimal limit policy (r_t, s_t).*

A proof of this proposition is available in Appendix B. Both the cash management problem described in section A.10 and the single-item stochastic inventory control problem described in section A.5, where procurement/disposal costs are V-shaped and holding cost is convex, satisfy the conditions stipulated in this proposition.

Note that Algorithm 4 can be easily modified to compute an *approximated* limit policy $(\hat{r}_t, \hat{s}_t)$ for the given problem. It is because all *approximated* functions calculated by the algorithm are piecewise-linear convex functions with breakpoints belonging to the approximation sets built during the execution of the algorithm. Hence, the *optimal* policy for these approximated functions is also a limit policy, with the additional property that both policy levels are at breakpoints. Since the algorithm checks all these breakpoints, the x_t values output by the algorithm follow the optimal limit policy for these (approximated) piecewise-linear convex functions, and these quantities will serve as *approximated* quantities for the exact convex functions.

10.5. Nonexact evaluation of cost functions. In our model stated in section 3, we assume that the input data includes an oracle that computes g_t exactly.

We can weaken this assumption by requiring that an FPTAS exists for evaluating g_t, i.e., weakening Condition 2 to the following condition.

Condition 5. *For every $t = 1, \ldots, T+1$, function f_t is either given explicitly (i.e., as an explicit formula) or accessed via oracle calls, and an FPTAS for evaluating g_t is given. Moreover, the values of g_t are nonnegative rational numbers that are polynomially bounded by the (binary) size of the input.*

We use such nonexact evaluation of the cost function in one of the applications of our framework, namely, dynamic capacity expansion (see Appendix A.3).

Assumption 10.4. For every $\Delta \geq 0$ and time period t, there exists a function $\bar{g}_t^{\Delta}$ such that

$$\frac{g_t(I,x,d)}{1+\Delta} \leq \bar{g}_t^{\Delta}(I,x,d) \leq (1+\Delta) g_t(I,x,d)$$

for every $I \in \mathcal{S}_t$, $x \in \mathcal{A}_t(I)$, and $d \in \mathcal{D}_t$, and function $\bar{g}_t^{\Delta}$ can be evaluated in time polynomial in the input size and $1/\Delta$.

Definition 10.5. *Let $K \geq 1$, and let $\varphi : D \to \mathbb{R}^+$ be a real-valued function over a finite set D. We say that $\tilde{\varphi} : D \to \mathbb{R}$ is a* two-sided K-approximation of φ *if $\varphi(x)/K \leq \tilde{\varphi}(x) \leq K\varphi(x)$ for all $x \in D$.*

Assumption 10.4 is equivalent to the statement that for every $K \geq 1$, function g_t has a two-sided K-approximation. The validity of the following proposition is obvious.

Proposition 10.6. *Let $K \geq 1$, and let $\varphi : D \to \mathbb{R}^+$ be a real-valued function over a finite domain D of real numbers. If $\tilde{\varphi} : D \to \mathbb{R}^+$ is a two-sided K-approximation of φ, then $K\tilde{\varphi}$ is a (one-sided) K^2-approximation of φ.*

For a monotone DP (i.e., either Condition 3(i) or Condition 3(ii) is satisfied), suppose an FPTAS for evaluating g_t is given. In order to apply our framework, we need to build a monotone K-approximation function for g_t. We achieve this as follows: Consider the case where $g_t(I, \cdot, d)$ is nondecreasing. (The case where $g_t(I, \cdot, d)$ is nonincreasing is analogous.) Let $I \in \mathcal{S}_t$ and $d \in \mathcal{D}_t$ be fixed. Let $\bar{g}_t^{\sqrt{K}}(I, \cdot, d)$ be a two-sided $\sqrt{K}$-approximation of $g_t(I, \cdot, d)$. Due to Assumption 10.4, such an approximation is available to us. Let $\tilde{g}_t(I, \cdot, d) = \sqrt{K}\bar{g}_t^{\sqrt{K}}(I, \cdot, d)$. By Proposition 10.6, $\tilde{g}_t(I, \cdot, d)$ is a K-approximation of $g(I, \cdot, d)$. Our framework remains valid if we apply Proposition 7.1 in the proof of Theorem 8.2 with $L'' = K$ instead of $L'' = 1$. Hence, we have the following result.

Theorem 10.7. *Every stochastic monotone DP satisfying Conditions* 1 *and* 5 *admits an FPTAS.*

We now turn to convex DPs. Suppose an FPTAS for evaluating g_t is given. If the FPTAS returns values of a convex function, then we say that it is a *convex* FPTAS. In this case, in the proof of Theorem 9.3, we use $\sqrt{K}\bar{g}_t^{\sqrt{K}}(I, \cdot, d)$ as a one-sided convex K-approximation of g_t and apply Proposition 7.2 with $K' = K$ instead of $K' = 1$.

If the FPTAS is not convex, we first convert it into a convex FPTAS and then repeat the arguments explained above. The conversion of a nonconvex FPTAS into a convex FPTAS is quite involved and is reported in a follow-up paper, and the result is summarized as follows.

Theorem 10.8 (see [31]). *A convex function $\varphi : [A, B] \to \mathbb{R}^+$ that cannot be evaluated directly but only via an FPTAS admits a convex FPTAS.*

Hence, we have the following result.

Theorem 10.9. *Every stochastic convex DP satisfying Conditions* 1 *and* 5 *admits an FPTAS.*

Combining Theorems 10.7 and 10.9, we conclude that every stochastic DP satisfying Conditions 1, 3, and 5 admits an FPTAS.

10.6. Multivariate functions. One may ask if Proposition 4.6 can be generalized to multivariate functions. In the special case where function φ is separable (i.e., it is the sum of d univariate functions $\varphi_1, \dots, \varphi_d$, each of which is either monotone or convex), the answer is in the affirmative. To do so, we build a K-approximation set W_i of φ_i and the approximation $\hat{\varphi}_i$ of φ_i induced by W_i for all $i = 1, \dots, d$. We note that due to summation of approximation (property 3 in Proposition 5.1), $\sum_{i=1}^{d} \hat{\varphi}_i$ is a K-approximation function of φ.

The analysis for other types of multivariate functions is more involved. For the ease of exposition, we limit the domain of the function to be for all $[A, \dots, B]^d$. We say that function $\varphi : [A, \dots, B]^d \to \mathbb{R}$ is *nondecreasing* if $\varphi(x_1, \dots, x_d) \geq \varphi(x'_1, \dots, x'_d)$ for every pair of vectors $(x_1, \dots, x_d), (x'_1, \dots, x'_d) \in [A, \dots, B]^d$ that satisfy $x_i \geq x'_i$, for all $i = 1, \dots, d$. We say that φ is *nonincreasing* if $-\varphi$ is nondecreasing. We say that φ is *monotone* if it is either nondecreasing or nonincreasing. Regarding the convexity of discrete functions, as mentioned in section 2.2, different classes of discrete convex functions have been considered [58, 23, 59]. Other known classes of discrete convex functions include convex extensible, separable convex, $\mathrm{L}^\natural$-convex, and $\mathrm{M}^\natural$-convex. See [59, sect. 1.4.5] for a discussion of various classes of multivariate discrete convex functions and the inclusion relationships among them. A function φ is said to be *Miller's discrete convex* if

$$\min\{\varphi(z) \mid z \in N(\alpha x + (1-\alpha)y)\} \leq \alpha\varphi(x) + (1-\alpha)\varphi(y)$$

holds for any $x, y \in [A, \dots, B]^d$ and any $0 \leq \alpha \leq 1$, where $N(t) = \{t' \in \mathbb{Z}^d \mid \|t - t'\|_\infty < 1\}$ for $t \in \mathbb{R}^d$ [58]. The following theorem states an example of a negative approximability result regarding Miller's discrete convex multivariate functions.

THEOREM 10.10 (nonexistence of succinct approximations for multivariate Miller's convex functions [31]). *For any $1 \leq K < 2$, a bivariate monotone discrete convex function in the sense of Miller does not necessarily admit a succinct K-approximation, regardless of the scheme used to represent the function.*

A few open research problems related to multivariate discrete convex functions are discussed in section 11.

10.7. Other recursive structures. The problems that fit into our framework all share the same recursive structure (3.3)–(3.4), and it is possible to extend the framework to other recursive structures. One such possible extension is as follows. We can view the recursion structure (3.3)–(3.4) as "walking" on a directed path from node $T + 1$ backward to node 1, where node t represents the optimal value function in time period t, for $t = T + 1, \dots, 1$. Now, consider optimization problems over other networks such as trees and series-parallel graphs. In these cases, the recursive structure may be different. As an example, we briefly describe here a time-cost tradeoff project scheduling model studied in a follow-up paper [34].

There is a series-parallel project network of n activities in activity-on-arc representation. Denote the activities as $1, \dots, n$. Associated with each activity i is a nonincreasing function $f_i : T_i \to \mathbb{Z}^+$, where $f_i(t_i)$ is the cost incurred when the activity time is t_i, and $T_i = [\underline{t}_i, \dots, \bar{t}_i]$ is the set of all possible time duration of activity i. Here, we assume that all activity times and costs are integer-valued.

Let $\phi(t_1, \dots, t_n)$ denote the total duration of the project (i.e., the length of the longest path in the network) when the time duration of activity i is t_i for

$i = 1, 2, \ldots, n$. Given a deadline d, we are interested in determining $t_1, \ldots, t_n$ so that $\phi(t_1, \ldots, t_n) \le d$ and that $f_1(t_1) + \cdots + f_n(t_n)$ is minimized.

A series-parallel network can be reduced to a single-arc network via a sequence of series and parallel reduction operations. A *series reduction* is an operation that replaces two series arcs by a single arc, while a *parallel reduction* is an operation that replaces two parallel arcs by a single arc. In a project network, a reduction of two series activities with time duration t' and t'' will result in a single activity with time duration $t' + t''$, while a reduction of two parallel activities with time duration t' and t'' will result in a single activity with time duration $\max\{t', t''\}$. Thus, for a given series-parallel project network of n activities, it takes only $n - 1$ series/parallel reduction operations to reduce it to a single-activity network. However, when there are time-cost tradeoff decisions for the activities, the integration of the two time-cost tradeoff functions during a series/parallel reduction operation becomes a challenge if we want to perform the computation efficiently.

First, suppose that we allocate t time units to a pair of *parallel* activities i_1 and i_2; that is, we allow each of these two activities to spend no more than t time units. Then, the merged activity, which has a maximum duration of t, will incur a cost of

$$f_i(t) = f_{i_1}(t) + f_{i_2}(t), \tag{10.1}$$

where $f_{i_1}(t)$ and $f_{i_2}(t)$ are the costs of the original activities i_1 and i_2, respectively.

Next, suppose that we allocate t time units to a pair of *series* activities i_1 and i_2; that is, we allow these two activities to spend no more than a total of t time units. Then, the merged activity i (along the merged arc $u \to w$), which has a duration of t, will incur a cost of

$$f_i(t) = \min_{t'=0,1,\ldots,t} \left\{ f_{i_1}(t') + f_{i_2}(t - t') \right\}, \tag{10.2}$$

where $f_{i_1}(t')$ and $f_{i_2}(t - t')$ are the costs of the original activities i_1 and i_2 if they are allocated t' and $t - t'$ time units, respectively.

Suppose we do not know the exact time-cost trade-off functions f_{i_1} and f_{i_2}, but instead we have a K_1-approximation $\tilde{f}_{i_1}$ for f_{i_1} and a K_2-approximation $\tilde{f}_{i_2}$ for f_{i_2}. Then, summation of approximation (Proposition 5.1) tells us that $\tilde{f}_i(t) = \tilde{f}_{i_1}(t) + \tilde{f}_{i_2}(t)$ is a $\max\{K_1, K_2\}$-approximation of (10.1). In the following proposition, we extend the calculus of K-approximation functions to deal with the recursive structure (10.2) in a way similar to the way Proposition 5.3 deals with the recursive structure (3.3)–(3.4).

PROPOSITION 10.11 (see [34]). *Let f_i be the functions defined in* (10.2). *For $j = 1, 2$, let $K_j \ge 1$, let $\tilde{f}_{i_j}(t)$ be a nonincreasing K_j-approximation function of f_{i_j}, and let W_{i_j} be K_j-approximation set of f_{i_j}. Then,*

$$\bar{f}_i(t) = \min_{t' \in \{0,1,\ldots,t\} \cap (W_{i_1} \cup \{t - x \mid x \in W_{i_2}\})} \left\{ \tilde{f}_{i_1}(t') + \tilde{f}_{i_2}(t - t') \right\}$$

is a $\max\{K_1, K_2\}$-approximation of f_i.

10.8. Different descriptions of stochastic events. One limitation of our framework is that it requires the probability distribution functions of the random variables to be given explicitly. In a follow-up work [35], we relax this requirement in the following way. The input of the stochastic variables is given as oracles for the cumulative distribution functions of each of the random variables, together with bounds on their supports. The FPTAS is required to query these oracles a polynomial number of times. One advantage of this model is that the assumptions are so weak that they encompass various ways of specifying random variables, such as truncated Poisson distribution with an a priori given rate. An example of how the calculus of K-approximation is extended in this setting is given below.

Proposition 10.12 (see [35]). *Let D be a nonnegative integer-valued random variable, and let F be its cumulative distribution function. Let $\xi : [L, \dots, U] \to \mathbb{Z}^+$ be a nondecreasing function. Let $K_1, K_2 \geq 1$, let $\xi(a_0) = 0$, and let $S = \{a_1 < \cdots < a_n\}$ be a K_1-approximation set of ξ. Let $\tilde{F}$ be a K_2-approximation of F. Then,*

$$\tilde{\xi}_1(x) = \sum_{i=1}^{n} \big[\xi(a_i) - \xi(a_{i-1})\big] \tilde{F}(x - a_i)$$

is a K_1K_2-approximation of $E_D[\xi(x - D)]$. Moreover, if $\tilde{F}(\cdot)$ is nondecreasing, then so is $\tilde{\xi}_1(\cdot)$.

Considering the stochastic single-item inventory control problem discussed in the introduction (and formally defined in Appendix A.5), we note that under general lead times, the optimal value function is multivariate. It is well-known that this DP can be transformed into a single-variate DP of the same form as the one presented in section A.5. (The state corresponds to inventory position, which is defined as the inventory on hand plus all outstanding inventory.) It is easy to show that this transformation preserves the approximation ratio, and, as a result, it suffices to find an FPTAS for this single-variate DP. If $L > 0$ is an arbitrary lead time, then the underlying demand distribution of the transformed problem is $\bar{D}_t = \sum_{\hat{t}=t}^{t+L-1} D_{\hat{t}}$. The presented FPTAS requires that we know $\mathrm{Prob}(\bar{D}_t = \bar{d}_{t,i})$, which is a convolution of L distributions. Consequently, computing these probabilities takes $(n^*)^L$ time. If L is 2 or 3 (or any other constant value), then the term $(n^*)^L$ is polynomial, and the algorithm is an FPTAS. If L is not constant (e.g., $L = T/4$), then the running time is exponentially large. In the latter case, our algorithm is not an FPTAS. An open question raised in [33] is whether one can modify the approach and create an FPTAS for the problem in which the lead times are permitted to be a fraction of T.

In [35], we give a positive answer to this question and design an FPTAS in the following way. For $0 \leq j \leq L$ and $1 \leq i \leq T - j$, let F_i^j be the cumulative distribution function of the convolution of $D_i, \dots, D_{i+j}$; that is, $F_i^j(x) = \mathrm{Prob}(D_i + \cdots + D_{i+j} \leq x)$. We compute F_i^j exactly for $j = 0, 1$ and $1 \leq i \leq T - j$. For $2 \leq j \leq L$ and $1 \leq i \leq T - j$, we build a K^{j-1}-approximation function $\tilde{F}_i^j$ for F_i^j via K-approximation sets in a recursive way by using the calculus of K-approximation and the equality

$$F_i^j(x) = P(D_i + \cdots + D_{i+j} \leq x) = \sum_{(y \leq x) \text{ and } (y \in \text{ support of } D_i)} \mathrm{Prob}(D_i = y) F_{i+1}^{j-1}(x - y).$$

(*Note*: Since the cumulative distribution function is monotone, a K-approximation set for it is well-defined.) By using Proposition 10.12 and other tools, [35] derives an FPTAS for this problem.

10.9. Approximating profits. All the problems studied in this paper and in [32, 34] are either for minimizing cost or for maximizing revenue. If one wants to maximize profit, i.e., the difference between revenue and cost, then having a rule in the calculus of K-approximation that deals with subtraction of functions is desirable. Note that such a rule cannot be analogous to "summation of approximation" (Property 3 in Proposition 5.1). This is because while the ratio between $\varphi_1 + \varphi_2$ and $\tilde{\varphi}_1 + \tilde{\varphi}_2$ is bounded, it is not necessarily so between $\varphi_1 - \varphi_2$ and $\tilde{\varphi}_1 - \tilde{\varphi}_2$ (e.g., when φ_1 and φ_2 are very close to each other).

The next proposition shows that by imposing the restriction that $\varphi_2 \leq c\varphi_1$ for some given constant $c > 0$, the aforementioned ratio will be bounded.

PROPOSITION 10.13 (subtraction of approximation from below [35]). *Let $\varphi_i : D \to \mathbb{R}^+$ be a nonnegative function over domain D and let $K_i \geq 1$ be arbitrary, $i = 1, 2$. Let $\tilde{\varphi}_1 : D \to \mathbb{R}^+$ be a K_1-approximation of φ_1 from below and $\tilde{\varphi}_2 : D \to \mathbb{R}^+$ be a K_2-approximation of φ_2 from above. Let $c < \frac{1}{K_1 K_2}$ be an arbitrary positive real number. If $x \in D$ satisfies $\varphi_2(x) \leq c\varphi_1(x)$, then $(\tilde{\varphi}_1 - \tilde{\varphi}_2)(x)$ is a $\frac{(1-c)K_1}{1-cK_1K_2}$-approximation of $(\varphi_1 - \varphi_2)(x)$ from below.*

By using Proposition 10.13, among other ingredients, [35] derives an FPTAS for a certain basic inventory control problem. Please refer to [35] for further details.

11. Concluding remarks and future research. In this paper we have presented a framework for obtaining FPTASs for stochastic monotone or convex DPs. While other recent developments in approximation algorithms for stochastic dynamic and multistage programs are based on gradients or sampling, our framework is based on the notion of approximation sets and functions. Under our framework, standard recurrence recursion is used, but we consider only polynomially many states. Our algorithm relies on either monotonicity or convexity of the value function.

We start with two complexity remarks. First, Alekhnovich et al. present a model for backtracking and DP [2]. They prove several upper and lower bounds on the capabilities of algorithms in their model and show that their model captures the DP framework of [79]. In their paper, they question whether their model could capture other DP algorithms. It would be interesting to explore the capabilities of our framework in this context.

Second, Dyer et al. investigate classes of counting problems that are interreducible under approximation-preserving reductions [19]. One of these classes is the class of counting problems that admit (randomized) FPRASs. It would be interesting to investigate the class of counting problems that admit FPTASs in this context.

It is appropriate to point out some limitations of our approach. It is interesting to consider relaxing any of Conditions 1–3. We have shown that the framework cannot be extended to general nonindependent random events (Corollary 10.2). We have also shown that the condition regarding convex DP (Condition 3(iii)) cannot be relaxed (Theorem 9.2). Regarding Condition 1, it would have been more desirable if we could extend our framework to deal with multivariate DPs, i.e., to allow fixed-dimensional state and action spaces. Following the discussion in section 10.6, our FPTAS framework can be extended to multivariate separable functions. Specifically, if $\mathcal{S}_t, \mathcal{A}_t \subset \mathbb{Z}^d$ for all t, then the transition function f_t and the cost function g_t can be expressed as $f_t(I, x, D) = (f_t^1(I_1, x_1, D), \ldots, f_t^d(I_d, x_d, D))$ and $g_t(I, x, D) = \sum_{i=1}^d g_t^i(I_i, x_i, D)$. In this case, the minimization and the expectation in the DP formulation (3.3)–(3.4) can be split into d separable problems, allowing the solution of the problem to be d unidimensional DPs. Thus, the FPTAS can be applied, provided that Conditions 2 and 3 are satisfied by each g_t^i and f_t^i. On the negative side, Theorem 10.10 tells us that when the cost functions g_t are either monotone or Miller's convex (or both), one cannot have a succinct K-approximation of the cost-to-go functions z_t, so our approach fails. This begs the question of whether other approaches can succeed. For multivariate monotone functions, the answer is in the negative. It is known that the existence of an FPTAS for the two-dimensional 0/1 knapsack problem (which can be formulated as a two-dimensional nondecreasing DP) would imply $P = NP$ (see [48, p. 252] and the references therein).

This leaves open the following interesting future research directions. First, it is interesting to characterize which of the various classes of discrete convex functions admit succinct representation. Second, for these classes, it is desirable to extend our FPTAS framework. The most interesting class of discrete convex functions to study is perhaps that of L$^\natural$-convex functions. Recently, [11] studied fixed-dimensional stochastic DPs in a discrete setting over a finite horizon, under the primary assumption that the cost-to-go functions are discrete $L^\natural$-convex. They proposed a *pseudopolynomial* time approximation scheme that solves multidimensional DPs to within an arbitrary prespecified *additive* error of $\epsilon > 0$. The proposed approximation algorithm is a generalization of the explicit-enumeration algorithm, offers a full control in the trade-off between accuracy and running time, but runs in time pseudopolynomial in the input size. If the class of discrete $L^\natural$-convex functions turns out not to admit efficient succinct approximations, then their result is in a way the best possible. Otherwise, the knowledge of how to construct efficient succinct approximations for $L^\natural$-convex functions may be a first step toward the design of an FPTAS for such DPs.

Despite the above limitations, our framework appears to be generalizable to other problems by using the general framework of the calculus of K-approximation. Indeed, since the calculus of K-approximation is modular, other researchers may choose various building blocks required for a problem at hand and develop an FPTAS by adding more tools as appropriate. This is exactly the approach used in [34, 35].

