NOT FOR DISTRIBUTION

Fully Polynomial Time Approximation Schemes for Stochastic Dynamic Programs*

Nir Halman†‡ Diego Klabjan‡§ Chung-Lun Li¶ James Orlin†‖
David Simchi-Levi†‡

E-mail: {halman,jorlin,dslevi}@mit.edu, d-klabjan@northwestern.edu, lgtclli@polyu.edu.hk

November 20, 2008

Abstract

We develop a framework for obtaining Fully Polynomial Time Approximation Schemes (FPTASs) for stochastic univariate dynamic programs with either convex or monotone single-period cost functions. Using our framework, we give the first FPTASs for several NP-hard problems in various fields of research such as knapsack-related problems, logistics, operations management, economics, and mathematical finance.

---

*An extended abstract containing parts of this work appeared in the proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2008

†Massachusetts Institute of Technology, Cambridge, MA

‡Research supported in part by NSF Contracts DMI-0085683 and DMI-0245352, and by NASA interplanetary supply chain management and logistics architecture.

§Northwestern University, Evanston, IL

¶The Hong Kong Polytechnic University

‖Research supported in part by ONR grant N00014-98-1-0317.

1 Introduction

Dynamic Programming (DP). Dynamic Programming is an algorithmic technique used for solving sequential, or multi-stage, decision problems and is a fundamental tool in combinatorial optimization (e.g., [Hoc97], Section 2.5 in [ACG+99], and Chapter 8 in [Vaz01]). The first work done on fully polynomial time approximation schemes could be tracked back to the mid-70s, starting with the classic work of Horowitz, Ibarra, Kim and Sahni [HS76, IK75, Sah76] on scheduling and knapsack problems. Those first FPTASs were based on discrete time finite time horizon dynamic programming formulations, which find an exact optimal solution in pseudo-polynomial time. A discrete time finite time horizon dynamic program is to find an optimal policy over a finite time horizon that minimizes the average cost. At the beginning of a time period, the state of the system is observed and next 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. Simultaneously, a single period cost is incurred.

We can formally model this by means of the optimality or Bellman equation. To this end, let $z_t(I_t)$ be the cost-to-go 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 horizon, given that at the beginning of time period t the state is I_t. The equation reads

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

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

Convex/Monotone DP. We study three special cases of such dynamic programs. In the first case the system dynamics are linear in the state I_t and the action x_t, and the cost function g_t is convex, for every t. Under these assumptions, we show that z_t is a convex function for every t. We call this the *convex case*. In the second case we require that g_t is nonincreasing in I_t and can be expressed as the sum of two functions monotone in x_t. We also require that f_t is nondecreasing in I_t and monotone in x_t, for every time period t. In this case, the value function is nonincreasing, and we address it as the *nonincreasing case*. The third case, whose conditions are analogous to the nonincreasing case, is called the *nondecreasing case*. We refer to these last two cases as the *monotone cases*.

Fully Polynomial Time Approximation Schemes. Among algorithms with performance guarantees on the maximum amount of relative error, *Fully Polynomial Time Approximation Schemes (FPTASs)* are by far the strongest results. 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 use a discrete approximation in which the number of bits of precision used for the cost functions is at most logarithmic in the data and in $1/\epsilon$ (so that the algorithm will be polynomial time), and so that no other data is approximated. It is critical to design the algorithms and the approximations so that small errors at one stage do not turn into large errors at subsequent stages. Early work on FPTASs was pioneered by Horowitz and Sahni [HS76], Ibarra and Kim [IK75], and Sahni [Sah76], and since then, the most common techniques for constructing FPTASs are dominance (i.e., omitting states and actions of the DP which are dominated, or approximately dominated, by another state or action) and scaling and rounding the data (see, for example, section 2.5 in [ACG+99] and [Hoc97]). Many FPTASs are easily constructed once the key ideas from [IK75, Sah76] are understood. But other FPTASs require great care in the algorithm design and analysis. We note that if FPTASs were easily developed whenever they exist, there would have been FPTASs for stochastic optimization prior to our result in 2006 ([HKM+06]).

Our results. In this paper we introduce a general framework for obtaining FPTASs for stochastic dynamic programs, and show that by making use of this framework, we can construct FPTASs for a number of difficult stochastic and deterministic optimization problems. These problems are all NP-hard; they have no known FPTASs; and they cover a broad range of applications. Our main result is the development of an initial sufficient set of conditions that guarantee the *existence* of an FPTAS, and the *construction* of such an FPTAS. In this way we get a framework for obtaining an FPTAS to *any* stochastic univariate convex or monotone dynamic program (with independent random variables). We show that our framework can handle several well-studied cases of non-independent random variables. We show that it is *not* possible to extend our framework for general non-independent random variables, unless $P = NP$.

Our approach. In a previous work [HKM+06], we have studied a single-item stochastic dynamic inventory control problem. In that work, we introduced the notions of K-approximation sets and K-approximation functions and "tailor" it 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 provide an ad-hoc FPTAS for the inventory control problem. Our current work makes use of the notion of K-approximation sets and functions as well, but we target at the development of a general framework for FPTAS construction. For this sake 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 last column in 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 of a given function φ_i, $i = 1, 2$, and α, β are nonnegative reals, Table 1 summarizes which operations on which functions 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 $\min\{\varphi_1, \varphi_2\}$. Moreover, $W_1 \cup W_2$ serves as a $\max\{K_1, K_2\}$-approximation set for $\min\{\varphi_1, \varphi_2\}$, whenever φ_1, φ_2 are monotone functions.

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

Table 1: The Calculus of K-approximation Sets and Functions.

The calculus serves as a simple (and often automated) tool for the analyses for error propagations for monotone, unimodal, and convex functions. While some of the rules and procedures in the calculus are very straightforward, others are far more subtle (including some of the rules restricted to convex functions) and require careful analysis.