Appendix A. Applications of the framework.

In this appendix we demonstrate the applications of our framework to various stochastic and deterministic optimization problems.

A.1. Stochastic ordered adaptive knapsack problem.

Consider the following stochastic ordered adaptive knapsack problem as studied in Dean, Goemans, and Vondrák [15]. We are given an ordered set of n items and a knapsack with constant capacity $B \in \mathbb{Z}^+$. Each item t has a constant profit $\pi_t \in \mathbb{Q}^+$. Item t has a volume v_t, which is a random variable having a known probability distribution with support $\mathcal{D}_t$ as described in section 3. We would like to determine sequentially whether an item should be placed into the knapsack. The decision of whether to put item t in the knapsack is made after knowing the remaining capacity of the knapsack resulted from the execution of the previous $t-1$ decisions. The actual volume of item t is unknown until we instantiate the item by attempting to place it in the knapsack. If its volume exceeds the remaining knapsack capacity, then the process will terminate immediately, and the final overflowing item will contribute no profit. The objective is to maximize the expected total profit of those items placed in the knapsack.

Clearly, our problem is a generalization of the following classical (deterministic) 0/1 knapsack problem:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n \pi_i x_i \\ \text{subject to} & \sum_{i=1}^n v_i x_i \le B, \\ & x_i \in \{0,1\}, \qquad i = 1,\dots,n, \end{array} \tag{A.1}$$

where π_i is the unit profit of item i, v_i is the (deterministic) volume of item i, and x_i indicates whether item i is selected. Problem (A.1) is known to be NP-hard [47].

Because our problem is a generalization of problem (A.1) (the latter is a special case of our problem in which every random variable accepts a single value with probability 1), our problem is also NP-hard.

Variants of the stochastic knapsack problem have been studied in the literature such as knapsack problems with deterministic item volumes and random item profits [9, 36, 70, 72]. Another somewhat related variant, known as the stochastic and dynamic knapsack problem, involves items that arrive online according to certain stochastic process [50, 62]. The works [49, 30] have considered a stochastic knapsack problem with "chance" constraints, in which the objective is to find a maximum-profit set of items whose probability of overflowing the knapsack is no greater than some specified value. Dean, Goemans, and Vondrák [15] have studied the stochastic ordered adaptive knapsack problem, in which the item volumes are independent random variables with arbitrary distributions. They have developed a polynomial-time algorithm for the problem. For every $\epsilon > 0$, their algorithm gives a solution in which the value is at least the optimal value, at the expense of a slight loss in terms of feasibility; that is, the total volume of the items placed in the knapsack does not exceed $(1+\epsilon)B$. While valuable, their algorithm is not in the spirit of FPTASs, in which constraints are treated as "hard" and feasibility is always maintained.

The stochastic ordered adaptive knapsack problem can be formulated as a pseudo-polynomial time DP as follows: Let $z_t(I_t)$ be the expected profit when considering only items $t, \ldots, n$, where the remaining available volume in the knapsack is I_t. The recurrence relation is

$$z_t(I_t) = \max\left\{E_{v_t}\{\pi_t \delta_{v_t \le I_t} + z_{t+1}((I_t - v_t)^+)\}, z_{t+1}(I_t)\right\} \tag{A.2}$$

for $I_t = 0, \ldots, B$ and $t = 1, \ldots, n$. The boundary condition is $z_{n+1}(I_{n+1}) = 0$ for $I_{n+1} \ge 0$. The optimal solution value is $z_1(B)$. The first term in the maximization function on the right-hand side of (A.2) is the outcome of attempting to place item t into the knapsack, while the second term is the outcome of not doing so.

In order to show that this DP fits into our framework, we need to reformulate (A.2) as a maximization of a function over an action space. It is easy to see that (A.3) below is equivalent to (A.2) and that it is indeed a maximization over an action space:

$$z_t(I_t) = \max_{x_t = 0,1} E_{v_t}\left\{x_t \pi_t \delta_{v_t \le I_t} + z_{t+1}((I_t - x_t v_t)^+)\right\}. \tag{A.3}$$

Next, we show that the above DP with recurrence relation (A.3) is a maximization nondecreasing DP that fits into our framework; that is, a DP satisfying Conditions 1, 2, and 4(i). (Recall that Condition 4 is the maximization counterpart of Condition 3 for maximization problems.) For simplicity, we assume that $\pi_t > 0$ for $t = 1, \ldots, T$. (Otherwise item t generates no profit and can be ignored.) Define $T = n$, $g_{T+1} \equiv 0$, and $\mathcal{S}_{T+1} = [0, \ldots, B]$. For $t = 1, \ldots, T$, we define $D_t = v_t$, $\mathcal{S}_t = [0, \ldots, B]$, $\mathcal{A}_t(I_t) = \{0, 1\}$, $g_t(I_t, x_t, D_t) = x_t \pi_t \delta_{D_t \le I_t}$, and $f_t(I_t, x_t, D_t) = (I_t - x_t D_t)^+$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(B+1)$. Thus, Condition 1 holds. Because the functions f_t, g_t are given explicitly, Condition 2 also holds. As for Condition 4(i), we notice that for $t = 1, \ldots, T$, both functions g_t and f_t are nondecreasing in I_t, function f_t is nonincreasing in x_t, and function g_t is nondecreasing in x_t. Furthermore, $\mathcal{A}_t(I_t)$ is independent of I_t, which implies that $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \le I'$. Therefore, Condition 4(i) is also satisfied.

A.2. Nonlinear knapsack problem. Consider the following nonlinear knapsack problem with a separable nondecreasing objective function, a separable nondecreasing packing constraint, and integer variables:

$$\begin{array}{llll} & \text{maximize} & \sum_{i=1}^{n} \pi_i(x_i) & \\ \text{(A.4)} & \text{subject to} & \sum_{i=1}^{n} v_i(x_i) \leq B, & \\ & & \ell_i \leq x_i \leq u_i, & i = 1, \ldots, n, \\ & & x_i \in \mathbb{Z}^+, & i = 1, \ldots, n. \end{array}$$

In this formulation, x_i represents the number of units of item i selected, $\pi_i(x_i)$ is the profit generated from these x_i units, $v_i(x_i)$ is the space or weight consumption of these x_i units, $B \geq 0$ is the capacity of the knapsack, $\ell_i \geq 0$ is a lower bound requirement of x_i, and $u_i \geq 0$ is an upper bound requirement of x_i. We assume that $\pi_i : \mathbb{Z}^+ \to \mathbb{Q}^+$ and $v_i : \mathbb{Z}^+ \to \mathbb{Z}^+$ are nondecreasing functions which satisfy $\pi_i(0) = v_i(0) = 0$ and the binary size of any of their values is polynomially bounded by the (binary) input size. The input data for the problem consists of (i) the knapsack size B, (ii) the bounds ℓ_i and u_i (for each $i = 1, \ldots, n$), and (iii) oracles that compute π_i and v_i for each $i = 1, \ldots, n$. Without loss of generality, we assume that $\sum_{i=1}^{n} v_i(\ell_i) \leq B$. (Otherwise the problem would be infeasible, and this can be checked in linear time.) Because problem (A.4) is a generalization of problem (A.1) and because the input size of (A.4) is bounded by a polynomial of the input size of (A.1), problem (A.4) is also NP-hard.

Many versions of the nonlinear knapsack problem have been addressed in the literature, but the setting of our problem is relatively general because functions π_i and v_i are not restricted to be linear, convex, or concave. For recent surveys on nonlinear knapsack models, see [8, 48, 45].

An FPTAS for the 0/1 knapsack problem (and for the integer knapsack problem where x_i can be any nonnegative integer) was first developed by Ibarra and Kim [41]. Lawler has improved the efficiency of Ibarra and Kim's FPTAS and has discussed its extension to the nonlinear case [51]. However, Lawler's approximation scheme is no longer polynomial when it is applied to the nonlinear knapsack problem. Hochbaum has demonstrated that Lawler's approximation scheme is implementable in polynomial time when π_i is concave and v_i is convex for $i = 1, \ldots, n$ [38]. Kameshwaran and Narahari [45] have developed an FPTAS for the case where v_i is linear and π_i is a piecewise-linear monotone function represented explicitly by tuples of breakpoints, slopes, and costs at breakpoints.

Problem (A.4) can be formulated as a pseudopolynomial time DP as follows. Define $\rho_t : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that $\rho_t(y) = \max\{x \mid v_t(x) \leq y\}$ for any $y \in \mathbb{Z}^+$. Function ρ_t is nondecreasing and can be evaluated in logarithmic numbers of oracle calls to $v(\cdot)$. Define $z_t(I_t)$ as the maximum total profit obtained from items $t, \ldots, n$, given that the available knapsack space is I_t. The recurrence relation is

$$z_t(I_t) = \max_{x_t \in [\ell_t, \ldots, \min\{u_t, \rho_t(I_t - \sum_{i=t+1}^{n} v_i(\ell_i))\}]} \left\{ \pi_t(x_t) + z_{t+1}(I_t - v_t(x_t)) \right\}$$

for $\sum_{i=t}^{n} v_i(\ell_i) \leq I_t \leq B - \sum_{i=1}^{t-1} v_i(\ell_i)$ and $t = 1, \ldots, n$. The boundary condition is $z_{n+1}(I_{n+1}) = 0$ for $I_{n+1} \geq 0$. The optimal solution value is $z_1(B)$.

Next, we show that the above is a maximization nondecreasing DP that fits into our framework. Since the problem is deterministic, the discrete random variables in the framework accept one single value with probability 1. For simplicity, in the following discussion we omit the random variables from our functions. Also, for simplicity, we assume that $B > 0$, $u_t > 0$, and $\pi_t \not\equiv 0$ for $t = 1, \ldots, T$. Define $T = n$, $\mathcal{S}_{T+1} = [0, \ldots, B - \sum_{i=1}^{T} v_i(\ell_i)]$, and $g_{T+1} \equiv 0$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [\sum_{i=t}^{n} v_i(\ell_i), \ldots, B - \sum_{i=1}^{t-1} v_i(\ell_i)]$, $\mathcal{A}_t(I_t) = [\ell_t, \ldots, \min\{u_t, \rho_t(I_t - \sum_{i=t+1}^{n} v_i(\ell_i))\}]$, $g_t(I_t, x_t) = \pi_t(x_t)$, and $f_t(I_t, x_t) = I_t - v_t(x_t)$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$, and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\max\{\log u_{\max}, \log B\}$, where $u_{\max} = \max\{u_1, \ldots, u_T\}$. Thus, Condition 1 holds. Because of our assumptions regarding the functions π_i, v_i, Condition 2 also holds. As for Condition 4(i), we notice that for $t = 1, \ldots, T$, both functions g_t and f_t are nondecreasing in I_t, function f_t is nonincreasing in x_t, and function g_t is nondecreasing in x_t. Furthermore, because $\min\{u_t, \rho_t(I_t)\}$ is nondecreasing in I_t, we have $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \leq I'$. Therefore, Condition 4(i) is also satisfied.

Note that using a similar approach, we can also provide an FPTAS for the following nonlinear minimization knapsack problem:

$$
\text{(A.5)} \qquad
\begin{array}{lll}
\text{minimize} & \sum_{i=1}^{n} \pi_i(x_i) & \\
\text{subject to} & \sum_{i=1}^{n} v_i(x_i) \geq B, & \\
& \ell_i \leq x_i \leq u_i, & i = 1, \ldots, n, \\
& x_i \in \mathbb{Z}^+, & i = 1, \ldots, n.
\end{array}
$$

Remark 8. There is an alternative way to develop an FPTAS for problem (A.4). Safer and Orlin have provided an FPTAS to the special case of problem (A.4) where v_i is a linear function [65, pp. 26–29]. We can transform problem (A.4) into Safer and Orlin's model as follows: For $i = 1, \ldots, n$, let $\ell_i' \leftarrow v_i(\ell_i)$, $u_i' \leftarrow v_i(u_i)$, and $\pi_i'(y) = \max\{\pi_i(x) \mid v_i(x) \leq y \text{ and } x \in \mathbb{Z}^+\}$ for all $y \in \mathbb{Z}^+$. Clearly, π_i' is a nondecreasing function. Given any feasible solution $(y_1, \ldots, y_n)$ of Safer and Orlin's model, we can obtain a feasible solution $(x_1, \ldots, x_n)$ to problem (A.4) with $\sum_{i=1}^{n} \pi_i(x_i) = \sum_{i=1}^{n} \pi_i'(y_i)$ by setting $x_i = \max\{x \mid v_i(x) \leq y_i \text{ and } x \in \mathbb{Z}^+\}$ for $i = 1, \ldots, n$. Thus, we can obtain an ϵ-approximation solution to problem (A.4) by transforming it to Safer and Orlin's model and apply their FPTAS.

Remark 9. Chauhan et al. have provided an FPTAS to a "supply scheduling problem" [10]. An alternative way of developing an FPTAS for problem (A.5) is to transform problem (A.5) into the model of Chauhan et al. using the technique presented in the previous remark and then apply their FPTAS.

A.3. Dynamic capacity expansion. Consider the following multiperiod capacity expansion problem in telecommunication network planning [67]: Given a set of transmission technologies $\{1, \ldots, n\}$ such as copper cables of various sizes, optical fiber cables with different bit rates, etc., we would like to determine a combination of sizes of these technologies to be installed in each time period. Our objective is to satisfy a given demand of circuits in each time period of the planning horizon at minimum cost. The problem is formulated as follows:

$$
\text{(A.6)} \qquad
\begin{array}{lll}
\text{minimize} & \sum_{t=1}^{T} \sum_{i=1}^{n} \pi_{t,i}(x_{t,i}) & \\
\text{subject to} & \sum_{j=1}^{t} \sum_{i=1}^{n} v_i x_{j,i} \geq C_t, & t = 1, \ldots, T, \\
& x_{t,i} \in \mathbb{Z}^+, & t = 1, \ldots, T;\ i = 1, \ldots, n.
\end{array}
$$

In this formulation, the planning horizon is divided into T time periods. Variable $x_{t,i}$ is the amount of technology i installed in period t. Parameter v_i is the unit capacity of technology i, where $v_i > 0$. Parameter C_t is the accumulated demand over time periods $1, \ldots, t$; that is, $C_t = \sum_{j=1}^{t} c_j$, where c_j as the added demand requirement (expansion) in period j. We assume that v_i and c_j are integers for $i = 1, \ldots, n$ and $j = 1, \ldots, T$. The quantity $\pi_{t,i}(x_{t,i})$ is the present value of the monetary resources spent on technology i in period t, where $\pi_{t,i} : \mathbb{Z}^+ \to \mathbb{Q}^+$ is a nondecreasing function. The input data for the problem consists of (i) the number of time periods T, (ii) the accumulated demand C_t (for each $t = 1, \ldots, T$), (iii) the number of transmission technologies n, (iv) the unit capacity v_i of technology i (for each $i = 1, \ldots, n$), and (v) an oracle that computes function $\pi_{t,i}$ (for each time period t and technology i). We assume that the binary size of each of the values of $\pi_{t,i}$ is polynomially bounded by the (binary) input size.

Note that when $T = 1$, problem (A.6) becomes a nonlinear minimization knapsack problem, which is NP-hard. (See [48, pp. 412–413] for a discussion of the equivalence between the minimization knapsack problem and the maximization knapsack problem.) Therefore, problem (A.6) is a generalization of the nonlinear minimization knapsack problem and is also NP-hard.

Saniee [67] has studied this multiperiod capacity expansion problem in which the function $\pi_{t,i}$ is of the form $\pi_{t,i}(x_{t,i}) = x_{t,i}\pi_i\gamma^{t-1}$, where π_i is the unit cost of technology i and γ is a constant discount factor ($0 < \gamma < 1$). He has developed a pseudopolynomial time DP algorithm for the problem. In our model we allow a general nondecreasing cost function $\pi_{t,i}$.

In what follows, we develop an FPTAS for problem (A.6) by modifying Saniee's DP and applying our framework. First, we consider a single time period t and let $\Pi_t(X_t)$ be the optimal cost to meet X_t units of demand in that period (assuming that there is no capacity carried over from the previous period). The value of $\Pi_t(X_t)$ is the optimal objective value of the following nonlinear minimization knapsack problem:

$$\text{(A.7)} \qquad \begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \pi_{t,i}(x_{t,i}) \\ \text{subject to} & \sum_{i=1}^{n} v_i x_{t,i} \geq X_t, \\ & x_{t,i} = 0, \ldots, \bar{x}_{t,i}, \qquad i = 1, \ldots, n. \end{array}$$

Here, $\bar{x}_{t,i}$ represents an upper bound on $x_{t,i}$. For example, we may set $\bar{x}_{t,i} = \lceil X_t/v_i \rceil$. Thus, problem (A.7) is an instance of problem (A.5). Clearly, Π_t is a nondecreasing function. Therefore, by Proposition 4.6, developing a K-approximation set of Π_t and the K-approximation function induced by it would require a computational time which is polynomial in the input size of the problem.

Problem (A.6) can be formulated as a pseudopolynomial time DP as follows. Define $z_t(I_t)$ as the minimum total cost to meet the demands of periods $t, \ldots, T$, given that there are already I_t units of accumulated capacity available from period $t-1$ (i.e., $I_t = \sum_{j=1}^{t-1}\sum_{i=1}^{n} v_i x_{j,i}$), for $t = 1, \ldots, T+1$ and $I_t = 0, \ldots, C_T$. The recurrence relation is

$$z_t(I_t) = \min_{X_t=(C_t-I_t)^+,\ldots,C_T} \Big\{ \Pi_t(X_t) + z_{t+1}(\min\{I_t + X_t, C_T\}) \Big\}$$

for $I_t = 0, \ldots, C_T$ and $t = 1, \ldots, T$, where X_t is the increase in capacity in period t. The boundary condition is $z_{T+1}(I_{T+1}) = 0$ for $I_{T+1} \geq 0$. The optimal solution value is $z_1(0)$.

Next, we show that problem (A.6) is a nonincreasing DP that fits into our framework. Since the problem is deterministic, the discrete random variables in the framework accept one single value with probability 1. For simplicity, in the following discussion we omit the random variables from our functions. Define $g_{T+1} \equiv 0$ and $\mathcal{S}_{T+1} = [0, \ldots, C_T]$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [0, \ldots, C_T]$, $\mathcal{A}_t(I_t) = [(C_t - I_t)^+, \ldots, C_T]$, $g_t(I_t, X_t) = \Pi_t(X_t)$, and $f_t(I_t, X_t) = \min\{I_t + X_t, C_T\}$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(C_T + 1)$. Thus, Condition 1 holds. As for Condition 3(ii), we notice that for $t = 1, \ldots, T$, function f_t is nondecreasing in both I_t and X_t, while function g_t is constant in I_t and nondecreasing in X_t. Furthermore, $\mathcal{A}_t(I_t)$ satisfies $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \leq I'$. Therefore, Condition 3(ii) is also satisfied. Regarding Condition 2, while the f_t's are given explicitly, we do not have oracles to compute the various g_t's (i.e., Π_t) exactly. However, as discussed above, we have an FPTAS for g_t. Hence, as explained in section 10.5, our framework can also be applied in this case.

Note that our FPTAS can be easily extended to the case with nonlinear capacities, where the capacity of $x_{t,i}$ unit of technology i in period t is a nondecreasing function $v_i : \mathbb{Z}^+ \to \mathbb{Z}^+$. In addition, it is not difficult to check that the following maximization version of problem (A.6) can be solved in an analogous way:

$$\text{(A.8)} \qquad \begin{array}{lll} \text{maximize} & \sum_{t=1}^{T} \sum_{i=1}^{n} \pi_{t,i}(x_{t,i}) & \\ \text{subject to} & \sum_{j=1}^{t} \sum_{i=1}^{n} v_i x_{j,i} \leq C_t, & t = 1, \ldots, T, \\ & x_{t,i} \in \mathbb{Z}^+, & t = 1, \ldots, T;\ i = 1, \ldots, n. \end{array}$$

Since problem (A.8) is a generalization of the (linear) multiperiod knapsack problem [22], our FPTAS is also applicable to the multiperiod knapsack problem.

A.4. Time-cost trade-off machine scheduling. Consider the following machine scheduling problem: There are one single machine and n jobs $J_1, \ldots, J_n$. Job J_j has a given due date $d_j \in \mathbb{Z}^+$, a late penalty $w_j \in \mathbb{Z}^+$, a "normal" processing time $\bar{p}_j \in \mathbb{Z}^+$, and a nonincreasing resource consumption function $\rho_j : \mathbb{Z}^+ \to \mathbb{Z}^+$ with $\rho_j(x) = 0$ for any $x \geq \bar{p}_j$. The processing time of J_j, denoted as x_j, is a nonnegative integer decision variable, and a cost of $\rho_j(x_j)$ is incurred if x_j is chosen to be less than $\bar{p}_j$. All jobs are available for processing at time 0, and job preemption is not allowed. The objective is to determine the job processing times and to schedule the jobs on the machine so that the total cost, $\sum_{j=1}^{n} [w_j \delta_{C_j > d_j} + \rho_j(x_j)]$, is minimized, where C_j is the completion time of processing of J_j. Here, $\delta_{C_j > d_j} = 1$ if J_j is a late job, and $\delta_{C_j > d_j} = 0$ if J_j is an on-time job. Note that in reality a job processing time x_j cannot be smaller than some lower limit $\underline{p}_j > 0$, no matter how much resources we allocate to the job. In such a case, we define $\rho_j(x_j) = M$ for $x_j < \underline{p}_j$, where M is a large number. The input data consists of (i) the number of jobs n, (ii) the parameters d_j, w_j, and $\bar{p}_j$ for every job $j = 1, \ldots, n$, and (iii) an oracle that computes function ρ_j (for each job j). We assume that the binary size of any value of ρ_j and the binary size of the number M are polynomially bounded by the (binary) input size.