Applications. Our newly developed framework has numerous applications. In this paper we present nine examples of such application problems. No known FPTAS has been reported in the literature for any of these problems except for problem 5. The first three problems are variants of the classical 0/1 knapsack problem:

1. Stochastic ordered adaptive knapsack problem [DGV04]: The input consists of the knapsack volume B, and a description of n items. The items arrive in a predefined order. Item i has deterministic profit π_i and a stochastic volume v_i, whose distribution is known in advance. The task is to maximize the expected total profit from items successfully placed in the knapsack (i.e., whose total volume does not exceed the knapsack volume). The actual volume of an item is unknown until we instantiate it by attempting to place it in the knapsack. The problem is called adaptive since the decision about item i is made only after knowing the actual volumes of the previous items which were selected.

2. Nonlinear knapsack problem [Hoc95, KN09, SO95]: The problem is similar to the classical integer knapsack problem. Instead of fixed volumes and profits per item, we are given general nondecreasing volume and profit functions, i.e., placing j items of type i in the knapsack results in total profit $\pi_i(j)$ and requires volume $v_i(j)$. The goal is to maximize the profit from items placed in the knapsack without exceeding its volume. [Hoc95] gives an FPTAS for the special case where π_i is concave and v_i is convex for $i = 1, \ldots, n$. [SO95] give an FPTAS for the special case where π_i is general and v_i is linear for $i = 1, \ldots, n$.

3. Dynamic capacity expansion problem [San95]: This problem is best viewed as a multi-period minimum integer knapsack problem. Given a sequence of demands $d_1, \ldots, d_t$ and a set of items $\{1, \ldots, n\}$ of various volumes v_i and costs c_i, we would like to determine a combination of quantities of each of these items to be placed in a knapsack in each time period. Our objective is to satisfy the accumulated demand in minimal cost. [San95] gives a pseudo-polynomial time solution for this problem.

4. Time-cost tradeoff machine scheduling problem [CCLL98]: There is a single machine and n jobs $J_1, \ldots, J_n$. Job J_j has a given due date d_j, a late penalty w_j, a "normal" processing time $\bar{p}_j$, and a nonincreasing resource consumption function ρ_j 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 it incurs a cost of $\rho_j(x_j)$. 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 onto the machine so that the total cost is minimized, Cheng *et al.* [CCLL98] have considered a special case of this problem in which ρ_j is a linear function. We show that this problem falls under the nonincreasing case.

5. Single-item stochastic inventory control problem [HKM$^+$06]: This is a classical problem in supply chain management (see Chapter 9 in [SCB05] for an overview). In this problem we need to find replenishment quantities in each time period that minimize the expected procurement and holding/backlogging cost. If we assume that the holding cost, which includes a potential penalty for backlogging, is convex, and, in addition, the procurement cost is convex, then we satisfy the convexity assumptions.

6. Batch disposal problem [PP03]: Consider managing a warehouse where a single truck of capacity Q is available to dispatch the goods. The goods are received randomly based on a distribution known in advance. The question is whether we dispatch the truck and if yes, what is its load. If we use the truck, a fixed cost is incurred in addition to a per-unit cost. The goods that remain in the warehouse incur a per-unit holding cost. We show that this problem falls under the nondecreasing case.

7. Lifetime consumption strategy with risk exposure [Phe62]: There is a single consumer who must manage her capital in discrete time periods. She can spend some amount of capital, which is governed by the underlying nondecreasing utility function. The remaining capital yields a stochastic return rate, and, in addition, she receives a fixed amount of wealth in each time period. The problem is to find an optimal consumption strategy. We show that this problem falls under the nondecreasing case.

8. Deterministic and stochastic growth models [AC03]: Consider a single consumer with initial capital who wishes to manage her capital in order to maximize total utility over a finite discrete time horizon. In each time period the capital grows based on a production function, there is a rate of return $1 - \delta$ for initial capital, and a utility function.

9. Cash management problem [HW01]: In this problem one needs to manage the cash flow of a mutual fund. In 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 discovered, and consequently the cash balance of the mutual fund is determined. If the balance is negative, the fund borrows money from the bank. If the balance is positive, than a cost is incurred due to the fact that the fund's money could have been invested elsewhere. Whenever all these costs are convex functions, then we satisfy the convex assumptions.

Organization of the paper. We state our model and give the sufficient conditions needed for our framework to work in Section 3. In Section A we give several applications of our model (i.e., formulate a few problems so that they satisfy the aforementioned conditions). In Section 4 we efficiently build succinct approximations using the notions of K-approximation sets and functions. We develop the Calculus of K-approximation functions in Section 5 and the Calculus of K-approximation sets in Section 6. In Section 7 we develop a theory which links K-approximation sets and functions to dynamic programming. Based upon this theory, we present in Section 8 our main result for convex DP, namely the FPTAS together with its analysis, while in Section 9 we present our FPTAS for monotone DP. Several extensions of our framework that deal with non-independent random vectors, non-exact evaluation of the cost functions, and an analysis of the structure of an optimal policy for Convex DP, are given in Section 10. We end this paper with some concluding remarks. In the Appendix we give proofs for the statements that were not proven earlier throughout the paper (mostly concerning the Calculus of K-approximation sets and various hardness results).