Note that the special case in which all job compressions are prohibitively expensive (denoted as $1||\sum w_j U_j$ in the machine scheduling literature) is already NP-hard [52]. Thus, our problem is also NP-hard. Cheng et al. [13] have considered a special case of this problem in which ρ_j is a linear function. They have converted the special case into a profit maximization problem and developed an FPTAS for it. However, the existence of an FPTAS for the profit maximization problem does not imply the existence of an FPTAS for the original cost minimization problem. We will present an

FPTAS for the original minimization problem and consider a general nonincreasing resource consumption function.

Our problem can be formulated as a pseudopolynomial time DP as follows. First, we renumber the jobs such that $d_1 \geq d_2 \geq \cdots \geq d_n$. Note that there exists an optimal schedule in which all on-time jobs are arranged in nondecreasing order of due dates and all late jobs are scheduled behind the on-time jobs. Hence, it suffices to consider the job list $J_n, \ldots, J_1$, decide which jobs in this list should be designated as late jobs (and are removed from the list), and decide how much resources should be allocated to the on-time jobs (which are retained in the list).

Let $z_t(I_t)$ be the minimum total cost of a partial schedule containing $J_t, \ldots, J_n$, given that the total processing time of the on-time jobs in this partial schedule is no greater than I_t. For notational convenience, we denote $d_{n+1} = 0$. The recurrence relation is

(A.9)

$$z_t(I_t) = \min\left\{ \min_{x_t=0,\ldots,I_t} \left\{ z_{t+1}(\min\{I_t - x_t,\, d_{t+1}\}) + \rho_t(x_t) \right\}, z_{t+1}(\min\{I_t, d_{t+1}\}) + w_t \right\}$$

for $t = 1, \ldots, n$ and $I_t = 0, \ldots, d_t$. Here, "$z_{t+1}(\min\{I_t - x_t,\, d_{t+1}\}) + \rho_t(x_t)$" is the cost of the partial schedule if J_t is made on time and is assigned x_t units of processing time, while "$z_{t+1}(\min\{I_t, d_{t+1}\}) + w_t$" is the cost of the partial schedule if J_t is selected to be a late job. The boundary condition is $z_{n+1}(0) = 0$. The optimal solution value is $z_1(d_1)$. It is easy to see that (A.9) can be rewritten as

$$z_t(I_t) = \min_{x_t=0,\ldots,I_t} \left\{ z_{t+1}(\min\{I_t - x_t,\, d_{t+1}\}) + \min\{\rho_t(x_t), w_t\} \right\}. \tag{A.10}$$

Next, we show that the above DP with recurrence relation (A.10) is a nonincreasing DP that fits into our framework. For simplicity, we assume that $w_{\max} > 0$, where $w_{\max} = \max\{w_1, \ldots, w_n\}$. (Otherwise there is an optimal solution where all jobs are late.) Define $T = n$, $g_{T+1} \equiv 0$, and $\mathcal{S}_{T+1} = \{0\}$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [0, \ldots, d_t]$, $\mathcal{A}_t(I_t) = [0, \ldots, I_t]$, $D_t = d_{t+1}$ with probability 1, $g_t(I_t, x_t, D_t) = \min\{\rho_t(x_t), w_t\}$, and $f_t(I_t, x_t, D_t) = \min\{I_t - x_t,\, D_t\}$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(d_1 + 1)$. Thus, Condition 1 holds. Because of our assumption on the values of ρ_t, we get that Condition 2 also holds. As for Condition 3(ii), we notice that for $t = 1, \ldots, T$, function f_t is nondecreasing in I_t and nonincreasing in x_t, and function g_t is nonincreasing in I_t and x_t. Furthermore, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \leq I'$. Therefore, Condition 3(ii) is also satisfied.

A.5. Single-item stochastic inventory control. Consider the following single-item stochastic inventory control problem with time-varying demand [33]. The planning horizon is divided into T time periods. At the beginning of a time period t, the inventory level I_t is observed, and then a replenishment decision is made. Let $x_t \geq 0$ denote the replenishment quantity in period t. We assume that the replenishment lead time is zero; that is, the x_t units will arrive in period t. After that, a demand D_t of the item occurs, where D_t is a nonnegative random variable. The ending inventory level of period t equals $I_{t+1} = I_t + x_t - D_t$. Backlogging is allowed, which implies that the inventory level I_{t+1} can be negative. If $I_{t+1} > 0$, then a holding cost is charged; if $I_{t+1} < 0$, then a backlogging cost is incurred. For ease of discussion, we refer to both of these components as the holding cost. Thus, our objective is

$$\min_{x_1,\ldots,x_T} E \sum_{t=1}^{T} \Big[c_t(x_t) + h_t(I_t + x_t - D_t) \Big],$$

subject to the system dynamics $I_{t+1} = I_t + x_t - D_t$ for $t = 1, \ldots, T$, where $c_t(x_t)$ is the procurement cost in period t when the order size is x_t, $h_t(y)$ is the holding cost in period t when the inventory level at the end of the period is y (y can be positive, zero, or negative), and the expectation is taken with respect to the joint distribution of the random variables involved. We assume that $I_1 = 0$ (i.e., we start with zero inventory). We assume that the procurement and holding costs are convex for all time periods. The input data for the problem consists of (i) the number of time periods T, (ii) an oracle that computes functions c_t and h_t (for each period t), and (iii) the demand distribution with support $\mathcal{D}_t$ as described in section 3 (for each period t). All demand, procurement, and inventory levels are integral. For every $t = 1, \ldots, T$, functions c_t and h_t are nonnegative integer-valued, and the binary size of any of their values is polynomially bounded by the (binary) input size.

When the demand is deterministic and the cost functions are linear, the problem is reduced to the classical Wagner–Whitin model, which is solvable in polynomial time [76]. However, with general cost functions, the problem becomes computationally intractable (see, e.g., [25]). A number of authors have developed FPTASs for various NP-hard deterministic inventory control problems with time-varying demand [74, 14]. Recently, there has been a growing interest in approximation algorithms for stochastic inventory control problems [53, 54, 55]. However, none of these algorithms is an FPTAS. In a previous work, we showed that the single-item stochastic inventory control problem with discrete demand is #P-hard, and we give an FPTAS for it [33]. While the FPTAS presented in [33] is an ad hoc algorithm tailored to the specific problem studied, we will show in the following how this problem can be fitted into our framework as a convex DP, and therefore the presented FPTAS can be applied.

To formulate the problem as a convex DP, we define $\mathcal{S}_{T+1} = [-D^*, \ldots, D^*]$ and $g_{T+1} \equiv 0$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [-D^*, \ldots, D^*]$, $\mathcal{A}_t(I_t) = [0, \ldots, \min\{D^* - I_t,\, D^*\}]$, $g_t(I_t, x_t, D_t) = c_t(x_t) + h_t(I_t + x_t - D_t)$, and $f_t(I_t, x_t, D_t) = I_t + x_t - D_t$. Then, our problem can be solved as a DP as presented in (3.3)–(3.4). Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of the absolute value of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(D^* + 1)$. Thus, Condition 1 holds. Since the binary size of any of the values of c_t and h_t is polynomially bounded by the (binary) input size, Condition 2 also holds. As for Condition 3(iii), we have $\mathcal{S}_t \otimes \mathcal{A}_t =$

$$\big\{(-D^*, 0), \ldots, (-D^*, D^*); (-D^*+1, 0), \ldots, (-D^*+1, D^*); \ldots; (-1, 0), \ldots, (-1, D^*); (0, 0), \ldots, (0, D^*); (1, 0), \ldots, (1, D^*-1); \ldots; (D^*-1, 0), (D^*-1, 1); (D^*, 0)\big\},$$

which is an integrally convex set. Let $g_t^I \equiv 0$, $g_t^x(\cdot, d) = c_t(\cdot)$, and $u_t = h_t$. Then, function g_t can be expressed as $g_t(I, x, d) = g_t^I(I, d) + g_t^x(x, d) + u_t(f_t(I, x, d))$, where $g_t^I(\cdot, d)$, $g_t^x(\cdot, d)$ and $u_t(\cdot)$ are univariate convex functions. Let $a = 1$, $b = 1$, and $c(d) = -d$. Then, function f_t can be expressed as $f_t(I, x, d) = aI + bx + c(d)$. Therefore, Condition 3(iii) is also satisfied.

It is easy to see that the three variants of the single-item stochastic inventory control problem with discrete demand stated in [33], namely, the capacitated version, the discounted version, and the version which allows disposal at a cost, can be formulated as convex DPs in a similar fashion.

A.6. Single-item stochastic batch dispatch. Consider managing a dispatch station over a finite time horizon with T time periods, where a single vehicle of capacity Q is available to dispatch the goods in each period. At the beginning of each period t, the decision maker has to decide whether to send off the vehicle or not and, if yes, how many units of goods should be dispatched to the vehicle. If the vehicle is dispatched, then a fixed cost of K_t and a variable cost of c_t per unit of goods are incurred. During period t, an amount of goods D_t will arrive, where D_t is a nonnegative random variable with a known probability distribution with support $\mathcal{D}_t$ as described in section 3. The goods left at the dispatch station at the end of period t will incur a per-unit holding cost of h_t. We assume that the cost parameters h_t, c_t, K_t are nonnegative rational values, and the capacity Q is a positive integer. Let I_t be the amount of goods in the dispatch station at the beginning of period t, where I_1 is a given constant. Our objective is

$$\min_{x_1,\dots,x_T} E\sum_{t=1}^{T}\Big[K_t\delta_{x_t>0} + c_t x_t + h_t\cdot(I_t - x_t + D_t)\Big],$$

subject to the system dynamics $I_{t+1} = I_t - x_t + D_t$ for $t = 1,\dots,T$, where x_t is the amount of goods dispatched in period t (which must not exceed I_t), and the expectation is taken with respect to the joint distribution of the random variables involved.

Papadaki and Powell have considered a multiproduct batch dispatch problem [61]. When the number of products equals one, Papadaki and Powell's problem becomes a single-item batch dispatch problem. They consider the case where the fixed costs and per-unit holding costs are time-independent, except that there is a constant discount factor. Here, we analyze a more general setting of the single-product problem with time-varying cost parameters. Neither a provably bounded approximation nor a complexity result is known for this problem. We provide a proof of the following theorem in Appendix C.

THEOREM A.1. *The single-item stochastic batch dispatch problem with time-varying costs is #P-hard.*

Define $z_t(I_t)$ as the optimal total cost incurred in periods $t,\dots,T$, given that the inventory level at the beginning of period t is I_t. Denote $h_0 = 0$. The problem can be formulated as a DP with recurrence relation

$$z_t(I_t) = \min_{x_t=0,\dots,\min\{Q,I_t\}} E_{D_t}\Big\{K_t\delta_{x_t>0} + c_t x_t + h_{t-1}I_t + z_{t+1}(I_t - x_t + D_t)\Big\} \tag{A.11}$$

for $I_t = 0,\dots,I_1 + D^*$ and $t = 1,\dots,T$. The boundary condition is $z_{T+1}(I_{T+1}) = h_T I_{T+1}$ for $I_{T+1} \geq 0$. The optimal solution value is $z_1(I_1)$.

Next, we show that this is a nondecreasing DP that fits into our framework. We define $\mathcal{S}_{T+1} = [0,\dots,I_1 + D^*]$ and $g_{T+1}(I_{T+1}) = h_T I_{T+1}$. For $t = 1,\dots,T$, we define $\mathcal{S}_t = [0,\dots,I_1 + D^*]$, $\mathcal{A}_t(I_t) = [0,\dots,\min\{Q, I_t\}]$, $g_t(I_t, x_t, D_t) = K_t\delta_{x_t>0} + c_t x_t + h_{t-1}I_t$ (for $t = 1,\dots,T$), and $f_t(I_t, x_t, D_t) = I_t - x_t + D_t$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1,\dots,T$ and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(I_1 + D^* + 1)$. Thus, Condition 1 holds. Since the binary size any of the values of g_t is polynomially bounded by the (binary) input size, Condition 2 also holds. As for Condition 3(i), we notice that for $t = 1, \ldots, T$, function f_t is nondecreasing in I_t and nonincreasing in x_t, and function g_t is nondecreasing in I_t and nondecreasing in x_t. Thus, to show that Condition 3(i) holds, it suffices to show by induction that z_t is nondecreasing for every t. Obviously, z_{T+1} is nondecreasing. For $t = 1, \ldots, T$, suppose that z_{t+1} is nondecreasing. Consider any $I \in [1, \ldots, I_1 + D^*]$, and let x_t^* be a minimizer of the right hand side of (A.11) when $I_t = I$. If $x_t^* = 0$, then

$$z_t(I-1) \leq E_{D_t}\{h_{t-1}\cdot(I-1) + z_{t+1}(I-1+D_t)\} \leq E_{D_t}\{h_{t-1}I + z_{t+1}(I+D_t)\} = z_t(I).$$

If $x_t^* \geq 1$, then

$$\begin{aligned} z_t(I-1) &\leq E_{D_t}\Big\{K_t\delta_{x_t^*-1>0} + c_t\cdot(x_t^*-1) + h_{t-1}\cdot(I-1) + z_{t+1}\big((I-1)-(x_t^*-1)+D_t\big)\Big\} \\ &\leq E_{D_t}\Big\{K_t\delta_{x_t^*>0} + c_t x_t^* + h_{t-1}I + z_{t+1}(I - x_t^* + D_t)\Big\} = z_t(I). \end{aligned}$$

Hence, z_t is also nondecreasing. Therefore, Condition 3(i) is satisfied.

We note that our FPTAS can be easily extended to the nonlinear case in which the disposal costs and holding costs are nondecreasing functions.

A.7. Single-resource revenue management. We consider a static setting of single-resource capacity control in revenue management as studied in Talluri and van Ryzin, in which customers arrive in batches, and only a single class can arrive in each time period [73, sect. 2.2]. We present a model that assumes no cancellations or no-shows, no overbookings, and independent customer arrivals. The model presented here does not follow the convention in Talluri and van Ryzin, in which their (pseudo-polynomial time) DP formulation assumes that in each period the action is taken after the demand of that period is revealed. Our framework requires that the action is taken before the demand is realized. (*Note*: The two conventions are mathematically equivalent.)

Let $C \in \mathbb{Z}^+$ be the available capacity. There are T customer classes, where class t has a revenue contribution of $r_t \in \mathbb{Q}^+$ per arrival ($t = 1, \ldots, T$). All customers in class t arrive in period t, and the number of such customers is distributed randomly based on nonnegative random variable D_t with support $\mathcal{D}_t$ as described in section 3. Let the state of the system I_t correspond to the number of bookings accepted up to period $t-1$, and let x_t be an upper bound on the number of accepted bookings in period t. The problem is to find acceptance policies (i.e., $x_1, \ldots, x_T$) to maximize the expected total revenue. To the best of our knowledge, no FPTAS is known for this model. A proof of the following theorem is provided in Appendix C.

THEOREM A.2. *The single-resource revenue management problem is #P-hard.*

Let us define $z_t(I_t)$ as the optimal expected total revenue in periods $t, \ldots, T$, given that I_t bookings have already been made in previous time periods. The problem can be formulated as a DP with recurrence relation

$$z_t(I_t) = \max_{x_t=0,\ldots,C-I_t} E_{D_t}\big\{r_t \min\{x_t, D_t\} + z_{t+1}(I_t + \min\{x_t, D_t\})\big\} \tag{A.12}$$

for $I_t = 0, \ldots, C$. The boundary condition is $z_{T+1}(I_{T+1}) = 0$ for $0 \leq I_{T+1} \leq C$, and the optimal solution value is $z_1(0)$.

We now show that the above DP with recurrence relation (A.12) is a maximization nonincreasing DP that fits into our framework. Define $T = n$, $g_{T+1} \equiv 0$, and $\mathcal{S}_{T+1} = [0, \ldots, C]$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [0, \ldots, C]$, $\mathcal{A}_t(I_t) = [0, \ldots, C - I_t]$, $g_t(I_t, x_t, D_t) = r_t \min\{x_t, D_t\}$, and $f_t(I_t, x_t, D_t) = I_t + \min\{x_t, D_t\}$. Note that $\mathcal{S}_{T+1}$,

$\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(C+1)$. Thus, Condition 1 holds. Because the functions f_t, g_t are given explicitly, Condition 2 also holds. As for Condition 4(ii), we notice that for $t = 1, \ldots, T$, function f_t is nondecreasing in I_t and x_t, and function g_t is nondecreasing in x_t and nonincreasing in I_t. Furthermore, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \geq I'$. Therefore, Condition 4(ii) is also satisfied.

Exogenous cancellations can also be handled by using random vectors as explained in section 10.2. To this end, let W_t be a random variable modeling the number of cancellations in time period t. Then, we have

$$f_t(I_t, x_t, (D_t, W_t)) = I_t - \min\{I_t, W_t\} + \min\{x_t, D_t\},$$

where we assume that in each period cancellations are recorded before sales.

Unfortunately, the addition of overbookings to the model does not fit our framework. Such a modification would require negative terminal costs.

A.8. Lifetime consumption of risky capital. Consider the following problem studied in Phelps [64]. There is an individual who manages her capital in discrete time periods. In each time period, she can consume some of her capital, and utility is derived from her consumption based on an underlying utility function. The remaining capital in the period yields a return at a stochastic rate. In addition, she receives a deterministic income at the end of the period. The problem is to find an optimal consumption strategy which maximizes her expected total utility throughout the time horizon.

To formalize this problem, let T be the number of time periods, and let I_t be the amount of capital on hand at the beginning of period t. In period t, the individual selects an amount of capital x_t to consume. The utility u_t of consuming x_t units is a nonnegative, nondecreasing, and rational-valued concave function. The unconsumed capital, $I_t - x_t$, grows according to an exogenous stochastic process specifying the growth rate and defined by discrete rational random variable D_t with support $\mathcal{D}_t$ as described in section 3. In addition to this stochastic growth, the individual receives an amount $y_t \geq 0$ units of nonwealth income at the end of period t. She would like to maximize her expected total utility in periods $1, \ldots, T$ by making dynamic consumption decisions $x_1, \ldots, x_t$; that is, her objective is

$$\max_{x_1, \ldots, x_T} E \sum_{t=1}^{T} u_t(x_t),$$

subject to the system dynamics

$$I_{t+1} = (1 + D_t)(I_t - x_t) + y_t$$

for $t = 1, \ldots, T$.

Phelps [64] considered a stationary growth rate distribution D, a stationary income y, and a discounted utility function of the form $u_t(x_t) = \alpha^{t-1} u(x_t)$, where $\alpha \in (0, 1]$ is the discount rate and u is a stationary (nonnegative, nondecreasing, and concave) utility function. Here, we consider a more general model with a time-dependent growth rate distribution D_t, a time-varying income y_t, and a general time-dependent utility function u_t that need not be concave. (See [56, 26] for discussions of time-dependent utility functions.)

We assume that the utility function u_t is rational-valued and that the binary size of any of its function values is polynomially bounded by the (binary) input size. We assume that I_1, y_t, and x_t are integer-valued. Recall that $\mathcal{D}_t = \{d_{t,1}, \ldots, d_{t,n_t}\}$. For $t = 1, \ldots, T$ and $i = 1, \ldots, n_t$, because $d_{t,i}$ is rational, we can express it as $d_{t,i} = r_{t,i}/q_{t,i}$, where $r_{t,i}, q_{t,i} \in \mathbb{N}$. The input data for the problem consists of (i) the number of time periods T, (ii) the initial capital I_1, (iii) the income y_t for all t, (iv) $r_{t,i}$ and $q_{t,i}$ for all t and i, and (v) an oracle that computes the utility function u_t (for each time period t). No FPTAS is known for this problem. A proof of the following theorem is provided in Appendix C.

THEOREM A.3. *The problem of lifetime consumption of risky capital is #P-hard when D_t and u_t are time-dependent.*

We now cast our problem as a maximization nondecreasing DP. To convert the problem into a DP with integer state spaces, we let $L_t = \prod_{j=t}^{T} \prod_{i=1}^{n_j} q_{j,i}$ for $t = 1, \ldots, T$ and let $L_{T+1} = 1$. We multiply I_1 and y_t by L_1 (so now I_1 and y_t become multiples of L_1) and restrict the x_t value to be a multiple of L_t (for $t = 1, \ldots, T$). Thus, I_t must be a multiple of L_t for every t. This ensures that $I_2, \ldots, I_{T+1}$ are all integers.

We define $g_{T+1} \equiv 0$. For $t = 1, \ldots, T+1$, we define $\mathcal{S}_t = \{0, L_t, 2L_t, \ldots, (1 + |d_{1,n_1}|)(1 + |d_{2,n_2}|) \cdots (1 + |d_{t-1,n_{t-1}}|)(I_1 + y_1 + \cdots + y_{t-1})\}$. For $t = 1, \ldots, T$, we define $\mathcal{A}_t(I_t) = \{0, L_t, 2L_t, \ldots, I_t\}$, $g_t(I_t, x_t, d_t) = u_t(x_t)$, and $f_t(I_t, x_t, d_t) = (1 + d_t)(I_t - x_t) + y_t$. Let $y_{\max} = \max_{t=1,\ldots,T}\{y_t\}$. Note that the kth largest element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ can be identified in constant time for any $1 \le k \le |\mathcal{A}_t(I_t)|$, $I_t \in \mathcal{S}_t$, and $t = 1, \ldots, T$. Furthermore, the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $T \log(1 + D^*) + \log(I_1 + T y_{\max}) = T \log(1 + D^*) + \log(\tilde{I}_1 + T\tilde{y}_{\max}) + \sum_{j=1}^{T} \sum_{i=1}^{n_j} \log q_{j,i}$, where $\tilde{I}_1$ and $\tilde{y}_{\max}$ are the values of I_1 and $y_{\max}$, respectively, before they are multiplied by L_1. Thus, Condition 1 holds. Note that the assumption on the utility function u_t implies that Condition 2 also holds. As for Condition 4(i), we notice that for $t = 1, \ldots, T$, function f_t is nondecreasing in I_t and nonincreasing in x_t, and function g_t is nondecreasing in I_t and x_t. Furthermore, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \le I'$. Hence, Condition 4(i) is also satisfied.