2 Preliminaries

2.1 Notation

Let $\mathbb{R}, \mathbb{Q}, \mathbb{Z}, \mathbb{N}$ denote the set of real numbers, rational numbers, integers, and positive integers, respectively. Let $D \subset \mathbb{R}$ be a finite set of reals. Let $D^{\min}, D^{\max}$ denote the minimal and maximal element in D, respectively. For $x < D^{\max}$ we let $\text{next}(x, D) = \min\{y \in D \mid y > x\}$. If $y = \text{next}(x, D)$ then we let $\text{prev}(y, D) = x$. For any pair of integers $-\infty \leq A < B \leq \infty$, let $[A, \ldots, B]$ denote the set of integers between A and B. We call $[A, \ldots, B]$ a *contiguous interval.* For every number $x \in \mathbb{R}$ we let $x^+ = \max\{0, x\}$ and $x^- = \max\{0, -x\}$. For a subset $X \subseteq \mathbb{R}$, we denote by X^+ the set of nonnegative numbers in X, i.e., $X^+ := \{x \in X \mid x \geq 0\}$. For $x \in \mathbb{R}$, $\lceil x \rceil$ denotes the smallest integer not smaller than x (rounding up to the nearest integer) and $\lfloor x \rfloor$ the largest integer not larger than x (rounding down to the nearest integer. 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 $\log z$.

Let $\varphi : D \to \mathbb{R}$ be a real-valued function. Let $\varphi^{\max} = \max_{x \in D} |\varphi(x)|$ and $\varphi^{\min} = \min\{|\varphi(x)| \mid x \in D \text{ and } \varphi(x) \neq 0\}$. We denote by t_φ the time needed to evaluate φ on a single point in its domain. φ is called *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 $D \cap \{x \mid x > x^*\}$). For any subset $D' \subseteq D$, we define the *piecewise linear extension of* φ *induced by* D' as the continuous function obtained by making φ linear between successive values of D'.

2.2 Discrete convex functions

For any subset $D' \subseteq D$, a function φ over a discrete set D is called *convex over* D' if its piecewise linear extension induced by D' is convex.

We now turn to define convex functions over two-dimensional discrete domains. While the definition of monotone functions over these domains is straitforward, it is not the case with convex functions as illustrated in the following key counterexample:

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* $\mathbb{R}^2$. *Let* $\psi_1, \psi_2 : \mathbb{R} \to \mathbb{R}$ *be defined as follows.*

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

and

$$\psi_2(x) = \min_{y \in \mathbb{Z}} \varphi(x, y). \tag{2}$$

Since $\psi_1 \equiv 0$, it is trivially convex over $\mathbb{R}$. On the other hand, as 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 us that if we want ψ_2 to be convex over $\mathbb{R}$, we need to impose on φ a stronger condition than being convex on $\mathbb{R}^2$.

Let X be a contiguous interval. For every $x \in X$, let $Y(x)$ be a nonempty contiguous interval. Let $X \otimes Y := \bigcup_{x \in X}(\{x\} \times Y(x)) \subset \mathbb{Z}^2$, see Figure 1.

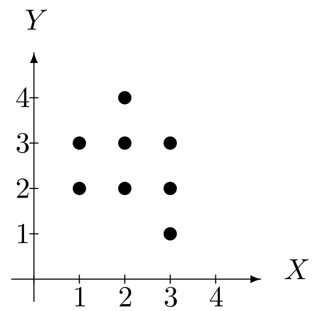

Figure 1: $X \otimes Y$ for $X = \{1,2,3\}$ and $Y(1) = \{2,3\},\ Y(2) = \{2,3,4\},\ Y(3) = \{1,2,3\}$.

We restrict our discussion to the special case of sets $X \otimes Y$ for which there exists a convex (and not necessarily bounded) polyhedron C_{XY}, whose edges slopes are in the set $\{-\infty, -1, 0, 1, \infty\}$ such that $X \otimes Y \equiv C_{XY} \cap \mathbb{Z}^2$. Such sets are called *integrally-convex*, see Figure 2 (taken from [Mur03]).

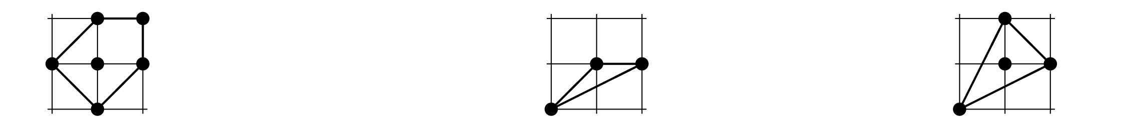

Figure 2: Concept of integrally-convex sets.

We note that in discrete optimization, discrete analogues of convexity, or "discrete convexity" for short, have already been considered. Miller [Mil70] investigated a class of discrete functions, called "discretely-convex" functions, such that local optimality implies global optimality. Favati-Tardella [FT90] considered a certain special way of extending functions defined over the integer lattice to piecewise-linear functions defined over the real space, and introduced the concept of "integrally-convex" functions. Murota [Mur03] introduced the concepts "L-convexity" and "M-convexity", where "L" stands for "Lattice" and "M" for "Matroid". M-convex/L-convex functions have various desirable properties as discrete convex functions: extendibility to ordinary (continuous) convex functions, duality theorems, conjugacy between M/L-convex functions, etc.

In this paper we show that a sufficient condition for ψ_2 in (2) to be convex is that φ is defined over an integrally-convex set, and can be expressed as $\varphi(x,y) = \varphi_1(x) + \varphi_2(y) + \omega(\tau(x,y))$, where $\varphi_1, \varphi_2, \omega$ are ordinary univariate convex functions, and $\tau(x,y) = ax + by + c$, where $a, c \in \mathbb{Z}$ and $b \in \{-1, 0, 1\}$. This is a key observation for developing our FPTAS for convex DP.

An alternative sufficient condition for ψ_2 in (2) to be convex is that function φ is "integrally-convex", as defined in [FT90]. Since all the convex problems we solve satisfy the first sufficient condition, we will not discuss in this paper the second condition.

3 Model statement

In this section we review a basic model of decision making under stochastic uncertainty over a finite number of time periods. We consider the following formulation for a finite horizon stochastic dynamic program, as defined in Bertsekas [Ber95]. The model has two principal features: (1) an underlying *discrete time dynamic system*, and (2) a *cost function that is additive over time.* The system dynamics are of the form

$$I_{t+1} = f_t(I_t, x_t, D_t), \quad t = 1, \ldots, T, \tag{3}$$

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,
T	is the number of time periods.