A.9. Stochastic growth model. This is a variant of "lifetime consumption of risky capital" proposed by Adda and Cooper [1, Chap. 5]. In each time period, a household decides how much of its capital to consume, and utility is derived from its consumption based on an underlying utility function. The rest of the capital can be used to produce output (and generate more capital) via a production process. There is a constant depreciation rate of capital, and there are fluctuations in capital created by random shocks to the production process. The objective is to maximize the expected total utility throughout the time horizon.

Adda and Cooper [1, Chap. 5] considered two versions of this growth model, namely, the deterministic version with no random shocks and the stochastic version in which the random shock is a random variable following a first-order Markov process. They assume that the time horizon is infinite. They use a discounted utility function of the form $u_t(x_t) = \alpha^{t-1} u(x_t)$, where $\alpha \in (0, 1]$ is the discount rate, and u is a stationary, nonnegative, strictly increasing, concave function.

Here, we consider the problem in a slightly different setting. Namely, the time horizon is finite; the utility function u_t is a general time-dependent, nonnegative, nondecreasing function which is not necessarily concave; and the random shock is defined by positive discrete random variable D_t with support $\mathcal{D}_t$ as described in section 3, which is independent of the random shock in other time periods.

To formalize our problem, let T be the number of time periods, Δ be the depreciation rate, I_t be the amount of capital on hand at the beginning of period t, and x_t be the amount of capital consumed in period t. The household's objective is

$$\max_{x_1,\dots,x_T} E\sum_{t=1}^{T} u_t(x_t),$$

and the system dynamics are

$$I_{t+1} = D_t p_t(I_t) + (1-\Delta)I_t - x_t$$

for $t = 1,\dots,T$, where p_t is a nondecreasing concave production function. We assume that in each period t, the realization of D_t takes place after the value of x_t has been decided, and x_t is no greater than $d_{t,1}p_t(I_t) + (1-\Delta)I_t$.

We assume that the utility function u_t is rational-valued and the production function p_t is integer-valued and that the binary size of any of their function values is polynomially bounded by the (binary) input size. (We do not assume concavity of u_t and p_t.) We assume that I_1 and x_t are integer-valued, and Δ is rational-valued. For $t = 1,\dots,T$ and $i = 1,\dots,n_t$, because $d_{t,i}$ is rational, we can express it as $d_{t,i} = r_{t,i}/q_{t,i}$, where $r_{t,i}, q_{t,i} \in \mathbb{N}$. Similarly, we can express Δ as $\Delta = \alpha/\beta$, where $\alpha, \beta \in \mathbb{N}$. The input data for the problem consists of (i) the number of time periods T, (ii) the initial capital I_1, (iii) α and β, (iv) $r_{t,i}$ and $q_{t,i}$ for all t and i, (v) an oracle that computes the utility function u_t (for each time period t), and (vi) an oracle that computes the production function p_t (for each time period t). No FPTAS is known for this problem. A proof of the following theorem is provided in Appendix C.

THEOREM A.4. *The stochastic growth model is #P-hard.*

We now cast our problem as a maximization nondecreasing DP. To convert the problem into a DP with integer state spaces, we let $L_t = \beta^{T-t+1}\prod_{j=t}^{T}\prod_{i=1}^{n_j} q_{j,i}$ for $t = 1,\dots,T$, and let $L_{T+1} = 1$. We multiply I_1 by L_1, and we scale the elements in the domain and range of p_t by a factor of L_t (so now I_1 and $p_t(I_t)$ become multiples of L_1 and L_t, respectively). We also restrict the x_t value to be a multiple of L_t (for $t = 1,\dots,T$). Thus, I_t must be a multiple of L_t for every t. This ensures that $I_2,\dots,I_{T+1}$ are all integers.

We define $g_{T+1} \equiv 0$. For $t = 1,\dots,T+1$, we define the state space recursively as follows: $\mathcal{S}_1 = \{0, L_1, 2L_1, \dots, I_1\}$, and

$$\mathcal{S}_t = \{0, L_t, 2L_t, \dots, d_{1,n_1}d_{2,n_2}\cdots d_{t-1,n_{t-1}}p_{t-1}(\bar{I}_{t-1}) + (1-\Delta)\bar{I}_{t-1}\},$$

where $\bar{I}_{t-1}$ is the largest element in $\mathcal{S}_{t-1}$. For $t = 1,\dots,T$, we define $\mathcal{A}_t(I_t) = \{0, L_t, 2L_t, \dots, d_{t,1}p_t(I_t) + (1-\Delta)I_t\}$, $g_t(I_t, x_t, d_t) = u_t(x_t)$, and $f_t(I_t, x_t, d_t) = d_t p_t(I_t) + (1-\Delta)I_t - x_t$. Note that the kth largest element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ can be identified in constant time for any $1 \le k \le |\mathcal{A}_t(I_t)|$, $I_t \in \mathcal{S}_t$, and $t = 1,\dots,T$. Furthermore, for $t = 2,\dots,T+1$, $\log \bar{I}_t$ is bounded from above by $T\log D^* + \log p_{t-1}(\bar{I}_{t-1}) + \log \bar{I}_{t-1}$. This implies that the logarithm of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by the (binary) input size. Thus, Condition 1 holds. Note that the assumptions on the utility function u_t and the production function p_t implies that Condition 2 also holds. As for Condition 4(i), we notice that for $t = 1,\dots,T$, function f_t is nondecreasing in I_t and nonincreasing in x_t, and function g_t is nondecreasing in I_t and x_t. Furthermore, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ for all $I, I' \in \mathcal{S}_t$ with $I \le I'$. Hence, Condition 4(i) is also satisfied.

A.10. Cash management problem. Consider the following cash management problem stated in Dreyfus and Law (with some changes in notation) [16, pp. 154–155]. A mutual fund would like to decide how much cash it should keep in its bank account for each of the next T time periods. At the beginning of each period, the cash balance can be increased by selling stocks (at a cost of $s \in \mathbb{Q}^+$ per \$1 value of stocks), decreased by buying stocks (at a cost of $b \in \mathbb{Q}^+$ per \$1 value of stocks), or left constant. We assume that the amount of time required to implement the decision is negligible. During the period (after implementing the decision), the mutual fund receives demands for cash from customers redeeming their mutual fund shares and cash inflows from customers buying the mutual fund shares. Let D_t be a discrete integer random variable describing the net amount of cash withdrawal made by customers during period t, where the distribution of D_t is given in section 3. Note that D_t may be positive or negative, where the latter case means that there is a net deposit of cash into the mutual fund. If the cash balance falls below zero during a period, the bank will automatically lend the fund the additional amount. However, the fund must pay the bank an interest charge of $k \in \mathbb{Q}^+$ per \$1 value of loan per period. Conversely, if the fund has a positive cash balance at the end of a period, it will incur a cost of $\ell \in \mathbb{Q}^+$ per \$1 excessive cash per period, since the fund's money could have been invested elsewhere. The given cash balance at the beginning of period 1 is I_1. We assume that there is a constant discount factor $\alpha \in \mathbb{Q}$ $(0 < \alpha \leq 1)$, and there are no terminal costs. We would like to determine the cash balance in each period so as to minimize the expected total discounted cost for operating the fund.

Let I_t be the cash balance at the beginning of period t. Let y_t be the cash balance after the buying/selling of stocks in period t has taken place and before the realization of D_t. Thus, the system dynamics are $I_{t+1} = y_t - D_t$. Let $L_t(y_t)$ denote the single-period cost of having a cash balance of y_t immediately before the realization of D_t. Then, the objective can be written as

$$\min_{y_1,\dots,y_T} E_{D_t} \sum_{t=1}^{T} \alpha^{t-1} \Big[s(y_t - I_t)^+ + b(I_t - y_t)^+ + L_t(y_t) \Big],$$

where

$$E_{D_t} L_t(y_t) = \textstyle\sum_{d_{t,i} \in \mathcal{D}_t |\, d_{t,i} \leq y_t} \ell(y_t - d_{t,i}) p_{t,i} + \sum_{d_{t,i} \in \mathcal{D}_t |\, d_{t,i} > y_t} k(d_{t,i} - y_t) p_{t,i}.$$

Dreyfus and Law have provided a pseudopolynomial time DP formulation for this problem [16, p. 272]. Elton and Gruber [20] have studied a similar problem with continuous and stationary demand distribution. Other classical models closely related to this problem include [21, 78]. Recently, Nascimento and Powell [60] have studied a similar cash balance problem with a fairly general setting, which includes a Markovian demand process and time-dependent costs on positive and negative cash levels. Their model assumes that the per-unit stock selling cost s and the per-unit stock buying cost b are equal. They have proposed a convergent approximate DP algorithm for their problem. To the best of our knowledge, no FPTAS is known for our problem.

Let D^* be the maximal aggregated positive demand change throughout the time horizon, and assume without loss of generality that D^* is greater than or equal to the absolute value of the minimal aggregated negative demand change throughout the time horizon and that $-D^* \leq I_1 \leq D^*$. A DP formulation of our problem is given as follows. (*Note*: this DP formulation is different from the one in Dreyfus and Law [1977, p. 272].) Let $z_t(I_t)$ be the minimum expected total discounted cost for periods

t through T, given that there is a cash balance of I_t at the beginning of period t. Define $c : \mathbb{Z} \to \mathbb{Q}^+$ such that $c(x) = bx$ if $x \geq 0$ and that $c(x) = -sx$ if $x < 0$. Define $h : \mathbb{Z} \to \mathbb{Q}^+$ such that $h(I) = \ell I$ if $I \geq 0$ and that $h(I) = -kI$ if $I < 0$. Note that functions c and h are V-shaped (see the definition in section 10.4) and thus convex. The recurrence relation is

$$z_t(I_t) = \min_{x_t = I_t - D^*, \ldots, I_t + D^*} E_D \left\{ \alpha^{t-1} \Big[c(x_t) + h(I_t - x_t - D_t) \Big] + z_{t+1}(I_t - x_t - D_t) \right\}, \tag{A.13}$$

where x_t represents the amount of cash holding reduction in period t. The boundary condition is $z_{T+1}(I_{T+1}) = 0$ for any $I_{T+1} \in \mathbb{Z}$. The optimal solution value of our problem is $z_1(I_1)$.

Next, we cast problem (A.13) as a convex DP. Define $g_{T+1} \equiv 0$ and $\mathcal{S}_{T+1} = [-D^*, \ldots, D^*]$. For $t = 1, \ldots, T$, we define $\mathcal{S}_t = [-D^*, \ldots, D^*]$, $\mathcal{A}_t(I_t) = [I_t - D^*, \ldots, I_t + D^*]$, $g_t(I_t, x_t, D_t) = \alpha^{t-1}[c(x_t) + h(I_t - x_t - D_t)]$, and $f_t(I_t, x_t, D_t) = I_t - x_t - D_t$. Note that $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ are all contiguous intervals for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$ and that the logarithm of the absolute value of any nonzero element in $\mathcal{S}_{T+1}$, $\mathcal{S}_t$, and $\mathcal{A}_t(I_t)$ is bounded from above by $\log(2D^* + 1)$. Hence, Condition 1 holds. Note also that the functions $c(\cdot)$ and $h(\cdot)$ are given explicitly. Hence, Condition 2 also holds. As for Condition 3(iii), we have $\mathcal{S}_t \otimes \mathcal{A}_t =$

$$\begin{aligned}
&\big\{(-D^*, -2D^*), \ldots, (-D^*, 0); (-D^* + 1, -2D^* + 1), \ldots, (-D^* + 1, 1); \ldots; \\
&\ (-1, -D^* - 1), \ldots, (-1, D^* - 1); (0, -D^*), \ldots, (0, D^*); (1, -D^* + 1), \ldots, \\
&\ (1, D^* + 1); \ldots; (D^* - 1, -1), \ldots, (D^* - 1, 2D^* - 1); (D^*, 0), \ldots, (D^*, 2D^*)\big\},
\end{aligned}$$

which is an integrally convex set. Let $g_t^I \equiv 0$, $g_t^x(\cdot, d) = \alpha^{t-1} c(\cdot)$, and $u_t = \alpha^{t-1} h$. Then, function g_t can be expressed as $g_t(I, x, d) = g_t^I(I, d) + g_t^x(x, d) + u_t(f_t(I, x, d))$, where $g_t^I(\cdot, d)$, $g_t^x(\cdot, d)$ and $u_t(\cdot)$ are univariate convex functions. Let $a = 1$, $b = -1$, and $c(d) = -d$. Then, function f_t can be expressed as $f_t(I, x, d) = aI + bx + c(d)$. Therefore, Condition 3(iii) is also satisfied.

The above analysis implies that our problem can be cast as a convex DP (and thus an FPTAS exists). It remains an open question of whether our problem is NP-hard. It is easy to check that problem (A.13) remains a convex DP if we generalize our model to allow the cost parameters s, b, k, and ℓ to be time-varying (denoted s_t, b_t, k_t, and ℓ_t, respectively). A proof of the following theorem is provided in Appendix C.

THEOREM A.5. *The generalized cash management problem with time-varying cost parameters s_t, b_t, k_t, and ℓ_t is #P-hard.*

Appendix B. Proofs of propositions in sections 4, 5, 6, and 10.

B.1. Proposition 4.5.

Proof. Let $x^* = \arg\min \varphi$. We first consider the case where $x^* = D^{\min}$. In this case, φ is nondecreasing, and function $\hat{\varphi}$ can be rewritten as

$$\hat{\varphi}(x) = \begin{cases} \varphi(x) & \text{if } x \in W, \\ \varphi(\text{next}(x, W)) & \text{otherwise.} \end{cases} \tag{B.1}$$

Note that if $x_W^* \neq x^*$, then φ (as well as $\hat{\varphi}$) is constant over $D \cap [x^*, \ldots, x_W^*]$. Note also that

$$\varphi(x) \leq \varphi(\text{next}(x, W)) \quad \forall x \in D \setminus W. \tag{B.2}$$

To prove property 1, consider any $x \in D$. If $x \in W$, then $\hat{\varphi}(x) = \varphi(x)$, which implies that $\varphi(x) \le \hat{\varphi}(x) \le K\varphi(x)$. If $x \notin W$, then

$$\hat{\varphi}(x) = \max\{\varphi(\mathrm{prev}(x, W)), \varphi(\mathrm{next}(x, W))\} \le K\varphi(x),$$

where the inequality is due to the locality of K-approximation sets. Furthermore, because φ is nondecreasing over W and due to (B.2), we have

$$\hat{\varphi}(x) = \max\{\varphi(\mathrm{prev}(x, W)), \varphi(\mathrm{next}(x, W))\} = \varphi(\mathrm{next}(x, W)) \ge \varphi(x).$$

Thus, $\hat{\varphi}$ is a K-approximation of φ. If φ is stored in a sorted array $\{(x, \varphi(x)) \mid x \in W\}$, then for any $x \in D$, we can determine the value of $\hat{\varphi}(x)$ in $O(\log |W|)$ time using binary search.

To prove property 2, we first show that W is a K-approximation set of $\hat{\varphi}$. Condition 1 of Definition 4.2 is satisfied because φ and $\hat{\varphi}$ share the same domain. Condition 2 is satisfied because $\varphi(x) = \hat{\varphi}(x)$ for all $x \in W$. By Proposition 4.3, condition 3 is also satisfied. Next, we show that $W \cup W^+$ is a 1-approximation set of $\hat{\varphi}$. Denote $W' = W \cup W^+$. (i) Because $D^{\min}, D^{\max} \in W$, we have $D^{\min}, D^{\max} \in W'$. (ii) Consider any $x \in W' \backslash \{D^{\max}\}$ such that $\mathrm{next}(x, D) \notin W'$. Then, $\mathrm{next}(x, D) \notin W^+$, which implies that $x \notin W$. Thus, $y \notin W$ for any $y \in D$ that satisfies $x \le y < \mathrm{next}(x, W)$. By (B.1),

$$\hat{\varphi}(y) = \varphi(\mathrm{next}(x, W)) \quad \forall y \in D \text{ s.t. } x \le y \le \mathrm{next}(x, W). \tag{B.3}$$

Applying (B.3) with $y = x$, we have $\hat{\varphi}(x) = \varphi(\mathrm{next}(x, W))$. Note that $\mathrm{next}(x, W') \le \mathrm{next}(x, W)$. Hence, applying (B.3) with $y = \mathrm{next}(x, W')$, we have $\hat{\varphi}(\mathrm{next}(x, W')) = \varphi(\mathrm{next}(x, W))$. Thus, $\max\{\hat{\varphi}(x), \hat{\varphi}(\mathrm{next}(x, W'))\} = \min\{\hat{\varphi}(x), \hat{\varphi}(\mathrm{next}(x, W'))\}$. Therefore, W' satisfies condition 2 of Definition 4.2. (iii) By Proposition 4.3, W' also satisfies condition 3.

To prove property 3, we consider any $x \in D \setminus \{D^{\max}\}$, we divide the analysis into four different cases, and we apply (B.1) to each case.

Case 1. $x \in W$ and $\mathrm{next}(x, D) \in W$. In this case, $\hat{\varphi}(x) = \varphi(x) \le \varphi(\mathrm{next}(x, D)) = \hat{\varphi}(\mathrm{next}(x, D))$, where the inequality follows from the fact that φ is nondecreasing over W.

Case 2. $x \in W$ and $\mathrm{next}(x, D) \notin W$. In this case, $\hat{\varphi}(x) = \varphi(x) \le \varphi(\mathrm{next}(x, W)) = \varphi(\mathrm{next}(\mathrm{next}(x, D), W)) = \hat{\varphi}(\mathrm{next}(x, D))$, where the inequality follows from the fact that φ is nondecreasing over W.

Case 3. $x \notin W$ and $\mathrm{next}(x, D) \in W$. In this case, $\hat{\varphi}(x) = \varphi(\mathrm{next}(x, W)) = \varphi(\mathrm{next}(x, D)) = \hat{\varphi}(\mathrm{next}(x, D))$.

Case 4. $x \notin W$ and $\mathrm{next}(x, D) \notin W$. In this case, $\hat{\varphi}(x) = \varphi(\mathrm{next}(x, W)) = \varphi(\mathrm{next}(\mathrm{next}(x, D), W)) = \hat{\varphi}(\mathrm{next}(x, D))$.

Combining these four cases, we conclude that $\hat{\varphi}(x) \le \hat{\varphi}(\mathrm{next}(x, D))$. Hence, $\hat{\varphi}$ is nondecreasing over D. Next, suppose that φ is convex over D. Let ψ be the convex extension of $\hat{\varphi}$ induced by W. Because $\hat{\varphi}$ is nondecreasing, ψ is also nondecreasing. Thus, it is minimized at x^*. Note that φ is a convex function whose values are no larger than those of $\hat{\varphi}$, while ψ is the greatest convex function whose values at the elements of W are no larger than those of $\hat{\varphi}$. Hence, $\varphi(x) \le \psi(x)$ for every $x \in D$ and $\psi(x) \le \hat{\varphi}(x)$ for every $x \in W$. This implies that $\psi(x) \le \hat{\varphi}(x)$ for every $x \in D$ (by definition of $\hat{\varphi}$ and the fact that ψ is convex). Because $\hat{\varphi}$ is a K-approximation of φ, we have $\hat{\varphi}(x) \le K\varphi(x)$ for every $x \in D$. Thus, $\varphi(x) \le \psi(x) \le \hat{\varphi}(x) \le K\varphi(x)$ for every $x \in D$. Therefore, ψ is a convex K-approximation of φ. Note that if $x^*_W \neq x^*$, then because $\hat{\varphi}$ is constant over $D \cap [x^*, \ldots, x^*_W]$, so is ψ. Hence, ψ is minimized at x^*_W.

The proof of the case where $x^* = D^{\max}$ is similar and is therefore omitted. For the case where $x^* \notin \{D^{\min}, D^{\max}\}$, we note that φ must be either nonincreasing over the domain $D \cap [D^{\min}, \ldots, x^*_W]$ or nondecreasing over $D \cap [x^*_W, \ldots, D^{\max}]$ (or both). Suppose, without loss of generality, that φ is nonincreasing over $D \cap [D^{\min}, \ldots, x^*_W]$. We then apply the proof of the case "$x^* = D^{\max}$" to the domain $D \cap [D^{\min}, \ldots, x^*_W]$. Consider the domain $D \cap [x^*_W, \ldots, D^{\max}]$. Note that $x^* \geq x^*_W$ and φ is nondecreasing over $W \cap [x^*_W, \ldots, D^{\max}]$. Note also that $\varphi(x) \leq \varphi(\text{next}(x, W))$ for all $x \in (D \cap [x^*_W, \ldots, D^{\max}]) \setminus W$. Thus, (B.1)–(B.2) hold. Since the proof of the case "$x^* = D^{\min}$" relies on the facts that φ is nondecreasing over W and that (B.1)–(B.2) hold (and does not require the monotonicity of φ over D), we can apply it to the domain $D \cap [x^*_W, \ldots, D^{\max}]$. This completes the proof of the proposition. ∎

B.2. Proposition 4.6.