The cost function is additive in the sense that the cost incurred at time period t, denoted by $g_t(I_t, x_t, D_t)$, 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 find

$$z^*(I_1) := \min_{x_1,\ldots,x_T} E\left\{ g_{T+1}(I_{T+1}) + \sum_{t=1}^{T} g_t(I_t, x_t, D_t) \right\}, \tag{4}$$

where the expectation is taken with respect to the joint distribution of the random variables involved. The optimization is over the actions $x_1, \ldots, x_T$ which are selected with the knowledge of the current state.

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 $(\mathcal{S}_t \otimes \mathcal{A}_t) \times \mathcal{D}_t$. We state now the well known DP recursion for this model.

Theorem 3.1 (The DP Recursion [Bel57])**.** *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:*

$$\begin{aligned} z_{T+1}(I_{T+1}) &= g_{T+1}(I_{T+1}), \\ z_t(I_t) &= \min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t}\left\{ g_t(I_t, x_t, D_t) + z_{t+1}(f_t(I_t, x_t, D_t)) \right\}, \quad t = 1, \ldots, T, \end{aligned} \tag{5}$$

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

Note that assuming $\mathcal{A}_t(I_t) \equiv \mathcal{A}$ and $\mathcal{S}_t \equiv \mathcal{S}$ for every t and $I_t \in \mathcal{S}_t$, the recurrence given in (5) yields an exact solution for $z_1(I_1)$ in $O(T|\mathcal{A}||\mathcal{S}|)$ time, which is pseudopolynomial in the input size, i.e., the cardinality of these sets may be exponential in the (binary) input size.

The input data of the problem consists of the number of time periods T, the initial state I_1, an oracle that evaluates g_{T+1}, and for each time period $t = 1, \ldots, T$, oracles that evaluate the functions g_t and f_t, and the discrete random variable D_t. 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 $i < j$. Moreover, we are also given positive integers $q_{t,1}, \dots, q_{t,n_t}$ such that

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

We define for every $t = 1, \dots, T$ and $i = 1, \dots, n_t$ the following values:

$p_{t,i} = \text{Prob}[D_t = d_{t,i}]$	probability that D_t has the value $d_{t,i}$ in time period t;
$n^* = \max_t n_t$	maximum number of different values D_t can take over the entire time horizon;
$D^* = \sum_{t=1}^{T} d_{t,n_t}$	maximum possible total value the random variables can take over the entire time horizon;
$Q_t = \sum_{j=1}^{n_t} q_{t,j}$	a common denominator of all the probabilities in time period t;
$M_t = \prod_{j=t}^{T} Q_j$	a common denominator of all the probabilities in all time periods following time period $t-1$;
$M_{T+1} = 1$.	

Note that with the above notation we have

$$E_{D_t}\{g_t(I_t, x_t, D_t) + z_{t+1}(f_t(I_t, x_t, D_t))\} = \sum_{j=1}^{n_t} p_{t,j}\left(g_t(I_t, x_t, d_{t,j}) + z_{t+1}(f_t(I_t, x_t, d_{t,j}))\right). \tag{6}$$

In order to derive an FPTAS for our dynamic program, we need the following conditions to hold:

Condition 1. *There exists a constant d such that $\mathcal{S}_{T+1}, \mathcal{S}_t, \mathcal{A}_t \subset \mathbb{Z}$, and $\mathcal{D}_t \subset \mathbb{Z}^d$, for every $t = 1, \dots, T$. For any set X among these sets, and any $1 \le k \le |X|$, $\log \max_{x \in X} |x|$ is bounded polynomially by the (binary) input size and the kth largest element in X can be found in time logarithmic in $|X|$.*

Typically the state and action spaces are contiguous intervals, so the kth largest element in X can be found in constant time. However, as stated in Condition 1, we allow a more general setting. As already mentioned in the previous part, for the ease of exposition we will assume that these spaces consist of contiguous intervals.

Let $U_D^{\mathcal{S}} = \max_{t=1,\dots,T+1} |\mathcal{S}_t|$ be the size of the maximal state space. Similarly, let $U_D^{\mathcal{A}} = \max_{t=1,\dots,T} |\mathcal{A}_t|$ be the size of the maximal action space. Condition 1 implies that $\log U_D^{\mathcal{S}}$ and $\log U_D^{\mathcal{A}}$ are polynomially bounded by the input size.

Condition 2. *For every $t = 1, \dots, T+1$, g_t is a function whose values are nonnegative rational numbers, and the binary size of any of its values is polynomially bounded by the (binary) size of the input.*

Let $U_R := \max_t \max_{I \in I_t,\, x \in \mathcal{A}_t(I),\, d \in \mathcal{D}_t} g_t(I, x, d)$ denote the maximal single-period cost value. In this way, Condition 2 tells us that the logarithm of $TU_R \ge \max_t \frac{g_t^{\max}}{g_t^{\min}}$, which serves as an upper bound on the value of (4), is polynomially bounded by the input size.
We also need to impose on $\mathcal{S}_t, \mathcal{A}_t, g_t$ and f_t some properties to make z_t "easier" to approximate.

Condition 3. *At least one of the following cases holds.*

(i) **(Convex DP)** *The terminal state space $\mathcal{S}_{T+1}$ is a contiguous interval. For all $t = 1, \dots, T+1$ and $I \in \mathcal{S}_t$, the state space $\mathcal{S}_t$ and the action space $\mathcal{A}(I_t)$ are both contiguous intervals. g_{T+1} is integer-valued convex function over $\mathcal{S}_{T+1}$. For every $t = 1, \dots, T$ and fixed $d \in \mathcal{D}_t$, $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex and $g_t(\cdot, \cdot, d)$ 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 the transition function f_t can be expressed as $f_t(I_t, x_t, d) \equiv a(d)I_t + b(d)x_t + c(d)$, where g_t^I, g_t^x, u_t are univariate integer-valued convex functions, $a(\cdot), b(\cdot), c(\cdot)$ are integer-valued functions of d, and $b(d) \in \{-1, 0, 1\}$, for every d.*

(*ii*) **(Nondecreasing DP)** *g_{T+1} is nondecreasing. For every $t = 1, \ldots, T$, and fixed $d \in \mathcal{D}_t$, $g_t(\cdot, \cdot, d)$ and $f_t(\cdot, \cdot, d)$ are nondecreasing in their first variable. $f_t(\cdot, \cdot, d)$ is monotone in its second variable. g_t can be expressed as $g_t(\cdot, \cdot, d) \equiv g_t^a(\cdot, \cdot, d) + g_t^b(\cdot, \cdot, d)$, where $g_t^a(\cdot, \cdot, d), g_t^b(\cdot, \cdot, d)$ are nonnegative functions monotone in their second variable. Moreover, either z_t is nondecreasing for $t = 1, \ldots, T$, or $\forall I', I \in \mathcal{S}_t$ with $I' \leq I$, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$.*

(*iii*) **(Nonincreasing DP)** *g_{T+1} is nonincreasing. For every $t = 1, \ldots, T$, and fixed $d \in \mathcal{D}_t$, $g_t(\cdot, \cdot, d)$ is nonincreasing in its first variable, $f_t(\cdot, \cdot, d)$ is nondecreasing in its first variable. $f_t(\cdot, \cdot, d)$ is monotone in its second variable. g_t can be expressed as $g_t(\cdot, \cdot, d) \equiv g_t^a(\cdot, \cdot, d) + g_t^b(\cdot, \cdot, d)$, where $g_t^a(\cdot, \cdot, d), g_t^b(\cdot, \cdot, d)$ are nonnegative functions monotone in their second variable. Moreover, either z_t is nonincreasing for $t = 1, \ldots, T$, or $\forall I', I \in \mathcal{S}_t$ with $I' \leq I$, $\mathcal{A}_t(I') \subseteq \mathcal{A}_t(I)$.*

We call the DP formulation (5) *convex* whenever it satisfies Condition 3(*i*) and *monotone* whenever either one of the last two cases holds.

At a first glace one may think that Condition 3, with its three cases, is quite cumbersome. It is possibly due to our effort to formulate it in a general way. We try to justify this formulation by showing later on in the paper that each of its various cases has applications. Case (*i*) seems particularly "clumsy". Unfortunately, as we show in Theorem 8.2, none of the conditions about either $\mathcal{S}_t$, $\mathcal{A}_t(I)$ and f_t can be dropped.

We aim at providing an FPTAS for the optimal solution $z_1(I_1)$. Note that even in the very restrictive case where the number of the states of the system is a constant, computing the optimal solution by the optimality equation 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. In [Woe00], Woeginger designs a framework for deriving an FPTAS for deterministic DP. Among other 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 [Woe00]). We do not require this. For this reason our framework, when applied on deterministic DP, is not a special case of Woeginger's framework.
The main result of this paper, which we prove in Sections 8 and 9, is

Theorem 3.2. *Every stochastic minimization dynamic program that satisfies Conditions 1-3 admits an FPTAS.*

We also deal with maximization problems, where the DP optimality equation (5) is with "max" instead of "min". In order to do so, we reformulate Condition 3 for maximization problems.

Condition 4. *At least one of the following properties holds.*

(*i*) **(Concave DP)** *The terminal state space $\mathcal{S}_{T+1}$ is a contiguous interval. For all $t = 1, \ldots, T+1$ and $I \in \mathcal{S}_t$, the state space $\mathcal{S}_t$ and the action space $\mathcal{A}(I_t)$ are both contiguous intervals. g_{T+1} is integer-valued concave function over $\mathcal{S}_{T+1}$. For every $t = 1, \ldots, T$ and fixed $d \in \mathcal{D}_t$, $\mathcal{S}_t \otimes \mathcal{A}_t$ is integrally convex and $g_t(\cdot, \cdot, d)$ 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 the transition function f_t can be expressed as $f_t(I_t, x_t, d) \equiv a(d)I_t + b(d)x_t + c(d)$, where g_t^I, g_t^x, u_t are univariate integer-valued concave functions, $a(\cdot), b(\cdot), c(\cdot)$ are integer-valued functions of d, and $b(d) \in \{-1, 0, 1\}$, for every d.*

(*ii*) **(Nondecreasing DP)** *g_{T+1} is nondecreasing. For every $t = 1, \ldots, T$, and fixed $d \in \mathcal{D}_t$, $g_t(\cdot, \cdot, d)$ and $f_t(\cdot, \cdot, d)$ are nondecreasing in their first variable. $f_t(\cdot, \cdot, d)$ is monotone in its second variable. g_t can be expressed as $g_t(\cdot, \cdot, d) \equiv g_t^a(\cdot, \cdot, d) + g_t^b(\cdot, \cdot, d)$, where $g_t^a(\cdot, \cdot, d), g_t^b(\cdot, \cdot, d)$ are nonnegative functions monotone in their second variable. Moreover, either z_t is nondecreasing for $t = 1, \ldots, T$, or $\forall I', I \in \mathcal{S}_t$ with $I' \leq I$, $\mathcal{A}_t(I') \subseteq \mathcal{A}_t(I)$ holds.*

(*iii*) **(Nonincreasing DP)** g_{T+1} *is nonincreasing. For every* $t = 1, \dots, T$, *and fixed* $d \in \mathcal{D}_t$, $g_t(\cdot,\cdot,d) \equiv g_t^a(\cdot,\cdot,d) + g_t^b(\cdot,\cdot,d)$ *is nonincreasing in its first variable and* $g_t^a(\cdot,\cdot,d), g_t^b(\cdot,\cdot,d)$ *are nonnegative functions monotone in their second variable.* $f_t(\cdot,\cdot,d)$ *is nondecreasing in its first variable, and is monotone in its second variable. Moreover, either* z_t *is nonincreasing for* $t = 1, \dots, T$, *or* $\forall I', I \in \mathcal{S}_t$ *with* $I' \leq I$, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$ *holds.*

Theorem 3.3. *Every stochastic maximization dynamic program that satisfies Conditions 1-2 and 4 admits an FPTAS.*

In Section A in the Appendix we present in detail the 9 problems mentioned in the Introduction, and specifically show how to cast each of them as either monotone or convex DP.