Proof. Let x^* be a minimizer of function φ. We first consider the case where $x^* = D^{\min}$. In this case, φ is nondecreasing, and function $\hat{\varphi}$ can be rewritten as (B.1). Note that when $x^* = D^{\min}$, step 5 of Algorithm 1 is executed in each iteration of the while-loop (and step 6 is not executed). We can perform this step via binary search in $O(\log |D| t_\varphi)$ time. Let $x^{(1)}, \ldots, x^{(k)}$ denote the sequence of x values that are included into W in the while-loop. We observe that $K\varphi(x^{(i+2)}) < \varphi(x^{(i)})$ for $i = 1, \ldots, k-2$. Thus, the while-loop repeats at most $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$ times. Moreover, for $i = 1, \ldots, k-1$, if $x^{(i+1)} \neq \text{prev}(x^{(i)}, D)$, then $K\varphi(x^{(i+1)}) \geq \varphi(x^{(i)})$. Hence, for any $x \in W \setminus \{D^{\max}\}$, if $\text{next}(x, D) \notin W$, then $K\varphi(x) \geq \varphi(\text{next}(x, W))$, or equivalently, $K \min\{\varphi(x), \varphi(\text{next}(x, W))\} \geq \max\{\varphi(x), \varphi(\text{next}(x, W))\}$. Thus, the set W returned by ApxSet satisfies condition 2 of Definition 4.2. Clearly, condition 1 is satisfied. By Proposition 4.3, condition 3 is also satisfied. Therefore, W is a K-approximation set of φ.

The proof of the case where $x^* = D^{\max}$ is similar and is therefore omitted. For the case where $x^* \notin \{D^{\min}, D^{\max}\}$, we note that x^* is included in the set W returned by the algorithm. Therefore we apply the proof of the case "$x^* = D^{\max}$" to the domain $D \cap [D^{\min}, \ldots, x^*]$ and apply the proof of the case "$x^* = D^{\min}$" to the domain $D \cap [x^*, \ldots, D^{\max}]$. This completes the proof of the proposition. ∎

B.3. Proposition 4.7.

Proof. We start by considering the cardinality of $\bar{W}$ and the running time of the algorithm. We execute step 4 of Algorithm 2 as follows. We determine x' by performing binary search over the domain D. In the first iteration of the search, the scope is $D_1 = D \cap \{y \in D \mid y \leq x\}$, and the condition of the while-loop implies that $K\bar{\varphi}(D_1^{\min}) < \bar{\varphi}(x)$. In addition, $K\bar{\varphi}(D_1^{\max}) \geq \bar{\varphi}(x)$ (since $D_1^{\max} = x$). We choose a middle element $m \in D_1$ (i.e., m is the $\lceil |D_1|/2 \rceil$-th largest element of D_1). We consider two different cases.

Case 1. $K\bar{\varphi}(m) < \bar{\varphi}(x)$. In this case, we set $m' \leftarrow \text{next}(m, D_1)$. If $K\bar{\varphi}(m') \geq \bar{\varphi}(x)$, then the search is completed by assigning $x \leftarrow m$. Otherwise, we set the new (reduced) scope of the search to $D_2 = D_1 \cap \{y \in D \mid y \geq m'\}$.

Case 2. $K\bar{\varphi}(m) \geq \bar{\varphi}(x)$. In this case, we set $m' \leftarrow \text{prev}(m, D)$. If $K\bar{\varphi}(m') < \bar{\varphi}(x)$, then the search is completed by assigning $x \leftarrow m'$. Otherwise, we set the new (reduced) scope of the search to $D_2 = D_1 \cap \{y \in D \mid y \leq m'\}$.

In both cases, if the search is not completed, then we get a new scope D_2 which is at most half the size of D_1, and it satisfies $K\bar{\varphi}(D_2^{\min}) < \bar{\varphi}(x)$ and $K\bar{\varphi}(D_2^{\max}) \geq \bar{\varphi}(x)$. We continue the search this way. Clearly, the search can be completed in $O(\log |D|)$

steps. Let $x^{(1)}, \text{next}(x^{(1)}, D), x^{(2)}, \text{next}(x^{(2)}, D), \ldots$ denote the sequence of x values that are included into $\bar{W}$ in the while-loop. We observe that $K\bar{\varphi}(x^{(i+1)}) < \bar{\varphi}(x^{(i)})$ for $i = 1, 2, \ldots$. Thus, the while-loop repeats at most $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$ times and therefore $|\bar{W}| = O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$. The computational time required in each iteration of the loop is $O(t_{\bar{\varphi}} \log |D|)$, so the claimed running time of the algorithm follows. We note in passing that

$$\bar{\varphi}(\text{next}(x, W)) \leq K\bar{\varphi}(x) \quad \forall x \in W \setminus \{D^{\max}\} \text{ such that } \text{next}(x, D) \notin W. \tag{B.4}$$

We now prove that $\bar{W}$ is a K-approximation set of $\tilde{\varphi}$. Consider any $x \in \bar{W}_t \setminus \{D^{\max}\}$ such that $\text{next}(x, D) \notin \bar{W}$. We divide the analysis into two cases.

Case 1. $\tilde{\varphi}(x) < \bar{\varphi}(x)$. In this case, $\tilde{\varphi}$ differs from $\bar{\varphi}$ on x. Since $\tilde{\varphi}$ is a maximal nondecreasing function bounded from above by $\bar{\varphi}$ over $\bar{W}$, we have $\tilde{\varphi}(x) = \tilde{\varphi}(\text{next}(x, \bar{W}))$.

Case 2. $\tilde{\varphi}(x) = \bar{\varphi}(x)$. In this case, applying (B.4), we have $\bar{\varphi}(\text{next}(x, \bar{W})) \leq K\bar{\varphi}(x) = K\tilde{\varphi}(x)$. By definition of $\tilde{\varphi}$, we have $\tilde{\varphi}(\text{next}(x, \bar{W})) \leq \bar{\varphi}(\text{next}(x, \bar{W}))$. Thus, $\tilde{\varphi}(\text{next}(x, \bar{W})) \leq K\tilde{\varphi}(x)$.

Combining Cases 1 and 2, we conclude that $\tilde{\varphi}(\text{next}(x, \bar{W})) \leq K\tilde{\varphi}(x)$ for any $x \in \bar{W} \setminus \{D^{\max}\}$ that satisfies $\text{next}(x, D) \notin \bar{W}$. Note that $D^{\min}, D^{\max} \in \bar{W}$. Hence, by Proposition 4.3, $\bar{W}$ is a K-approximation set of $\tilde{\varphi}$.

We next prove that $\tilde{\varphi}$ is a nondecreasing KL-approximation step function of φ. By definition of $\tilde{\varphi}$, we have $\tilde{\varphi}(x) \leq \bar{\varphi}(x)$ for any $x \in \bar{W}$. This, together with the fact that $\bar{\varphi}$ is an L-approximation of φ, implies that

$$\tilde{\varphi}(x) \leq L\varphi(x) \quad \forall x \in \bar{W}.$$

On the other hand, for any $x \in \bar{W}$, there exists $y \in \bar{W}$ such that $y \geq x$ and $\tilde{\varphi}(x) = \bar{\varphi}(y)$. Because $\bar{\varphi}$ is an L-approximation of φ, we have $\bar{\varphi}(y) \geq \varphi(y)$. Thus,

$$\tilde{\varphi}(x) = \bar{\varphi}(y) \geq \varphi(y) \geq \varphi(x) \quad \forall x \in \bar{W},$$

where the second inequality is due to the monotonicity of φ. Hence, $\tilde{\varphi}$ is a nondecreasing L-approximation step function of the restriction of φ over $\bar{W}$. Applying approximation of approximation sets (Proposition 6.2) with $\varphi_1 = \tilde{\varphi}$, $\varphi_2 = \varphi$, $K_1 = K$, $K_2 = L$, and $W_1 = \bar{W}$, we get that $\hat{\tilde{\varphi}}$ is a nondecreasing KL-approximation step function of φ. Note that since $\tilde{\varphi}$ is a nondecreasing step function, we get that $\hat{\tilde{\varphi}} \equiv \tilde{\varphi}$. Therefore, $\tilde{\varphi}$ is a nondecreasing KL-approximation step function of φ. ∎

B.4. Proposition 5.2.

Proof. By summation of approximation and composition of approximation (i.e., properties 3 and 4 of Proposition 5.1), $\sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i)$ is a $\max\{K_1, \ldots, K_n\}$-approximation function of $\sum_{i=1}^{n} \varphi_i(\psi_i)$. Consider any fixed $x \in D$. The quantity $\sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y))$ is a $\max\{K_1, \ldots, K_n\}$-approximation value of $\sum_{i=1}^{n} \varphi_i(\psi_i(x, y))$ for all $y \in C(x)$. By minimization of approximation (i.e., property 6 of Proposition 5.1), this implies that $\tilde{\varphi}(x)$ is a $\max\{K_1, \ldots, K_n\}$-approximation value of $\varphi(x)$. Hence, $\tilde{\varphi}$ is a $\max\{K_1, \ldots, K_n\}$-approximation function of φ. ∎

B.5. Proposition 5.3.

Proof. Consider any fixed $x \in D$. Let $y^{@} \in \bigcup_{i=1}^{m} W_i(x)$ be a realizer of $\tilde{\varphi}(x)$; that is, $\tilde{\varphi}(x) = \sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y^{@}))$. By composition of approximation (Proposition 5.1), $\tilde{\varphi}_i(\psi_i(x, \cdot))$ is an L_i-approximation of $\varphi_i(\psi_i(x, \cdot))$ for $i = 1, \ldots, n$, and therefore,

$$\varphi(x) = \min_{y \in C(x)} \left\{ \sum_{i=1}^{n} \varphi_i(\psi_i(x, y)) \right\} \leq \sum_{i=1}^{n} \varphi_i(\psi_i(x, y^{@})) \leq \sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y^{@})) = \tilde{\varphi}(x).$$

Let y^* be the smallest realizer of $\varphi(x)$.

We first consider the scenario where $m \neq n$ and divide the analysis into two cases.

Case 1. $\tilde{\varphi}_1(\psi_1(x,\cdot))$ is nondecreasing, and $\tilde{\varphi}_n(\psi_n(x,\cdot))$ is nonincreasing. In this case, for $i = 1,\dots,m$, define $y'_i = y^*$ if $y^* \in W_i(x)$, and define $y'_i = \text{next}(y^*, W_i(x))$ if $y^* \notin W_i(x)$. Because $W_i(x)$ is a K_i-approximation set of $\tilde{\varphi}_i(\psi_i(x,\cdot))$, we have $\tilde{\varphi}_i(\psi_i(x,y'_i)) \leq K_i\tilde{\varphi}_i(\psi_i(x,y^*))$ for $i = 1,\dots,m$. Let $y' = \min_{i=1,\dots,m}\{y'_i\}$. For $i = 1,\dots,m$, since $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is nondecreasing, we have

$$\tilde{\varphi}_i(\psi_i(x,y')) \leq \tilde{\varphi}_i(\psi_i(x,y'_i)) \leq K_i\tilde{\varphi}_i(\psi_i(x,y^*)) \leq K_iL_i\varphi_i(\psi_i(x,y^*)), \tag{B.5}$$

where the last inequality follows from the fact that $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is an L_i-approximation of $\varphi_i(\psi_i(x,\cdot))$. For $i = m+1,\dots,n$, since $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is nonincreasing and $y' \geq y^*$, we have $\tilde{\varphi}_i(\psi_i(x,y')) \leq \tilde{\varphi}_i(\psi_i(x,y^*))$. This, together with the fact that $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is an L_i-approximation of $\varphi_i(\psi_i(x,\cdot))$, gives us

$$\tilde{\varphi}_i(\psi_i(x,y')) \leq L_i\varphi_i(\psi_i(x,y^*)) \tag{B.6}$$

for $i = m+1,\dots,n$. Because $y' \in \bigcup_{i=1}^m W_i(x)$, we have

$$\tilde{\varphi}(x) \leq \sum_{i=1}^n \tilde{\varphi}_i(\psi_i(x,y')). \tag{B.7}$$

From (B.5), (B.6), and (B.7), we have

$$\begin{aligned}\tilde{\varphi}(x) &\leq \sum_{i=1}^m \tilde{\varphi}_i(\psi_i(x,y')) + \sum_{i=m+1}^n \tilde{\varphi}_i(\psi_i(x,y')) \\ &\leq \sum_{i=1}^m K_iL_i\varphi_i(\psi_i(x,y^*)) + \sum_{i=m+1}^n L_i\varphi_i(\psi_i(x,y^*)) \\ &\leq \max\{K_1L_1,\dots,K_mL_m,L_{m+1},\dots,L_n\}\varphi(x).\end{aligned} \tag{B.8}$$

Case 2. $\tilde{\varphi}_1(\psi_1(x,\cdot))$ is nonincreasing, and $\tilde{\varphi}_n(\psi_n(x,\cdot))$ is nondecreasing. In this case, for $i = 1,\dots,m$, define $y'_i = y^*$ if $y^* \in W_i(x)$, and define $y'_i = \text{prev}(y^*, W_i(x))$ if $y^* \notin W_i(x)$. Letting $y' = \max_{i=1,\dots,m}\{y'_i\} \leq y^*$ and following the same argument as in Case 1, it is easy to verify that inequality (B.8) holds.

Next, we consider the scenario where $m = n$. If $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is nondecreasing for $i = 1,\dots,n$, then inequality (B.5) holds for $i = 1,\dots,n$, and therefore (B.8) remains valid. On the other hand, if $\tilde{\varphi}_i(\psi_i(x,\cdot))$ is nonincreasing for $i = 1,\dots,n$, then inequality (B.6) holds for $i = 1,\dots,n$, which implies that $\tilde{\varphi}(x) \leq \sum_{i=1}^n \tilde{\varphi}_i(\psi_i(x,y')) \leq \sum_{i=1}^n L_i\varphi_i(\psi_i(x,y^*)) \leq \max\{L_1,\dots,L_n\}\varphi_i(x) \leq \max\{K_1L_1,\dots,K_nL_n\}\varphi_i(x)$. This completes the proof of the proposition. ∎

B.6. Proposition 6.1.

Proof. Properties 1 and 4 follow directly from the definitions of K-approximation sets and functions, and their proofs are therefore omitted. A proof for monotonicity of approximation sets, composition of approximation sets, and maximization of approximation sets is given below.

Monotonicity of approximation sets. Clearly, $D^{\min}, D^{\max} \in W_1 \subseteq W'$. Thus, the first condition in Definition 4.2 is satisfied.

To prove the boundedness of W', consider any $x \in W' \setminus \{D^{\max}\}$ such that $\text{next}(x,D) \notin W'$. We will show that

$$\max\{\varphi_1(x), \varphi_1(\text{next}(x,W'))\} \leq K_1 \min\{\varphi_1(x), \varphi_1(\text{next}(x,W'))\}. \tag{B.9}$$

Note that $\mathrm{prev}(x, W_1) \le x < \mathrm{next}(x, W') \le \mathrm{next}(x, W_1)$. (If $x = D^{\min}$ we slightly abuse notation by defining $\mathrm{prev}(D^{\min}, W_1) = D^{\min}$.) Hence, by the unimodality of φ_1,

$$\max\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W'))\} \le \max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\}. \tag{B.10}$$

We consider four different cases.

Case 1. $x \in W_1$ and $\mathrm{next}(x, W') \in W_1$. In this case, $\mathrm{next}(x, W') = \mathrm{next}(x, W_1)$. Hence, (B.9) holds due to the boundedness of W_1.

Case 2. $x \notin W_1$ and $\mathrm{next}(x, W') \in W_1$. In this case, by the locality of W_1, we have $\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(x)$. In addition, because $\mathrm{prev}(x, W_1) \in W_1 \setminus \{D^{\max}\}$ and $\mathrm{next}(\mathrm{prev}(x, W_1), D) \notin W_1$, by the boundedness of W_1, we have $\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(\mathrm{next}(x, W_1))$. Hence,

$$\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1 \min\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W_1))\}.$$

This, together with (B.10) and the fact that $\mathrm{next}(x, W') = \mathrm{next}(x, W_1)$, implies the validity of (B.9).

Case 3. $x \notin W_1$ and $\mathrm{next}(x, W') \notin W_1$. By the locality of W_1, we have $\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(x)$. Because $\mathrm{next}(x, W') \in D \setminus W_1$, by the locality of W_1, we have

$$\max\{\varphi_1(\mathrm{prev}(\mathrm{next}(x, W'), W_1)), \varphi_1(\mathrm{next}(\mathrm{next}(x, W'), W_1))\} \le K_1\varphi_1(\mathrm{next}(x, W'));$$

that is, $\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(\mathrm{next}(x, W'))$. Hence,

$$\max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1 \min\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W'))\}.$$

This, together with (B.10), implies the validity of (B.9).

Case 4. $x \in W_1$ and $\mathrm{next}(x, W') \notin W_1$. In this case, $\mathrm{next}(x, D) \notin W_1$. Thus, by the boundedness of W_1, we have $\max\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(x)$. Because $\mathrm{next}(x, W') \in D \setminus W_1$, by the locality of W_1, we have $\max\{\varphi_1(\mathrm{prev}(\mathrm{next}(x, W'), W_1)), \varphi_1(\mathrm{next}(\mathrm{next}(x, W'), W_1))\} \le K_1\varphi_1(\mathrm{next}(x, W'))$; that is, $\max\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1\varphi_1(\mathrm{next}(x, W'))$. Hence,

$$\max\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W_1))\} \le K_1 \min\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W'))\}.$$

This, together with the fact that $\varphi_1(\mathrm{next}(x, W')) \le \max\{\varphi_1(x), \varphi_1(\mathrm{next}(x, W_1))\}$ (by the unimodality of φ_1), implies the validity of (B.9).

To show the locality of W', consider any $x \in D \setminus W'$. We have

$$\begin{aligned}\max\{\varphi_1(\mathrm{prev}(x, W')), \varphi_1(\mathrm{next}(x, W'))\} &\le \max\{\varphi_1(\mathrm{prev}(x, W_1)), \varphi_1(\mathrm{next}(x, W_1))\}\\ &\le K_1\varphi_1(x),\end{aligned}$$

where the first inequality is due to the unimodality of φ_1 and the second inequality is due to the locality of W_1. This completes the proof of the monotonicity of approximation sets.

Composition of approximation sets. We give a proof for the case where ψ is nondecreasing. The proof for the case with nonincreasing ψ is similar. Note that $\varphi_1(\psi)$ is a unimodal function, and hence a K-approximation set of it is well-defined. Because $\psi(x) \ge D^{\min}$ for all $x \in D'$, we have $\min\{x \mid \psi(x) \ge D^{\min}\} = D'^{\min}$. This, together with the fact that $D^{\min} \in W_1$, implies that $D'^{\min} \in \psi^{-1}(W_1)$. Because $\psi(x) \le D^{\max}$ for all $x \in D'$, we have $\max\{x \mid \psi(x) \le D^{\max}\} = D'^{\max}$. This, together with the fact that $D^{\max} \in W_1$, implies that $D'^{\max} \in \psi^{-1}(W_1)$. Therefore, the first condition in Definition 4.2 is satisfied.

To prove the boundedness of $\psi^{-1}(W_1)$, consider any $x \in \psi^{-1}(W_1) \setminus \{D'^{\max}\}$ such that $\mathrm{next}(x, D') \notin \psi^{-1}(W_1)$. We will show that

$$\varphi_1(\psi(\mathrm{next}(x, \psi^{-1}(W_1)))) \leq K_1 \varphi_1(\psi(x)) \tag{B.11}$$

and

$$\varphi_1(\psi(x)) \leq K_1 \varphi_1(\psi(\mathrm{next}(x, \psi^{-1}(W_1)))). \tag{B.12}$$

Suppose, by negation, that there is $y \in W_1$ such that $\psi(x) < y < \psi(\mathrm{next}(x, \psi^{-1}(W_1)))$. Then, $y \neq \psi(x')$ for any $x' \in D'$. (Otherwise, by definition of $\psi^{-1}(W_1)$, $y = \psi(\bar{x})$ for some $\bar{x} \in \psi^{-1}(W_1)$, implying that $x < \bar{x} < \mathrm{next}(x, \psi^{-1}(W_1))$ for some $\bar{x} \in \psi^{-1}(W_1)$, which is impossible.) Thus, $x = \max\{u \in D' \mid \psi(u) \leq y\}$ and $\mathrm{next}(x, \psi^{-1}(W_1)) = \min\{u \in D' \mid \psi(u) \geq y\}$. This implies that $\mathrm{next}(x, \psi^{-1}(W_1)) = \mathrm{next}(x, D')$, which contradicts that $\mathrm{next}(x, D') \notin \psi^{-1}(W_1)$. Hence, there does not exist any $y \in W_1$ such that $\psi(x) < y < \psi(\mathrm{next}(x, \psi^{-1}(W_1)))$. This implies that

$$\psi(x) \leq \psi(\mathrm{next}(x, \psi^{-1}(W_1))) \leq \mathrm{next}(\psi(x), W_1).$$

By the unimodality of φ_1, we have

$$\varphi_1(\psi(\mathrm{next}(x, \psi^{-1}(W_1)))) \leq \max\left\{\varphi_1(\psi(x)), \varphi_1(\mathrm{next}(\psi(x), W_1))\right\}. \tag{B.13}$$

Furthermore, if $\psi(x) \in W_1$, then

$$\mathrm{prev}(\psi(\mathrm{next}(x, \psi^{-1}(W_1))), W_1) = \psi(x). \tag{B.14}$$

We consider three different cases.

Case 1. $\psi(x) \in W_1$ and $\mathrm{next}(\psi(x), D) \notin W_1$. In this case, by the boundedness of W_1,

$$\max\{\varphi_1(\psi(x)), \varphi_1(\mathrm{next}(\psi(x), W_1))\} \leq K_1 \varphi_1(\psi(x)).$$

This, together with (B.13), implies the validity of (B.11). We now prove inequality (B.12). If $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) \notin W_1$, then by the locality of W_1 we have

$$\varphi_1(\mathrm{prev}(\psi(\mathrm{next}(x, \psi^{-1}(W_1))), W_1)) \leq K_1 \varphi_1(\psi(\mathrm{next}(x, \psi^{-1}(W_1)))).$$

By (B.14), $\varphi_1(\mathrm{prev}(\psi(\mathrm{next}(x, \psi^{-1}(W_1))), W_1)) = \varphi_1(\psi(x))$. Hence, (B.12) is valid. If $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) \in W_1$, then $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) = \mathrm{next}(\psi(x), W_1)$. Because $\mathrm{next}(\psi(x), D) \notin W_1$, by the boundedness of W_1, we have $\varphi_1(\psi(x)) \leq K_1 \varphi_1(\mathrm{next}(\psi(x), W_1))$, which implies the validity of (B.12).