4 K-approximation sets and functions

In this section we efficiently construct a succinct approximation of a given function $\varphi : D \rightarrow \mathbb{R}^+$ over a finite domain D. By *succinct* we mean that the space used for the representation of the approximation must be polylogarithmic in $|D|$ and $\frac{\varphi^{\max}}{\varphi^{\min}}$. By *efficient* we mean that the time to create the approximation function must be polylogarithmic in these two terms as well. We obtain such succinct approximations by means of constructing a special approximation set W for D (that we call a *K-approximation set of φ*, a notion introduced in [HKM+06]), and building upon it a step function $\hat{\varphi}$ that we call the *approximation of φ induced by W* in the following way: if $x \in W$, then $\hat{\varphi}(x) = \varphi(x)$, else $\hat{\varphi}(x) = \max\{\varphi(\text{prev}(x, D')), \varphi(\text{next}(x, D'))\}$. We show that $\hat{\varphi}$ is a K-approximation function of φ. Most functions do not admit a succinct K-approximation function. But some functions with nice structure can be approximated succinctly. We show that when φ is either unimodal, monotone or convex, it admits a succinct K-approximation.

Definition 4.1. *Let $K \geq 1$ and let $\varphi : D \rightarrow \mathbb{R}^+$ be a function, where $D \subset \mathbb{R}$ is a finite set. φ is called* succinct *if it admits a representation in space polylogarithmic in both $|D|$ and $\frac{\varphi^{\max}}{\varphi^{\min}}$. A subset of D is called* succinct *if its size is polylogarithmic in both these terms. We say that $\tilde{\varphi} : D \rightarrow \mathbb{R}^+$ is a* K-approximation function of φ *(or more briefly, a K-approximation of φ) if for all $x \in D$ we have $\varphi(x) \leq \tilde{\varphi}(x) \leq K\varphi(x)$. $\tilde{\varphi}$ is called a* succinct K-approximation of φ *if it is a succinct function that K-approximates φ. If such $\tilde{\varphi}$ can be constructed in time polylogarithmic in both $|D|$ and $\frac{\varphi^{\max}}{\varphi^{\min}}$, we call it* efficient.

Remark: 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. [1] *Let $K \geq 1$ and let $\varphi : D \rightarrow \mathbb{R}^+$ be a unimodal function defined over a finite domain of real numbers. We say that $W \subseteq D$ is a* K-approximation set of φ *if the following 3 properties are satisfied:*

1. $D^{\min}, D^{\max} \in W$;
2. *(boundness) for every $x \in W$ and $x \neq D^{\max}$, either $\mathit{next}(x, W) = \mathit{next}(x, D)$ or $\max\{\varphi(x), \varphi(\mathit{next}(x, W))\} \leq K \min\{\varphi(x), \varphi(\mathit{next}(x, W))\}$;*
3. *(locality) for every $x \in D \setminus W$ we have $\varphi(x) \leq \max\{\varphi(\mathit{prev}(x, W)), \varphi(\mathit{next}(x, W))\} \leq K\varphi(x)$.*

[1] The original definition of K-approximation sets as in [HKM+06, HKL+08] (which the later called weak K-approximation sets) required $\arg\min \varphi \in W$ and that W satisfies the first two properties of Definition 4.2. Hence, due to Proposition 4.3, our modified definition of K-approximation sets is a generalization of the original definition. In our current definition we do not request W to include $\arg\min \varphi$. Due to this modification we get stronger results, e.g., Proposition 6.3 and Property 4, each of which does not necessarily hold for approximation sets under the original definition.

There is an easy way to satisfy locality:

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

Clearly, if there exists a succinct 1-approximation set of φ then φ is succinct. Whenever φ does not admit a succinct 1-approximation set, we will be interested in finding succinct K-approximations of φ for $K > 1$, by constructing succinct K-approximation sets for φ. Function ApxSet (Algorithm 1) is a *canonical algorithm* for constructing a K-approximation set.

1: **Function ApxSet**(φ, D, K)
2: $x \leftarrow D^{\max}$
3: $W \leftarrow \{D^{\min}, D^{\max}\}$
4: **while** $x > D^{\min}$ **do**
5: $\quad z \leftarrow \min\left\{\text{prev}(x, D), \min\{y \in D \mid K\varphi(y) \geq \varphi(x)\}\right\}$
6: $\quad W \leftarrow W \cup \{x\}$
7: **end while**
8: Return W

Algorithm 1: Constructing a canonical K-approximation set for a nondecreasing φ over D.

If φ is nonincreasing we replace Step 5 by
$5a:\quad x \leftarrow \min\left\{\text{prev}(x, D), \min\{y \in D \mid K\varphi(x) \geq \varphi(y)\}\right\}$
If φ is unimodal and $x^* = \arg\min\{\varphi(x) \mid x \in D\}$, then we replace Step 5 by
5b: **if** $x > x^*$ **then** $x \leftarrow \min\left\{\text{prev}(x, D), \min\{y \in D \mid y \geq x^* \text{ and } K\varphi(y) \geq \varphi(x)\}\right\}$
5c: **else** $x \leftarrow \min\left\{\text{prev}(x, D), \min\{y \in D \mid K\varphi(x) \geq \varphi(y)\}\right\}$
We refer to the K-approximation set obtained by Function ApxSet as the *canonical K-approximation set.*

K-approximation functions are too general to maintain the structure of the function they approximate. We conclude this section with the next theorem that tells us that for monotone and convex functions, as well as unimodal function with a given minimizer, it is possible to efficiently build succinct K-approximation functions that preserve the same structure. The proof of this theorem is straighforward by using binary search, and therefore is omitted.

Theorem 4.4. *Let $\varphi : D \rightarrow \mathbb{R}^+$ be a unimodal function over a finite domain of real numbers, and let $x^* = \arg\min\{\varphi(x) \mid x \in D\}$. Let t_φ be an upper bound on the time needed to evaluate $\varphi(x)$, for any $x \in D$. Then for every $K > 1$ the following holds.*

1. *Function ApxSet computes a K-approximation set W of φ of cardinality $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$ in $O(t_\varphi(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}}) \log |D|)$ time,*

2. *$\hat{\varphi}$, the approximation of φ induced by W, is a K-approximation of φ, and is stored in a sorted array $(x, \varphi(x))$, $x \in W$ so $\varphi(x)$ can be determined in $O(\log |W|)$ time, using binary search,*

3. *$\{D^{\max}\} \cup \{x, next(x, D) \mid D^{\max} > x \in W\}$ is a 1-approximation set of $\hat{\varphi}$,*

4. *if φ is unimodal or monotone, then so is $\hat{\varphi}$,*

5. *if φ is convex over D, then so is the linear extension of φ induced by W.*

Note that $\varphi_{\min} > 0$ whenever $\varphi_{\max} > 0$, and so the ratio is well defined when the function is non-zero. Also, we could write expressions in terms of ϵ by observing that $\frac{1}{\log K} = O(\frac{1}{\epsilon})$.

5 Calculus of K-approximation functions

In this section we develop a theory, based on the notion of K-approximation sets and functions. The theory consists of a set of computational rules for manipulating K-approximation functions.

The following proposition, which we call Calculus of K-approximation Functions, follows directly from the definition of K-approximation functions, and its proof is therefore omitted. (Properties 3 and 4 can also be derived from [HKM$^+$06].)

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 domain D, and let $\tilde{\varphi}_i : D \to \mathbb{R}$ be a K_i-approximation of φ_i. Let $\psi_1 : D \to D$, and 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.*

The following simple propositions will be useful in Section 8, when "plugging-in" K-approximation functions into convex dynamic programming.

Proposition 5.2 ([HKM$^+$06]). *Let $K > 1$ and φ be a nonnegative integer-valued function. If φ' is a (general) K-approximation of φ then $\lfloor\varphi'\rfloor$ is an integer-valued K-approximation of φ.*

This proposition is due to $\varphi \leq \lfloor\varphi'\rfloor \leq \varphi' \leq K\varphi$, where the first inequality derives from the integrality of φ and since φ' is a K-approximation of φ.

Proposition 5.3 ([HKM$^+$06]). *Let $K > 1$ and φ be a nonnegative integer-valued function. If $\check{\varphi}$ is a convex K-approximation of φ with $\arg\min \varphi = x^*$, then $\lfloor\check{\varphi}\rfloor$ is a unimodal integer-valued K-approximation of φ with the same $\arg\min$.*

Proof. The monotonicity of the floor function coupled with the convexity of $\check{\varphi}$ implies that $\lfloor\check{\varphi}\rfloor$ is a unimodal function that is minimized at x^*. Function $\lfloor\check{\varphi}\rfloor$ is an integer-valued K-approximation of φ due to the previous proposition. □

Proposition 5.4 (minimization of summation of composition). *Let D and E be arbitrary domains, $n \in \mathbb{N}$ be an arbitrary positive integer and $K_i \geq 1$, $i = 1, \ldots, n$ be arbitrary reals. Let $\varphi_i : D \to \mathbb{R}^+$, let $\tilde{\varphi}_i$ be a K_i-approximation of it, and let $\psi_i : D \times E \to D$, $\forall\, i = 1, \ldots, n$. Then*

$$\tilde{\varphi}(x) := \min_{y \in E}\{\sum_{i=1}^{n} \tilde{\varphi}_i(\psi_i(x, y))\} \tag{7}$$

is a $\max\{K_1, \ldots, K_n\}$-approximation of $\varphi(x) := \min_{y \in E}\{\sum_{i=1}^{n} \varphi_i(\psi_i(x, y))\}$.

The proof of Proposition 5.4 (as well as its name), is due to summation of approximation, minimization of approximation, and composition of approximation (Properties 3,4 and 6 in Proposition 5.1).

Since the cardinality of E may be "big", applying Proposition 5.4 and calculating the minimum over all of the elements of E may take time exponential in the input size. For this reason we would like to "approximate" E succinctly in such a way that performing the minimization in (7) under this approximated set, instead of under the entire E, will result in a succinct approximation of $\tilde{\varphi}_3$. This is possible whenever the functions we want to approximate are monotone.

Theorem 5.5. *For $i = 1, \dots, m, \dots, n$ let $L_i \geq 1$, $K_i > 1$ and let $\varphi_i : D \to \mathbb{R}^+$ be a function over a finite domain $D \subset \mathbb{R}$. Let $\tilde{\varphi}_i : D \to \mathbb{R}$ be an L_i-approximation of φ_i. For every fixed $x \in D$, let $\psi_i : D \times E \to D$ be a function such that $\tilde{\varphi}_i(\psi_i(x, \cdot))$ is monotone over a totally ordered domain E. Suppose $\tilde{\varphi}_i(\psi_i(x, \cdot))$ are monotone in one sense for $i = 1, \dots, m$ (e.g., nondecreasing) and are monotone in the other sense for $i = m+1, \dots, n$ (e.g., nonincreasing). If $W_i(x) \subseteq D$ is a K_i-approximation set of $\tilde{\varphi}_i(\psi_i(x, \cdot))$, then*

$$\tilde{\varphi}(x) := \min_{y \in \cup_{i=1}^n W_i(x)} \sum_{i=1}^n \tilde{\varphi}_i(\psi_i(x, y))$$

is an $L := \max\{K_1 L_1, \dots, K_m L_m, L_{m+1}, \dots, L_n\}$-approximation of $\varphi(x) := \min_{y \in E} \sum_{i=1}^n \varphi_i(\psi_i(x, y))$.