Case 2. $\psi(x) \in W_1$ and $\mathrm{next}(\psi(x), D) \in W_1$. In this case, either $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) = \psi(x)$ or $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) = \mathrm{next}(\psi(x), D)$. Suppose, to the contrary, that $\psi(\mathrm{next}(x, \psi^{-1}(W_1))) = \mathrm{next}(\psi(x), D)$. Because $\mathrm{next}(x, D') \notin \psi^{-1}(W_1)$, there exists $\bar{x} \in D'$ such that $x < \bar{x} < \mathrm{next}(x, \psi^{-1}(W_1))$. Then, $\psi(x) \leq \psi(\bar{x}) \leq \psi(\mathrm{next}(x, \psi^{-1}(W_1)))$. In other words, either $\psi(\bar{x}) = \psi(x)$ or $\psi(\bar{x}) = \mathrm{next}(\psi(x), D)$. If $\psi(\bar{x}) = \psi(x)$, then let x' be the largest element in D' such that $\psi(x') = \psi(x)$. Clearly, $x < x' < \mathrm{next}(x, \psi^{-1}(W_1))$. By definition of $\psi^{-1}(W_1)$, we have $x' \in \psi^{-1}(W_1)$, which is a contradiction. If $\psi(\bar{x}) = \mathrm{next}(\psi(x), D)$, then let x'' be the smallest element in D' such that $\psi(x'') = \mathrm{next}(\psi(x), D)$. Clearly, $x < x'' <$

$\text{next}(x, \psi^{-1}(W_1))$. By definition of $\psi^{-1}(W_1)$, we have $x'' \in \psi^{-1}(W_1)$, which is also a contradiction. Therefore, $\psi(\text{next}(x, \psi^{-1}(W_1))) = \psi(x)$, which implies the validity of both (B.11) and (B.12).

Case 3. $\psi(x) \notin W_1$. In this case, by the locality of W_1,

$$\varphi_1(\text{next}(\psi(x), W_1)) \leq K_1 \varphi_1(\psi(x)).$$

This, together with (B.13), implies the validity of (B.11). We now prove inequality (B.12). If $\psi(\text{next}(x, \psi^{-1}(W_1))) \notin W_1$, then by the locality of W_1 we have

$$\varphi_1(\text{prev}(\psi(\text{next}(x, \psi^{-1}(W_1))), W_1)) \leq K_1 \varphi_1(\psi(\text{next}(x, \psi^{-1}(W_1)))).$$

By (B.14), $\varphi_1(\text{prev}(\psi(\text{next}(x, \psi^{-1}(W_1))), W_1)) = \varphi_1(\psi(x))$. Hence, (B.12) is valid. If $\psi(\text{next}(x, \psi^{-1}(W_1))) \in W_1$, then $\psi(\text{next}(x, \psi^{-1}(W_1))) = \text{next}(\psi(x), W_1)$. Because $\psi(x) \notin W_1$, we have $\text{next}(\text{prev}(\psi(x), W_1), D) \notin W_1$. Thus, by the boundedness of W_1,

$$\begin{aligned} &\max\left\{\varphi_1(\text{prev}(\psi(x), W_1)), \varphi_1(\text{next}(\text{prev}(\psi(x), W_1), W_1))\right\} \\ &\leq K_1 \varphi_1(\text{next}(\text{prev}(\psi(x), W_1), W_1)). \end{aligned} \tag{B.15}$$

By the unimodality of φ_1, we have

$$\varphi_1(\psi(x)) \leq \max\left\{\varphi_1(\text{prev}(\psi(x), W_1)), \varphi_1(\text{next}(\text{prev}(\psi(x), W_1), W_1))\right\}.$$

This, together with (B.15), implies that

$$\varphi_1(\psi(x)) \leq K_1 \varphi_1(\text{next}(\text{prev}(\psi(x), W_1), W_1)) = K_1 \varphi_1(\psi(\text{next}(x, \psi^{-1}(W_1))),$$

where the equality is due the fact that there is no $y \in W_1$ such that $\psi(x) < y < \psi(\text{next}(x, \psi^{-1}(W_1)))$. This implies the validity of (B.12).

To prove the locality of $\psi^{-1}(W_1)$, consider any $x \in D' \setminus \psi^{-1}(W_1)$. We will show that

$$\max\left\{\varphi_1(\psi(\text{prev}(x, \psi^{-1}(W_1)))), \varphi_1(\psi(\text{next}(x, \psi^{-1}(W_1))))\right\} \leq K_1 \varphi_1(\psi(x)). \tag{B.16}$$

By the same argument as in the proof of the boundedness of $\psi^{-1}(W_1)$, there does not exist any $y \in W_1$ such that $\psi(\text{prev}(x, \psi^{-1}(W_1))) < y < \psi(\text{next}(x, \psi^{-1}(W_1)))$. Hence, either $\psi(\text{prev}(x, \psi^{-1}(W_1))) = \psi(x) = \psi(\text{next}(x, \psi^{-1}(W_1)))$ or $\psi(x) \notin W_1$. If $\psi(\text{prev}(x, \psi^{-1}(W_1))) = \psi(x) = \psi(\text{next}(x, \psi^{-1}(W_1)))$, then clearly inequality (B.16) is valid. If $\psi(x) \notin W_1$, then by the locality of W_1,

$$\max\left\{\varphi_1(\text{prev}(\psi(x), W_1)), \varphi_1(\text{next}(\psi(x), W_1))\right\} \leq K_1 \varphi_1(\psi(x)). \tag{B.17}$$

Note that $\text{prev}(\psi(x), W_1) \leq \psi(\text{prev}(x, \psi^{-1}(W_1))) \leq \psi(\text{next}(x, \psi^{-1}(W_1))) \leq \text{next}(\psi(x), W_1)$. Thus, by the unimodality of φ_1,

$$\begin{aligned} &\max\left\{\varphi_1(\psi(\text{prev}(x, \psi^{-1}(W_1)))), \varphi_1(\psi(\text{next}(x, \psi^{-1}(W_1))))\right\} \\ &\leq \max\left\{\varphi_1(\text{prev}(\psi(x), W_1)), \varphi_1(\text{next}(\psi(x), W_1))\right\}. \end{aligned}$$

This, together with (B.17), implies the validity of (B.16).

Maximization of approximation sets. Let $\varphi_{\max}(x) = \max\{\varphi_1(x), \varphi_2(x)\}$ for all $x \in D$. Note that unimodality is closed under maximization. Thus, $\varphi_{\max}$ is a unimodal function, and a K-approximation set of it is well-defined. Denote

$W_{12} = W_1 \cup W_2$. Clearly, $D^{\min}, D^{\max} \in W_{12}$. Therefore, the first condition in Definition 4.2 is satisfied.

To prove the boundedness of W_{12} (with respect to function $\varphi_{\max}$), consider any $x \in W_{12} \setminus \{D^{\max}\}$ such that $\mathrm{next}(x, D) \notin W_{12}$. By the monotonicity of approximation sets, W_{12} is a K_i-approximation set of φ_i for $i = 1, 2$. Hence, by the boundedness of W_{12} (with respect to function φ_i), we have $\max\{\varphi_i(x), \varphi_i(\mathrm{next}(x, W_{12}))\} \leq K_i \min\{\varphi_i(x), \varphi_i(\mathrm{next}(x, W_{12}))\}$ for $i = 1, 2$. This implies that

$$\varphi_i(x) \leq \max\{K_1, K_2\} \min\{\varphi_{\max}(x), \varphi_{\max}(\mathrm{next}(x, W_{12}))\}$$

and

$$\varphi_i(\mathrm{next}(x, W_{12})) \leq \max\{K_1, K_2\} \min\{\varphi_{\max}(x), \varphi_{\max}(\mathrm{next}(x, W_{12}))\}$$

for $i = 1, 2$. Therefore,

$$\max\{\varphi_{\max}(x), \varphi_{\max}(\mathrm{next}(x, W_{12}))\} \leq \max\{K_1, K_2\} \min\{\varphi_{\max}(x), \varphi_{\max}(\mathrm{next}(x, W_{12}))\},$$

and the boundedness of W_{12} (with respect to function $\varphi_{\max}$) follows.

To show the locality of W_{12}, consider any $x \in D \setminus W_{12}$. For $i = 1, 2$, because $\mathrm{prev}(x, W_i) \leq \mathrm{prev}(x, W_{12}) < x$, by the unimodality of φ_i, we have $\varphi_i(\mathrm{prev}(x, W_{12})) \leq \max\{\varphi_i(\mathrm{prev}(x, W_i)), \varphi_i(x)\}$. By the locality of W_i, we have $\varphi_i(\mathrm{prev}(x, W_i)) \leq K_i \varphi_i(x)$. Thus,

$$\varphi_i(\mathrm{prev}(x, W_{12})) \leq \max\{K_i \varphi_i(x), \varphi_i(x)\} \leq \max\{K_1, K_2\} \varphi_{\max}(x). \tag{B.18}$$

Because $x < \mathrm{next}(x, W_{12}) \leq \mathrm{next}(x, W_i)$, by the unimodality of φ_i, we have $\varphi_i(\mathrm{next}(x, W_{12})) \leq \max\{\varphi_i(x), \varphi_i(\mathrm{next}(x, W_i))\}$. By the locality of W_i, we have $\varphi_i(\mathrm{next}(x, W_i)) \leq K_i \varphi_i(x)$. Thus,

$$\varphi_i(\mathrm{next}(x, W_{12})) \leq \max\{\varphi_i(x), K_i \varphi_i(x)\} \leq \max\{K_1, K_2\} \varphi_{\max}(x). \tag{B.19}$$

From (B.18) and (B.19),

$$\max\{\varphi_{\max}(\mathrm{prev}(x, W_{12})), \varphi_{\max}(\mathrm{next}(x, W_{12}))\} \leq \max\{K_1, K_2\} \varphi_{\max}(x),$$

and the locality of W_{12} follows. This completes the proof of the maximization of approximation sets. ∎

B.7. Proposition 6.2.

Proof. Denote $W_{12} = W_1 \cup W_2$. We give a proof for the case where φ_1 and φ_2 are both nondecreasing. The proof for the case with nonincreasing φ_1 and φ_2 is similar. We first prove the summation of approximation sets. Let $\varphi_{\mathrm{sum}}(x) = \varphi_1(x) + \varphi_2(x)$ for all $x \in D$. Note that φ_{sum} is nondecreasing, and hence a K-approximation set of it is well-defined. Clearly, $D^{\min}, D^{\max} \in W_{12}$. Thus, the first condition of Definition 4.2 is satisfied. To prove the boundedness of W_{12} (with respect to function φ_{sum}), consider any $x \in W_{12} \setminus \{D^{\max}\}$ such that $\mathrm{next}(x, D) \notin W_{12}$. By the monotonicity of approximation sets, W_{12} is a K_i-approximation set of φ_i for $i = 1, 2$. Hence, by the boundedness of W_{12} (with respect to function φ_i), we have $\max\{\varphi_i(x), \varphi_i(\mathrm{next}(x, W_{12}))\} \leq K_i \min\{\varphi_i(x), \varphi_i(\mathrm{next}(x, W_{12}))\}$ for $i = 1, 2$. This, together with the fact that φ_{sum} is nondecreasing, implies that

$$\begin{aligned}
&\max\{\varphi_{\mathrm{sum}}(x), \varphi_{\mathrm{sum}}(\mathrm{next}(x, W_{12}))\} = \varphi_{\mathrm{sum}}(\mathrm{next}(x, W_{12})) = \textstyle\sum_{i=1}^{2} \varphi_i(\mathrm{next}(x, W_{12})) \\
&\leq \textstyle\sum_{i=1}^{2} K_i \varphi_i(x) \leq \max\{K_1, K_2\} \varphi_{\mathrm{sum}}(x) \\
&= \max\{K_1, K_2\} \min\{\varphi_{\mathrm{sum}}(x), \varphi_{\mathrm{sum}}(\mathrm{next}(x, W_{12}))\},
\end{aligned}$$

and the boundedness of W_{12} (with respect to function φ_{sum}) follows. The locality of W_{12} follows directly from Proposition 4.3.

Next, we prove the minimization of approximation sets. Let $\varphi_{\min}(x) = \min\{\varphi_1(x), \varphi_2(x)\}$ for all $x \in D$. Note that $\varphi_{\min}$ is nondecreasing and hence, a K-approximation set of it is well-defined. Clearly, $D^{\min}, D^{\max} \in W_{12}$. Thus, the first condition of Definition 4.2 is satisfied. To prove the boundedness of W_{12} (with respect to function $\varphi_{\min}$), consider any $x \in W_{12} \setminus \{D^{\max}\}$ such that $\text{next}(x, D) \notin W_{12}$. As mentioned above, $\max\{\varphi_i(x), \varphi_i(\text{next}(x, W_{12}))\} \le K_i \min\{\varphi_i(x), \varphi_i(\text{next}(x, W_{12}))\}$ for $i = 1, 2$. This, together with the fact that $\varphi_{\min}$ is nondecreasing, implies that

$$\begin{aligned}&\max\{\varphi_{\min}(x), \varphi_{\min}(\text{next}(x, W_{12}))\} = \varphi_{\min}(\text{next}(x, W_{12})) = \min\nolimits_{i=1,2} \varphi_i(\text{next}(x, W_{12}))\\ &\le \min\nolimits_{i=1,2} K_i\varphi_i(x) \le \max\{K_1, K_2\}\varphi_{\min}(x)\\ &= \max\{K_1, K_2\} \min\{\varphi_{\min}(x), \varphi_{\min}(\text{next}(x, W_{12}))\},\end{aligned}$$

and the boundedness of W_{12} (with respect to function $\varphi_{\min}$) follows. The locality of W_{12} follows directly from Proposition 4.3.

Finally, we prove the approximation of approximation sets. The condition "φ_1 is a K_2-approximation of the restriction of φ_2 over W_1" is equivalent to

$$\varphi_2(x) \le \varphi_1(x) \le K_2\varphi_2(x) \quad \forall x \in W_1. \tag{B.20}$$

This implies that

$$\varphi_2(x) \le \hat{\varphi}_1(x) \le K_2\varphi_2(x) \quad \forall x \in W_1. \tag{B.21}$$

Note that

$$\hat{\varphi}_1(x) = \varphi_1(\text{next}(x, W_1)) \ge \varphi_2(\text{next}(x, W_1)) \ge \varphi_2(x) \quad \forall x \in D \setminus W_1, \tag{B.22}$$

where the equality follows from the definition of approximation induced by an approximation set, the first inequality follows from (B.20), and the second inequality holds because φ_2 is nondecreasing. Note also that

$$\begin{aligned}&\hat{\varphi}_1(x) = \varphi_1(\text{next}(x, W_1)) = \varphi_1(\text{next}(\text{prev}(x, W_1), W_1))\\ &\qquad \le K_1\varphi_1(\text{prev}(x, W_1)) \le K_1K_2\varphi_2(\text{prev}(x, W_1)) \le K_1K_2\varphi_2(x) \quad \forall x \in D \setminus W_1,\end{aligned} \tag{B.23}$$

where the first inequality is due to the boundedness of W_1, the second inequality follows from (B.20), and the last inequality holds because φ_2 is nondecreasing. From (B.21)–(B.23), we conclude that $\hat{\varphi}_1$ is a K_1K_2-approximation of φ_2 over D. ∎

B.8. Proposition 6.3.

Proof. Let $W_{12} = W_1 \cup W_2$ and $\varphi_{\text{sum}}(x) = \varphi_1(x) + \varphi_2(x)$ for all $x \in D$. Note that the sum of two convex functions is a convex function. Thus, φ_{sum} is a convex function, and a K-approximation set of it is well-defined. Clearly, $D^{\min}, D^{\max} \in W_{12}$. Thus, the first condition of Definition 4.2 is satisfied. To prove the boundedness of W_{12} (with respect to function φ_{sum}), consider any $x \in W_{12} \setminus \{D^{\max}\}$ such that $\text{next}(x, D) \notin W_{12}$. By the monotonicity of approximation sets, W_{12} is a K_i-approximation set of φ_i for $i = 1, 2$. Hence, by the boundedness of W_{12} (with respect to function φ_i), we have $\max\{\varphi_i(x), \varphi_i(\text{next}(x, W_{12}))\} \le K_i \min\{\varphi_i(x), \varphi_i(\text{next}(x, W_{12}))\}$ for $i = 1, 2$. If $\varphi_{\text{sum}}(x) \le \varphi_{\text{sum}}(\text{next}(x, W_{12}))$, then

$$\begin{aligned}&\max\{\varphi_{\text{sum}}(x), \varphi_{\text{sum}}(\text{next}(x, W_{12}))\} = \varphi_{\text{sum}}(\text{next}(x, W_{12})) = \textstyle\sum_{i=1}^{2} \varphi_i(\text{next}(x, W_{12}))\\ &\le \textstyle\sum_{i=1}^{2} K_i\varphi_i(x) \le \max\{K_1, K_2\}\varphi_{\text{sum}}(x)\\ &= \max\{K_1, K_2\} \min\{\varphi_{\text{sum}}(x), \varphi_{\text{sum}}(\text{next}(x, W_{12}))\}.\end{aligned}$$

If $\varphi_{\rm sum}(x) > \varphi_{\rm sum}(\text{next}(x, W_{12}))$, then

$$\begin{aligned}&\max\{\varphi_{\rm sum}(x), \varphi_{\rm sum}(\text{next}(x, W_{12}))\} = \varphi_{\rm sum}(x) = \textstyle\sum_{i=1}^{2} \varphi_i(x) \le \sum_{i=1}^{2} K_i\varphi_i(\text{next}(x, W_{12}))\\ &\le \max\{K_1, K_2\}\varphi_{\rm sum}(\text{next}(x, W_{12})) = \max\{K_1, K_2\}\min\{\varphi_{\rm sum}(x), \varphi_{\rm sum}(\text{next}(x, W_{12}))\}.\end{aligned}$$

Therefore, the boundedness of W_{12} (with respect to function $\varphi_{\rm sum}$) follows. To prove the locality of W_{12}, consider any $x \in D \setminus W_{12}$. By the locality of W_{12} (with respect to φ_i), we have

$$\max\{\varphi_i(\text{prev}(x, W_{12})), \varphi_i(\text{next}(x, W_{12}))\} \le K_i\varphi_i(x)$$

for $i = 1, 2$. Thus,

$$\begin{aligned}&\max\{\varphi_{\rm sum}(\text{prev}(x, W_{12})), \varphi_{\rm sum}(\text{next}(x, W_{12}))\}\\ &= \max\{\varphi_1(\text{prev}(x, W_{12})) + \varphi_2(\text{prev}(x, W_{12})), \varphi_1(\text{next}(x, W_{12})) + \varphi_2(\text{next}(x, W_{12}))\}\\ &\le \max\{\varphi_1(\text{prev}(x, W_{12})), \varphi_1(\text{next}(x, W_{12}))\} + \max\{\varphi_2(\text{prev}(x, W_{12})), \varphi_2(\text{next}(x, W_{12}))\}\\ &\le \textstyle\sum_{i=1}^{2} K_i\varphi_i(x) \le \max\{K_1, K_2\}\varphi_{\rm sum}(x).\end{aligned}$$

Therefore, the locality of W_{12} (with respect to function $\varphi_{\rm sum}$) follows. ☐

B.9. Proposition 10.3.

Proof. Consider any fixed t. For ease of exposition, we refer to the state I_t as inventory level and the action x_t as production/disposal quantity ("production" for positive values of x_t and "disposal" for negative values of x_t). Due to Condition 3(iii), and because $|a| = |b|$, the coefficients $|a|$ and $|b|$ are either -1, 0, or 1. From (3.4), and by the linearity of expectation, the optimal policy chooses x_t^* that minimizes

$$E_{D_t} g_t(I_t, \cdot, D_t) + E_{D_t} z_{t+1}(f_t(I_t, \cdot, D_t)). \tag{B.24}$$

The convex invariant (Proposition 9.1) implies that z_{t+1} is convex. Since $f_t(I_t, \cdot, D_t)$ is linear in its second variable, $z_{t+1}(f_t(I_t, \cdot, D_t))$ is also convex. Since a convex combination of convex functions is convex, the expression in (B.24) is also convex. Let

$$\begin{aligned}\alpha_t(x_t) &= E_{D_t} v_t(x_t, D_t) \text{ and}\\ \beta_t(I_t, x_t) &= E_{D_t} u_t(f_t(I_t, x_t, D_t), D_t) + E_{D_t} z_{t+1}(f_t(I_t, x_t, D_t)).\end{aligned}$$

Then,

$$z_t(I_t) = \min_{x_t \in \mathcal{A}_t(I_t)} \{\alpha_t(x_t) + \beta_t(I_t, x_t)\},$$

where α_t is V-shaped and $\beta_t(I_t, \cdot)$ is convex for every fixed I_t. Let α_t^- and α_t^+ be the slope of α_t on negative and positive x_t's, respectively.

Case 1. Both coefficients of I_t and x_t in function f_t are 0. In this case, $f_t(\cdot, \cdot, D_t)$ is indifferent to the action x_t, and so is β_t. Thus, the best strategy is to minimize the V-shaped function α_t (i.e., do nothing). Hence, in this case $r_t = -\infty$ and $s_t = \infty$.

Case 2. Both coefficients of I_t and x_t in function f_t are 1. In this case, $\beta_t(I, x) = \beta_t(I', x')$ for any I, I', x, x' such that $I + x = I' + x'$. Note that it is beneficial to move from inventory level $I - 1$ to inventory level I by producing one unit when

$$\alpha_t(1) + \beta_t(I - 1, 1) < \alpha_t(0) + \beta_t(I - 1, 0),$$

that is, when

$$\beta_t(0, I - 1) - \beta_t(0, I) > \alpha_t^+.$$

Let $R_t = \{I \in \mathbb{Z} \mid \beta_t(0, I-1) - \beta_t(0, I) > \alpha_t^+\}$ be all the inventory levels at which the production of one unit is beneficial. Similarly, let $S_t = \{I \in \mathbb{Z} \mid \beta_t(0, I+1) - \beta_t(0, I) > |\alpha_t^-|\}$ be all the inventory levels at which the disposal of one unit is beneficial. Let

$$r_t = \max R_t \quad \text{and} \quad s_t = \min S_t.$$

(*Note*: In case a set is empty, we define its maximum and minimum to be $-\infty$ and ∞, respectively.) The convexity of function $\beta_t(0, \cdot)$ implies that the sets R_t and S_t are connected over $\mathbb{Z}$. Thus, if the current inventory level is $I_t \in R_t$, then it is beneficial to produce $r_t - I_t$ units. Similarly, if the current inventory is $I_t \in S_t$, then it is beneficial to dispose of $I_t - s_t$ units. Hence, r_t and s_t are the threshold levels of a limit policy.

Case 3. The coefficients of I_t and x_t in function f_t are 1 and -1, respectively. Then, a negative x_t represents production, and a positive x represents disposal. In this case, $\beta_t(I, x) = \beta_t(I', x')$ for any I, I', x, x' such that $I - x = I' - x'$. Note that it is beneficial to move from inventory level $I - 1$ to inventory level I by producing one unit (i.e., by setting $x = -1$) when

$$\alpha_t(-1) + \beta_t(I-1, -1) < \alpha_t(0) + \beta_t(I-1, 0),$$

that is, when

$$\beta_t(0, 1-I) - \beta_t(0, -I) > |\alpha_t^-|.$$

Let $R_t' = \{I \in \mathbb{Z} \mid \beta_t(0, 1-I) - \beta_t(0, -I) > |\alpha_t^-|\}$ be all the inventory levels at which the production of one unit is beneficial. Similarly, let $S_t' = \{I \in \mathbb{Z} \mid \beta_t(0, -I-1) - \beta_t(0, -I) > \alpha_t^+\}$ be all the inventory levels at which the disposal of one unit is beneficial. Let $r_t = \max R_t'$ and $s_t = \min S_t'$. The convexity of function $\beta_t(0, \cdot)$ implies that the sets R_t' and S_t' are connected over $\mathbb{Z}$. Thus, if the current inventory level is $I_t \in R_t'$, then it is beneficial to produce $r_t - I_t$ units. Similarly, if the current inventory is $I_t \in S_t'$, then it is beneficial to dispose of $I_t - s_t$ units. Hence, r_t and s_t are the threshold levels of a limit policy.

Case 4. Both coefficients of I_t and x_t in function f_t are -1. This case is analogous to Case 2, and the proof is therefore omitted.

Case 5. The coefficients of I_t and x_t in function f_t are -1 and 1, respectively. This case is analogous to Case 3, and the proof is therefore omitted. □

Appendix C. Hardness results.

In this appendix, some #P-hardness and approximation hardness proofs are presented. These proofs make transformations from the following problems:

Problem: Partition
Instance: Finite set $V = \{v_1, \ldots, v_n\}$ of positive integers.
Question: Is there a subset $V' \subseteq V$ such that $\sum_{v \in V'} v = \sum_{v \in V \setminus V'} v$?
Note: Partition is NP-hard [29, p. 223].

Problem: Evaluating the CDF of convolution of discrete random variables (CDF)
Instance: Discrete random variables $X_1, \ldots, X_n$ and probabilities $p_{i,j} = \text{Prob}(X_i = a_{i,j})$, where $a_{i,j} \in \mathbb{Z}^+$, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Values $\Gamma \in \mathbb{Z}^+$ and $\gamma \in \mathbb{Q}^+$, where $0 < \gamma \leq 1$.
Question: Is $\text{Prob}(\sum_{i=1}^n X_i \leq \Gamma) \geq \gamma$?

Note: CDF is #P-hard even when $m = 2$, $p_{i,j} = \frac{1}{2}$ for all i, j, and $a_{i,2} = 0$ for all i [33].

Problem: Max k-Cover (MkC)
Instance: A set $S = \{1, \ldots, m\}$, ℓ distinct subsets $S_1, \ldots, S_\ell \subset S$, and a positive integer $k \leq \ell$.
Question: What is the maximum number of elements of S that can be covered by k subsets?
Note: MkC is known not to be approximated within a factor of $1 - \frac{1}{e}$, unless $P = NP$ [24].

C.1. Theorem 9.2.

Proof. Let $K \geq 1$ be a fixed desired approximation ratio. We will transform any given arbitrary instance of the partition problem to a convex DP except that one of the conditions stated in the theorem is not satisfied. Let $M = \frac{1}{2}\sum_{v \in V} v$. Without loss of generality, we assume M is a positive integer. (Otherwise the problem is trivially solved.)

We first consider the case where $\mathcal{S} \otimes \mathcal{A}$ is not integrally convex. We consider the following deterministic DP with $T = 2$ time periods. (For simplicity, we slightly abuse the notation by omitting the random variable D_t from the functions.) In the first time period, we set $\mathcal{S}_1 = [0]$, $\mathcal{A}_1(0) = [0, \ldots, 2^n - 1]$, $g_1(0, x_1) = 0$, and $f_1(0, x_1) = x_1$. The meaning of action x_1 is that we choose the ith element of V if and only if the ith bit of x_1 is 1. In the second time period, we set $\mathcal{S}_2 = [0, \ldots, 2^n - 1]$, $\mathcal{A}_2(I_2) = \{\delta_{val(I_2)=M}\}$, $g_2(I_2, x_2) = 1 - x_2$, $f_2(I_2, x_2) = 0$, where $val(I_2) = \sum_{i=1}^{n} v_i \delta_{\text{the } i\text{th bit of } I_2 \text{ is } 1}$. Here, function val decodes the value of the state. We let the terminal cost function be $g_3(I_3) = 0$ and the terminal state be $\mathcal{S}_3 = \{0\}$. For every time period, the state space is a contiguous interval, and the logarithm of its maximal element is polynomially bounded by the (binary) input size. The action space in period 1 is a contiguous interval for every state I_1, and the logarithm of its maximal element is polynomially bounded by the (binary) input size. The action space in time period 2 is a singleton for every state I_2, and the value of this singleton is computed in polynomial time. Thus, Condition 1 holds. Clearly, Condition 2 also holds. Note that this is a convex DP, except that $\mathcal{S}_2 \otimes \mathcal{A}_2$ is not necessarily an integrally convex set when there exists a partition $V' \subseteq V$ such that $\sum_{v \in V'} v = M$.

In this DP, $z_1(0) = 0$ if there exists $V' \subseteq V$ such that $\sum_{v \in V'} v = M$, and $z_1(0) = 1$ otherwise. Therefore, unless $P = NP$, there is no polynomial-time K-approximation algorithm for this DP.

Next, we consider the case where b is not restricted to -1, 0, or 1. We consider a deterministic DP with $T = n$ time periods, initial state M in period 1, and the following parameter setting: $\mathcal{S}_t = [-M, \ldots, M]$ for $t = 1, \ldots, T + 1$; $\mathcal{A}_t(I_t) = \{0, 1\}$, $g_t(I_t, x_t) = v_t x_t$, and $f_t(I_t, x_t) = I_t - v_t x_t$ for any $I_t \in \mathcal{S}_t$ and $t = 1, \ldots, T$; and $g_{T+1}(I_{T+1}) = KM|I_{T+1}|$ for any $I_{T+1} \in \mathcal{S}_{T+1}$. Clearly, this is a convex DP, except that the coefficient of the second variable of f_t is not restricted to -1, 0, or 1.

In this DP, $z_1(M) = M$ if there exists $V' \subseteq V$ such that $\sum_{v \in V'} v = M$, and $z_1(M) > KM$ otherwise. Hence, we can distinguish whether V' exists such that $\sum_{v \in V'} v = M$ by calculating a K-approximation for $z_1(M)$. Therefore, unless $P = NP$, there is no polynomial-time K-approximation algorithm for this DP. ∎

C.2. Theorem 10.1.

Proof. Given an arbitrary instance of MkC, we denote $\sigma_j = |S_j|$ and $S_j = \{s_{j,1}, \ldots, s_{j,\sigma_j}\}$ for $j = 1, \ldots, \ell$, and we construct the following instance of the stochastic ordered adaptive knapsack problem with nonindependent item volumes (SKP):

- Number of items, $n = \ell + 2m$.
- Knapsack capacity, $B = 2^n \ell k + (2^n - 1)k$.
- Profit of item t,

$$\pi_t = \begin{cases} 0 & \text{for } t = 1, \dots, \ell; \\ 1 & \text{for } t = \ell + 1, \dots, \ell + 2m. \end{cases}$$

- For $j = 1, \dots, m$, define a random variable $Y(j)$ with $\text{Prob}(Y(j) = 0) = \text{Prob}(Y(j) = 1) = \frac{1}{2}$.
- Volume of item t,

$$v_t = \begin{cases} 2^0 Y(s_{t,1}) + 2^1 Y(s_{t,2}) + \cdots & \\ +2^{\sigma_t - 1} Y(s_{t,\sigma_t}) + 2^n \ell & \text{for } t = 1, \dots, \ell; \\ M \cdot Y(t - \ell) & \text{for } t = \ell + 1, \dots, \ell + m; \\ M[1 - Y(t - \ell - m)] & \text{for } t = \ell + m + 1, \dots, \ell + 2m; \end{cases}$$

where $M = B + 1$.

Clearly, this construction can be done in polynomial time. Note that random variables $Y(1), \dots, Y(m)$ introduce dependencies among the item volumes.

To solve SKP, we can first select certain items among items $1, \dots, \ell$. Although these items have zero profit, they help to reveal which of the items $\ell + 1, \dots, \ell + 2m$ have volume 0 and which of them have volume M. (Those items have volume M will overflow the knapsack.) More specifically, for $t = 1, \dots, \ell$, we can express v_t as a binary number. If the jth last digit of this binary number is 1, then $Y(s_{t,j}) = 1$, which implies that item $\ell + s_{t,j}$ has volume M and item $\ell + m + s_{t,j}$ has volume 0. If the jth last digit is 0, then $Y(s_{t,j}) = 0$, which implies that item $\ell + s_{t,j}$ has volume 0 and item $\ell + m + s_{t,j}$ has volume M.

Suppose a maximum of q elements of S can be covered by k subsets. Then, the optimal solution to SKP should have an expected total profit of at least q but less than $q + 1$. To see this, let $S_{r_1}, \dots, S_{r_k}$ be those subsets by which the q elements are covered. We put items $r_1, \dots, r_k$ into the knapsack. These items help us identify q out of the items $\ell + 1, \dots, \ell + 2m$ with zero volume. These q items have a total profit of q. Note that among items $1, \dots, \ell$, we can put at most k items into the knapsack (otherwise the knapsack capacity will be exceeded), and selecting these k items allows us to reveal at most q items with zero volume. The other $2m - q$ items can only be chosen randomly. The total expected profit obtained from items $\ell + 1, \dots, \ell + 2m$ via random selection is less than 1. (Since each item has a 0.5 probability of overflowing the knapsack, the expected number of items that are chosen successfully is less than 1.)

The above analysis implies that if we know the optimal solution value of SKP is within $[q, q + 1)$, then in the given instance of MkC, a maximum of q elements of S can be covered by k subsets. Now, suppose we have a polynomial-time algorithm which can approximate SKP within a factor of $r = \frac{3(e-1)}{2e} \approx 0.948$. Then, given any instance of MkC, we apply this r-approximation algorithm to the corresponding SKP. If this r-approximation algorithm generates a solution with an expected total profit less than 3, then the optimal solution value of SKP is less than $\frac{2e}{e-1} < 4$, implying that the number of elements of S that can be covered by k subsets is at most 3. In such a case, $k \leq 3$ and $\sigma_j \leq 3$ for $j = 1, \dots, \ell$. Thus, the number of choices of k distinct subsets among $S_1, \dots, S_\ell$ is $\binom{\ell}{k} \leq \ell^3$. Hence, we can obtain in polynomial time an optimal solution to the given instance of MkC by enumerating all possible solutions. (Note that a more careful algorithm would find an optimal solution in time linear in ℓ.) On the other hand, if the r-approximation algorithm generates a solution with an expected total profit of $z \geq 3$, then the optimal solution value of SKP is at most $\frac{z}{r}$, implying that the number of elements of S which can be covered by k subsets must be within $[\lfloor z \rfloor, \ldots, \lfloor \frac{z}{r} \rfloor]$. It is easy to check that $\frac{z-1}{z/r} \geq 1 - \frac{1}{e}$. Hence, $\lfloor z \rfloor / \lfloor \frac{z}{r} \rfloor \geq 1 - \frac{1}{e}$, which implies that the solution to the given instance of MkC is approximated within a factor of $1 - \frac{1}{e}$. This is impossible unless $P = NP$. Therefore, unless $P = NP$, there is no polynomial-time algorithm which can approximate SKP within a factor of r. ∎

C.3. Theorem A.1.

Proof. Consider an arbitrary instance of the special case of CDF with $m = 2$, and $p_{i,j} = \frac{1}{2}$ for all i, j, and $a_{i,2} = 0$ for all i. We denote $M = 2\sum_{i=1}^{n} a_{i,1} + 1$ and construct the following instance of the single-item stochastic batch dispatch problem:

- Number of time periods, $T = n + 1$ time periods.
- Vehicle capacity, $Q = \sum_{i=1}^{n} a_{i,1}$.
- Initial amount of goods in the dispatch station, $I_1 = Q$.
- Amount of goods arriving in period t ($t = 1, \ldots, n$),

$$D_t = X_t = \begin{cases} a_{t,1} & \text{with probability } \frac{1}{2}; \\ 0 & \text{with probability } \frac{1}{2}. \end{cases}$$

- Amount of goods arriving in period $n + 1$, $D_{n+1} = 0$ with probability 1.
- Fixed cost of dispatching the vehicle in period t,

$$K_t = \begin{cases} 0 & \text{for } t = 1 \text{ and } n + 1; \\ M & \text{for } t = 2, \ldots, n. \end{cases}$$

- Unit cost of dispatching the goods in period t,

$$c_t = \begin{cases} 1 - \gamma & \text{for } t = 1; \\ 0 & \text{for } t = 2, \ldots, n + 1. \end{cases}$$

- Unit holding cost in period t,

$$h_t = \begin{cases} 0 & \text{for } t = 1, \ldots, n; \\ 1 & \text{for } t = n + 1. \end{cases}$$

Clearly, this construction can be done in polynomial time.

Note that in the optimal solution of this constructed instance, there must be no dispatching of goods in periods $2, \ldots, n$ (because dispatching a vehicle in any of these periods costs M, which exceeds the total cost of the trivial feasible solution of never dispatching any vehicle). Denote $X = \sum_{t=1}^{n} D_t$. Let x_1 be the amount of goods dispatched in period 1. Clearly, once the value of x_1 is chosen, the optimal decision is to dispatch $\min\{Q - x_1 + X, Q\}$ units (at zero cost) in period $n + 1$, and the holding cost incurred in period $n + 1$ is $(X - x_1)^+$. Thus, choosing the optimal value of x_1 is a newsvendor problem, in which the cost of ordering one unit too many is $1 - \gamma$, and the cost of ordering one unit too few is γ. Hence, the optimal decision is to select the smallest $x_1 \in \mathbb{Z}$ such that $\text{Prob}(X \leq x_1) \geq \frac{\gamma}{\gamma + (1-\gamma)} = \gamma$ (see, e.g., [69, sect. 9.2.1]). Therefore, if there exists a polynomial-time algorithm for solving the single-item stochastic batch dispatch problem, then this algorithm can be used for evaluating the CDF of convolution of discrete random variables. This implies that the single-item stochastic batch dispatch problem with time-varying costs is #P-hard. ∎

C.4. Theorem A.2.

Proof. Consider an arbitrary instance of the special case of CDF with $m = 2$, and $p_{i,j} = \frac{1}{2}$ for all i, j, and $a_{i,2} = 0$ for all i. We construct the following instance of the single-resource revenue management problem:

- Number of customer classes, $T = n + 1$.
- Available capacity, $C = \sum_{i=1}^{n} a_{i,1}$.
- Number of class 1 customers, $D_1 = C$ with probability 1.
- Number of class t customers ($t = 2, \ldots, n + 1$),

$$D_t = X_{t-1} = \begin{cases} a_{t-1,1} & \text{with probability } \frac{1}{2}; \\ 0 & \text{with probability } \frac{1}{2}. \end{cases}$$

- Revenue contribution per arrival of class t customer,

$$r_t = \begin{cases} 1 - \gamma & \text{for } t = 1; \\ 1 & \text{for } t = 2, \ldots, n + 1. \end{cases}$$

Clearly, this construction can be done in polynomial time.

Let x_t be the upper limit on the number of accepted bookings in period t. Obviously, it is optimal to set $x_2 = \cdots = x_{n+1} = C$ (as $r_2, \ldots, r_{n+1}$ are the same). Thus, choosing the optimal value of x_1 is a newsvendor problem, where the cost of ordering one unit too many (i.e., setting x_1 too low) is $1 - \gamma$, and the cost of ordering one unit too few (i.e., setting x_1 too high) is γ. The rest of the proof follows the same argument as in the proof of Theorem A.1. $\square$

C.5. Theorem A.3.

Proof. Consider an arbitrary instance of the special case of CDF with $m = 2$, and $p_{i,j} = \frac{1}{2}$ for all i, j, and $a_{i,2} = 0$ for all i. In this instance of CDF, $\text{Prob}(\sum_{i=1}^{n} X_i \leq \Gamma)$ must be a multiple of $\frac{1}{2^n}$ regardless of what Γ is. Hence, we may assume that γ is a multiple of $\frac{1}{2^n}$.

We define $\bar{a}_i = a_{i,1} + \frac{1}{2^i}$ ($i = 1, \ldots, n$) and $\bar{\Gamma} = \Gamma + 1 - \frac{1}{2^n}$. Because $\Gamma, a_{i,1}, \ldots, a_{i,n} \in \mathbb{Z}^+$,

$$\sum_{i \in U} a_{i,1} \leq \Gamma \text{ if and only if } \sum_{i \in U} \bar{a}_i \leq \bar{\Gamma}, \quad \forall\, U \subseteq \{1, \ldots, n\}. \tag{C.1}$$

Thus, the given CDF problem is equivalent to the problem of determining whether $\text{Prob}(\sum_{i=1}^{n} Y_i \leq \bar{\Gamma}) \geq \gamma$, where $Y_1, \ldots, Y_n$ are random variables with $\text{Prob}(Y_i = \bar{a}_i) = \text{Prob}(Y_i = 0) = \frac{1}{2}$ for $i = 1, \ldots, n$. Let $\bar{a}_{\max} = \max\{\bar{a}_1, \ldots, \bar{a}_n\}$ and $M = 2^{2n+1}\bar{a}_{\max}^2$. It is easy to check that for any $U \subseteq \{1, \ldots, n\}$,

$$1 + \frac{\sum_{i \in U} \bar{a}_i}{M} \leq \prod_{i \in U} \left(1 + \frac{\bar{a}_i}{M}\right) \leq 1 + \frac{\sum_{i \in U} \bar{a}_i}{M} + \frac{2^n \bar{a}_{\max}^2}{M^2} = 1 + \frac{\sum_{i \in U} \bar{a}_i}{M} + \frac{1}{2^{n+1} M}.$$

Let $\hat{\Gamma} = 1 + \frac{\bar{\Gamma}}{M} + \frac{1}{2^{n+1}M}$. Hence, for any $U \subseteq \{1, \ldots, n\}$, $\sum_{i \in U} \bar{a}_i \leq \bar{\Gamma}$ if and only if $\prod_{i \in U}(1 + \frac{\bar{a}_i}{M}) \leq \hat{\Gamma}$. (Note that if $\prod_{i \in U}(1 + \frac{\bar{a}_i}{M}) \leq \hat{\Gamma}$, then $\sum_{i \in U} \bar{a}_i \leq \bar{\Gamma} + \frac{1}{2^{n+1}}$, which implies that $\sum_{i \in U} \bar{a}_i \leq \bar{\Gamma}$ because $\bar{\Gamma}, \bar{a}_1, \ldots, \bar{a}_n$ are all multiples of $\frac{1}{2^n}$.) Therefore, by (C.1), for any $U \subseteq \{1, \ldots, n\}$, $\sum_{i \in U} a_{i,1} \leq \Gamma$ if and only if $\prod_{i \in U}(1 + \frac{\bar{a}_i}{M}) \leq \hat{\Gamma}$.

We now construct the following instance of lifetime consumption of risky capital:

- Number of time periods, $T = n + 1$.
- Initial capital, $I_1 = M$.

- Income received by the individual at the end of period t, $y_t = 0$.
- Utility function

$$u_t(x_t) = \begin{cases} x_t & \text{if } t = 1; \\ 0 & \text{if } t = 2, \dots, n; \\ \min\left\{\frac{x_t}{\gamma},\, M\right\} & \text{if } t = n+1. \end{cases}$$

- $\text{Prob}(D_t = 0) = \text{Prob}(D_t = \frac{\bar{a}_t}{M}) = \frac{1}{2}$ for $t = 1, \dots, n$, and $\text{Prob}(D_{n+1}{=}\,0) = 1$.

Obviously, this construction can be done in polynomial time.

Let $(H-1)\times 100\%$ denote the total percentage growth of capital between period 1 and period T, where H is a random variable. Since the utility is zero in periods $2, \dots, T-1$, the optimal decision is to make consumption in periods 1 and T only and set $x_T = I_T$. Thus, the only decision is to select a consumption amount x_1 in period 1. Let $x_1^* \in \mathbb{Z}^+$ denote the optimal value of x_1. Let $S = \{t \mid D_t = 1 + \frac{\bar{a}_t}{M};\, 1 \le t \le T-1\}$, which is a random subset of time periods. Then, $H = \prod_{t \in S}\left(1 + \frac{\bar{a}_t}{M}\right)$. Note that

$$\begin{aligned} H &\le \left(1 + \tfrac{1}{4^n}\right)^n \le \left[1 + \tfrac{1}{(n+1)2^n}\right]^n \\ &= 1 + \tfrac{n}{(n+1)2^n} + \tbinom{n}{2}\tfrac{1}{(n+1)^2 2^{2n}} + \tbinom{n}{3}\tfrac{1}{(n+1)^3 2^{3n}} + \cdots + \tbinom{n}{n}\tfrac{1}{(n+1)^n 2^{n\cdot n}} \\ &\le 1 + \tfrac{n}{(n+1)2^n} + \left[\tbinom{n}{2} + \tbinom{n}{3} + \cdots + \tbinom{n}{n}\right]\tfrac{1}{(n+1)2^{2n}} \\ &\le 1 + \tfrac{n}{(n+1)2^n} + \tfrac{1}{(n+1)2^n} = 1 + \tfrac{1}{2^n}. \end{aligned}$$

Hence,

$$1 \le H \le 1 + \frac{1}{2^n}. \tag{C.2}$$

If x_1 is less than $I_1 - \frac{\gamma M}{H}$, then $x_T = I_T = (I_1 - x_1)H > \gamma M$, and therefore the utility in period T is M, regardless of how much x_1 is below $I_1 - \frac{\gamma M}{H}$. Thus, the unit cost of setting x_1 too low is 1 (i.e., the utility of consuming one unit in period 1). If x_1 is greater than $I_1 - \frac{\gamma M}{H}$, then $x_T = I_T = (I_1 - x_1)H < \gamma M$, and therefore the utility in period T is $\frac{1}{\gamma}(I_1 - x_1)H$. Thus, the unit cost of setting x_1 too high is $\frac{1}{\gamma}H - 1$. By (C.2), this unit cost is at least $\frac{1}{\gamma} - 1$ and at most $\frac{1}{\gamma} - 1 + \frac{1}{2^n\gamma}$.

Define problem $\mathbf{P}_H$ to be the same as the constructed instance, except that the unit cost of setting x_1 too high is replaced by $\frac{1}{\gamma} - 1$. We let $\bar{x}_1^*$ denote the optimal value of x_1 in this problem. Define problem $\mathbf{P}_L$ to be the same as the constructed instance, except that the unit cost of setting x_1 too high is replaced by $\frac{1}{\gamma} - 1 + \frac{1}{2^n\gamma}$. We let $\underline{x}_1^*$ denote the optimal value of x_1 in this problem. Clearly, $\underline{x}_1^* \le x_1^* \le \bar{x}_1^*$.

The optimal solution to $\mathbf{P}_H$ can be obtained by first solving a newsvendor problem, in which the decision is to select the smallest value of $x \in \mathbb{R}^+$, denoted as x', such that

$$\text{Prob}\left(H \le \frac{\gamma M}{I_1 - x}\right) \ge \frac{1}{1 + (\frac{1}{\gamma} - 1)} = \gamma,$$

and then setting $\bar{x}_1^*$ to either $\lfloor x' \rfloor$ or $\lceil x' \rceil$, whichever yields a higher expected total utility. The optimal solution to $\mathbf{P}_L$ can also be obtained by first solving a newsvendor problem, in which the decision is to select the smallest value of $x \in \mathbb{R}^+$, denoted as x'', such that

$$\text{Prob}\left(H \le \frac{\gamma M}{I_1 - x}\right) \ge \frac{1}{1 + (\frac{1}{\gamma} - 1 + \frac{1}{2^n\gamma})} = \left(1 - \frac{1}{2^n + 1}\right)\gamma, \tag{C.3}$$

and then setting $\underline{x}_1^*$ to either $\lfloor x'' \rfloor$ or $\lceil x'' \rceil$, whichever yields a higher expected total utility. Note that by our assumption γ is a multiple of $\frac{1}{2^n}$, $\text{Prob}(H \leq \Gamma')$ is a multiple of $\frac{1}{2^n}$ for any $\Gamma' \in \mathbb{R}$, and $\gamma - \frac{1}{2^n} < (1 - \frac{1}{2^n+1})\gamma < \gamma$. Thus, (C.3) can be rewritten as

$$\text{Prob}\left(H \leq \frac{\gamma M}{I_1 - x}\right) \geq \gamma.$$

Hence, $x' = x''$, which implies that $\underline{x}_1^* = x_1^* = \bar{x}_1^*$.

We conclude that the constructed instance is reduced to the problem of selecting the smallest value $x \in \mathbb{R}^+$ such that $\text{Prob}(H \leq \frac{\gamma M}{I_1 - x}) \geq \gamma$, that is, selecting the smallest $\hat{\Gamma}$ such that $\text{Prob}(\prod_{t \in S}(1 + \frac{\bar{a}_t}{M}) \leq \hat{\Gamma}) \geq \gamma$. This is equivalent to selecting the smallest Γ such that $\text{Prob}(\sum_{i \in U} a_{i,1} \leq \Gamma) \geq \gamma$, or equivalently, $\text{Prob}(\sum_{i=1}^n X_i \leq \Gamma) \geq \gamma$. Therefore, if there exists a polynomial-time algorithm for solving our problem, this algorithm can be used for evaluating the CDF of convolution of discrete random variables. This implies that lifetime consumption of risky capital is #P-hard. ∎

C.6. Theorem A.4.

Proof. The proof is similar to the one of Theorem A.3. Let $\bar{a}_j$ and M be defined as in the proof of Theorem A.3. Consider an arbitrary instance of the special case of CDF with $m = 2$, and $p_{i,j} = \frac{1}{2}$ for all i, j, and $a_{i,2} = 0$ for all i. We transform it into the following instance of our problem:

- Number of time periods, $T = n + 2$.
- Initial capital, $I_1 = M$.
- Production function, $p_t(I_t) = I_t$, for $I_t \geq 0$ and $t = 1, \dots, n + 2$.
- Depreciation rate, $\Delta = 1$.
- Utility function

$$u_t(x_t) = \begin{cases} x_t & \text{if } t = 1; \\ 0 & \text{if } t = 2, \dots, n+1; \\ \min\left\{\frac{x_t}{\gamma}, M\right\} & \text{if } t = n+2. \end{cases}$$

- $\text{Prob}(D_t = 1) = \text{Prob}(D_t = 1 + \frac{\bar{a}_{t-1}}{M}) = \frac{1}{2}$ for $t = 2, \dots, n+1$, and $\text{Prob}(D_1 = 1) = \text{Prob}(D_{n+2} = 1) = 1$.

The rest of the proof follows the same argument as used in the proof of Theorem A.3. ∎

C.7. Theorem A.5.

Proof. Consider an arbitrary instance of the special case of CDF with $m = 2$, and $p_{i,j} = \frac{1}{2}$ for all i, j and $a_{i,2} = 0$ for all i. We denote $M = \sum_{i=1}^n a_{i,1} + 1$, and we construct the following instance of the generalized cash management problem:

- Number of time periods, $T = n$.
- Net amount of cash withdrawal made by customers in period t ($t = 1, \dots, n$),

$$D_t = X_t = \begin{cases} a_{t,1} & \text{with probability } \frac{1}{2}; \\ 0 & \text{with probability } \frac{1}{2}. \end{cases}$$

- Cost of selling \$1 value of stock in period t,

$$s_t = \begin{cases} 1 - \gamma & \text{for } t = 1; \\ M & \text{for } t = 2, \dots, n. \end{cases}$$

- Cost of buying \$1 value of stock in period t ($t = 1, \dots, n$), $b_t = M$.

- Cost of having \$1 value of shortage in cash at the end of period t,

$$k_t = \begin{cases} 0 & \text{for } t = 1, \dots, n-1; \\ 1 & \text{for } t = n. \end{cases}$$

- Cost of holding \$1 value of excessive cash at the end of period t ($t = 1, \dots, n$), $\ell_t = 0$.
- Initial cash balance, $I_1 = 0$.
- Discount factor, $\alpha = 1$.

Clearly, this construction can be done in polynomial time.

Let x_t be the amount of cash holding reduction in period t. Note that $D_1, \dots, D_n$ are all integers, and therefore it suffices to consider solutions with integer x_t values. Obviously, it is optimal to set $x_1 \le 0$ and $x_2 = \cdots = x_n = 0$ (because M is greater than the total cost of the trivial feasible solution of never buying or selling any stock). Thus, choosing the optimal value of x_1 is a newsvendor problem, in which the cost of ordering one unit too many (i.e., setting x_1 too low or, equivalently, raising too much cash in period 1) is $1 - \gamma$, and the cost of ordering one unit too few (i.e., setting x_1 too high or, equivalently, raising too little cash in period 1) is γ. The rest of the proof follows the same argument as in the proof of Theorem A.1. ∎

Acknowledgments. The authors thank the anonymous referees for their helpful comments and suggestions. The first author also thanks Oded Goldreich, Sudipto Guha, and Asaf Levin, for their inspiring discussions regarding the conference version of this paper, and Nimrod Megiddo, for bringing his attention to the book of Murota and the references therein.

REFERENCES

[1] J. Adda and R. Cooper, *Dynamic Economics: Quantitative Methods and Applications*, MIT Press, Cambridge, MA, 2003.

[2] M. Alekhnovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi, *Toward a model for backtracking and dynamic programming*, in Proceedings of the 20th Annual IEEE Conference on Computational Complexity, 2005.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Springer, Berlin, 1999.

[4] A. Bandyopadhyay and D. Gamarnik, *Counting without sampling: Asymptotics of the log-partition function for certain statistical physics models*, Random Structures Algorithms, 33 (2008), pp. 452–479.

[5] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali, *Simple deterministic approximation algorithms for counting matchings*, in Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007, pp. 122–127.

[6] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, NJ, 1962.

[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vol.* I, 3rd ed., Athena Scientific, Belmont, MA, 2005.

[8] K. M. Bretthauer and B. Shetty, *The nonlinear knapsack problem—algorithms and applications*, European J. Oper. Res., 138 (2002), pp. 459–472.

[9] R. L. Carraway, R. L. Schmidt, and L. R. Weatherford, *An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns*, Naval Res. Logist., 40 (1993), pp. 161–173.

[10] S. S. Chauhan, A. V. Eremeev, A. A. Romanova, V. V. Servakh, and G. J. Woeginger, *Approximation of the supply scheduling problem*, Oper. Res. Lett., 33 (2005), pp. 249–254.

[11] W. Chen, M. Dawande, and G. Janakiraman, *Fixed-dimensional stochastic dynamic programs: An approximation scheme and an inventory application*, Oper. Res., 62 (2014), pp. 81–103.

[12] X. Chen and D. Simchi-Levi, *Coordinating inventory control and pricing strategies with random demand and fixed ordering cost: The finite horizon case*, Oper. Res., 52 (2004), pp. 887–896.
[13] T. C. E. Cheng, Z.-L. Chen, C.-L. Li, and B. M.-T. Lin, *Scheduling to minimize the total compression and late costs*, Naval Res. Logist., 45 (1998), pp. 67–82.
[14] S. Chubanov, M. Y. Kovalyov, and E. Pesch, *An FPTAS for a single-item capacitated economic lot-sizing problem with monotone cost structure*, Math. Programming, 106 (2006), pp. 453–466.
[15] B. C. Dean, M. X. Goemans, and J. Vondrák, *Approximating the stochastic knapsack problem: The benefit of adaptivity*, Math. Oper. Res., 33 (2008), pp. 945–964.
[16] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*, Academic Press, New York, 1977.
[17] M. Dyer, *Approximate counting by dynamic programming*, in Proceedings of the 35th Annual ACM Symposium on the Theory of Computing, San Diego, CA, 2003, pp. 693–699.
[18] M. Dyer, A. Frieze, and M. Jerrum, *Approximately counting Hamilton paths and cycles in dense graphs*, SIAM J. Comput., 27 (1998), pp. 1262–1272.
[19] M. Dyer, L. A. Goldberg, C. Greenhill, and M. Jerrum, *The relative complexity of approximate counting problems*, Algorithmica, 38 (2004), pp. 471–500.
[20] E. J. Elton and M. J. Gruber, *On the cash balance problem*, Oper. Res. Quart., 25 (1974), pp. 553–572.
[21] G. D. Eppen and E. F. Fama, *Cash balance and simple dynamic portfolio problems with proportional costs*, Internat. Econ. Rev., 10 (1969), pp. 119–133.
[22] B. H. Faaland, *The multiperiod knapsack problem*, Oper. Res., 29 (1981), pp. 612–616.
[23] P. Favati and F. Tardella, *Convexity in nonlinear integer programming*, Ric. Operativa, 53 (1990), pp. 3–44.
[24] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.
[25] M. Florian, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Deterministic production planning: Algorithms and complexity*, Management Sci., 26 (1980), pp. 669–679.
[26] S. Frederick, G. Loewenstein, and T. O'Donoghue, *Time discounting and time preference: A critical review*, J. Econom. Literature, 40 (2002), pp. 351–401.
[27] G. Gallego and H. Hu, *Optimal policies for production/inventory systems with finite capacity and Markov-modulated demand and supply processes*, Ann. Oper. Res., 126 (2004), pp. 21–41.
[28] D. Gamarnik and D. Katz, *Correlation decay and deterministic FPTAS for counting colorings of a graph*, J. Discrete Algorithms, 12 (2012), pp. 29–47.
[29] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, New York, 1979.
[30] A. Goel and P. Indyk, *Stochastic load balancing and related problems*, in Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, 1999, pp. 579–586.
[31] N. Halman, *Approximating Convex Functions via Non-convex Oracles Under the Relative Noise Model*, Technical report 4226, www.optimization-online.org/DB_HTML/2014/02/4226.html, 2014.
[32] N. Halman, D. Klabjan, C.-L. Li, J. Orlin, and D. Simchi-Levi, *Fully polynomial time approximation schemes for stochastic dynamic programs*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 700–709.
[33] N. Halman, D. Klabjan, M. Mostagir, J. Orlin, and D. Simchi-Levi, *A fully polynomial-time approximation scheme for single-item stochastic inventory control with discrete demand*, Math. Oper. Res., 34 (2009), pp. 674–685.
[34] N. Halman, C.-L. Li, and D. Simchi-Levi, *Fully polynomial-time approximation schemes for time-cost tradeoff problems in series-parallel project networks*, Oper. Res. Lett., 37 (2009), pp. 239–244.
[35] N. Halman, J. B. Orlin, and D. Simchi-Levi, *Approximating the nonlinear newsvendor and single-item stochastic lot-sizing problems when data is given by an oracle*, Oper. Res., 60 (2012), pp. 429–446.
[36] M. I. Henig, *Risk criteria in a stochastic knapsack problem*, Oper. Res., 38 (1990), pp. 820–825.
[37] K. Hinderer and K.-H. Waldmann, *Cash management in a randomly varying environment*, European J. Oper. Res., 130 (2001), pp. 468–485.
[38] D. S. Hochbaum, *A nonlinear Knapsack problem*, Oper. Res. Lett., 17 (1995), pp. 103–110.
[39] D. S. Hochbaum, *Various notions of approximations: Good, better, best, and more*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., PWS Publishing, Boston, MA, 1997, pp. 346–398.
[40] E. Horowitz and S. Sahni, *Exact and approximate algorithms for scheduling nonidentical processors*, J. ACM, 23 (1976), pp. 317–327.

[41] O. H. Ibarra and C. E. Kim, *Fast approximation algorithms for the knapsack and sum of subset problems*, J. ACM, 22 (1975), pp. 463–468.
[42] D. L. Iglehart and S. Karlin, *Optimal policy for dynamic inventory process with nonstationary stochastic demands*, in Studies in Applied Probability and Management Science, K. J. Arrow, S. Karlin, and H. Scarf, eds., Stanford University Press, Stanford, CA, 1962, pp. 127–147.
[43] M. Jerrum and A. Sinclair, *Approximating the permanent*, SIAM J. Comput., 18 (1989), pp. 1149–1178.
[44] M. Jerrum, A. Sinclair, and E. Vigoda, *A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries*, J. ACM, 51 (2004), pp. 671–697.
[45] S. Kameshwaran and Y. Narahari, *Nonconvex piecewise linear knapsack problems*, European J. Oper. Res., 192 (2009), pp. 56–68.
[46] S. Karlin and A. J. Fabens, *A stationary inventory model with Markovian demand*, in Mathematical Methods in the Social Sciences, K. J. Arrow, S. Karlin, and P. Suppes, eds., Stanford University Press, Stanford, CA, 1959, pp. 159–175.
[47] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.
[48] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer, Berlin, 2004.
[49] J. Kleinberg, Y. Rabani, and É. Tardos, *Allocating bandwidth for bursty connections*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 664–673.
[50] A. J. Kleywegt and J. D. Papastavrou, *The dynamic and stochastic knapsack problem with random sized items*, Oper. Res., 49 (2001), pp. 26–41.
[51] E. L. Lawler, *Fast approximation algorithms for knapsack problems*, Math. Oper. Res., 4 (1979), pp. 339–356.
[52] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, *Sequencing and scheduling: Algorithms and complexity*, in Logistics of Production and Inventory, S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, eds., Handbooks Oper. Res. Management Sci. 4, North-Holland, Amsterdam, 1993, pp. 445–522.
[53] R. Levi, M. Pál, R. O. Roundy, and D. B. Shmoys, *Approximation algorithms for stochastic inventory control models*, Math. Oper. Res., 32 (2007), pp. 284–302.
[54] R. Levi, R. O. Roundy, and D. B. Shmoys, *Provably near-optimal sampling-based policies for stochastic inventory control models*, Math. Oper. Res., 32 (2007), pp. 821–839.
[55] R. Levi, R. O. Roundy, D. B. Shmoys, and V. A. Truong, *Approximation algorithms for capacitated stochastic inventory control models*, Oper. Res., 56 (2008), pp. 1184–1199.
[56] L. W. McKenzie, *Turnpike theory*, Econometrica, 44 (1976), pp. 841–865.
[57] M. Mihail and P. Winkler, *On the number of Eulerian orientations of a graph*, Algorithmica, 16 (1996), pp. 402–414.
[58] B. L. Miller, *On minimizing nonseparable functions defined on the integers with an inventory application*, SIAM J. Appl. Math., 21 (1971), pp. 166–185.
[59] K. Murota, *Discrete Convex Analysis*, SIAM, Philadelphia, 2003.
[60] J. Nascimento and W. Powell, *Dynamic programming models and algorithms for the mutual fund cash balance problem*, Management Sci., 56 (2010), pp. 801–815.
[61] K. P. Papadaki and W. B. Powell, *An adaptive dynamic programming algorithm for stochastic multiproduct batch dispatch problem*, Naval Res. Logist., 50 (2003), pp. 742–769.
[62] J. D. Papastavrou, S. Rajagopalan, and A. J. Kleywegt, *The dynamic and stochastic knapsack problem with deadlines*, Management Sci., 42 (1996), pp. 1706–1718.
[63] M. Parlar, Y. Wang, and Y. Gerchak, *A periodic review inventory model with Markovian supply availability*, Internat. J. Production Econom., 42 (1995), pp. 131–136.
[64] E. S. Phelps, *The accumulation of risky capital: A sequential utility analysis*, Econometrica, 30 (1962), pp. 729–743.
[65] H. M. Safer and J. B. Orlin, *Fast approximation schemes for multi-criteria flow, knapsack, and scheduling problems*, Working paper 3757-95, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, 1995.
[66] S. K. Sahni, *Algorithms for scheduling independent tasks*, J. ACM, 23 (1976), pp. 116–127.
[67] I. Saniee, *An efficient algorithm for the multiperiod capacity expansion of one location in telecommunications*, Oper. Res., 43 (1995), pp. 187–190.
[68] D. B. Shmoys and C. Swamy, *An approximation scheme for stochastic linear programming and its application to stochastic integer programs*, J. ACM, 53 (2006), pp. 978–1012.
[69] D. Simchi-Levi, X. Chen, and J. Bramel, *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*, 3rd ed., Springer-Verlag, New York, 2014.
[70] M. Sniedovich, *Preference order stochastic knapsack problems: Methodological issues*, J. Oper. Res. Soc., 31 (1980), pp. 1025–1032.

[71] J.-S. Song and P. Zipkin, *Inventory control in a fluctuating demand environment*, Oper. Res., 41 (1993), pp. 351–370.
[72] E. Steinberg and M. S. Parks, *A preference order dynamic program for a knapsack problem with stochastic rewards*, J. Oper. Res. Soc., 30 (1979), pp. 141–147.
[73] K. T. Talluri and G. J. van Ryzin, *The Theory and Practice of Revenue Management*, Kluwer, Boston, 2004.
[74] C. P. M. Van Hoesel and A. P. M. Wagelmans, *Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems*, Math. Oper. Res., 26 (2001), pp. 339–357.
[75] V. V. Vazirani, *Approximation Algorithms*, Springer, Berlin, 2001.
[76] H. M. Wagner and T. M. Whitin, *Dynamic version of the economic lot size model*, Management Sci., 5 (1958), pp. 89–96.
[77] D. Weitz, *Counting independent sets up to the tree threshold*, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing, 2006, pp. 140–149.
[78] W. D. Whisler, *A stochastic inventory model for rented equipment*, Management Sci., 13 (1967), pp. 640–647.
[79] G. J. Woeginger, *When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)?*, INFORMS J. Comput., 12 (2000), pp. 57–74.
[80] C. A. Yano and H. L. Lee, *Lot sizing with random yields: A review*, Oper. Res., 43 (1995), pp. 311–334.
[81] P. H. Zipkin, *Foundations of Inventory Management*, McGraw-Hill, Boston, 2000.