Proof. Let us fix x. Let $y^@ \in \cup_{i=1}^n W_i(x)$ be a realizer of $\tilde{\varphi}(x)$, i.e., $\tilde{\varphi}(x) = \sum_{i=1}^n \tilde{\varphi}_i(\psi_i(x, y^@))$. Due to composition of approximation (Property 4 in Proposition 5.1), $\tilde{\varphi}_i(\psi_i(x, \cdot))$ is an L_i-approximation of $\varphi_i(\psi_i(x, \cdot))$, for $i = 1, \dots, n$, and therefore

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

Let y^* be the smallest realizer of $\varphi(x)$. Due to symmetry arguments we can assume without loss of generality that $\tilde{\varphi}_1(\psi_1(x, \cdot))$ is nondecreasing. By the definition of $W_i(x)$ there exists $y_i' \in W_i(x)$ such that $y_i' \geq y^*$ and $\tilde{\varphi}_i(\psi_i(x, y^*)) \leq \tilde{\varphi}_i(\psi_i(x, y_i')) \leq K_i \tilde{\varphi}_i(\psi_i(x, y^*))$, $\forall i = 1, \dots, m,$. Let $y' := \min\{y_i' \mid i = 1, \dots, m\}$. By the monotonicity of $\tilde{\varphi}_i(\psi_i(x, \cdot))$ we get

$$\tilde{\varphi}_i(\psi_i(x, y')) \leq \tilde{\varphi}_i(\psi_i(x, y_i')) \leq K_i \tilde{\varphi}_i(\psi_i(x, y^*)) \leq K_i L_i \varphi_i(\psi_i(x, y^*)), \ \forall i = 1, \dots, m, \tag{8}$$

where the last inequality is since $\tilde{\varphi}_i(\psi_i(x, \cdot))$ is an L_i-approximation of $\varphi_i(\psi_i(x, \cdot))$.

Since $\tilde{\varphi}_i(\psi_i(x, \cdot))$ are nonincreasing for $i = m+1, \dots, n$, from the definition of K-approximation functions we get

$$\tilde{\varphi}_i(\psi_i(x, y')) \leq \tilde{\varphi}_i(\psi_i(x, y^*)) \leq L_i \varphi_i(\psi_i(x, y^*)), \ \forall i = m+1, \dots, n. \tag{9}$$

We conclude the proof by combining (8) with (9):

$$\tilde{\varphi}(x) \leq \sum_{i=1}^n \tilde{\varphi}_i(\psi_i(x, y')) \leq \sum_{i=1}^m K_i L_i \varphi_i(\psi_i(x, y^*)) + \sum_{i=m+1}^n L_i \varphi_i(\psi_i(x, y^*)) \leq \max\{K_1 L_1, \dots, K_m L_m, L_{m+1}, \dots, L_n\} \varphi(x).$$

∎

Remark: Note that while Theorem 5.5 provides an upper bound to the approximation ratio of $\tilde{\varphi}$, $\tilde{\varphi}$ is not necessarily monotone. But by linearly scanning $\tilde{\varphi}$ and using the monotonicity of the original function φ, one can build a monotone L-approximation function for φ. We use this approach in Section 9.

6 Calculus of K-approximation sets

In this section we develop a calculus based on the notion of K-approximation sets and functions. 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. We present this calculus in a way symmetric to the presentation of Section 5, and give the somewhat involved proofs in Section B in the Appendix.

Proposition 6.1 (Calculus of K-approximation Sets of unimodal functions)**.** *Let $K_1, K_2 \geq 1$, let $\varphi_1, \varphi_2 : D \to \mathbb{R}^+$ be unimodal functions over a finite domain D of real numbers. Let W_i be a K_i-approximation set of φ_i, for $i = 1, 2$. Let $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ be a (weakly) monotone function, and let $\alpha, \beta \in \mathbb{R}^+$. The following properties hold:*

1. *(monotonicity of approximation sets) every superset $W_1 \subset W' \subset D$ is a K_1-approximation set of φ_1,*
2. *(composition of approximation sets) $\psi^{-1}(W_1) = \{\max_i, \min_i \mid \psi(i) \in W_1\}$ is a K_1-approximation set of $\varphi_1(\psi)$,*
3. *(linearity of approximation sets) W_1 is a K_1-approximation set of $\alpha + \beta\varphi_1$.*
4. *(maximization of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\max\{\varphi_1, \varphi_2\}$.*

We could have approximation sets W_1 and W_2, both containing the argmin. But it is possible that $W_1 \cup W_2$ does not contain the argmin of $\max\{\varphi_1, \varphi_2\}$. There are two ways around this. One is to calculate the argmin directly. (This is actually pretty easy to do using binary search since we know where the argmin of each of the functions occurs. But it means that we cannot rely entirely on the approximations to φ_1 and φ_2.) The other alternative is to relax the restriction in the original definition of K-approximation sets of [HKM+06, HKL+08] that we need to have the argmin, and make it sufficient to have an approximated argmin. This is the approach that we take here (and it differs from previous work), and it is more general. In particular, it does not require that we have access to the original function.

Proposition 6.2 (Calculus of K-approximation Sets of monotone functions)**.** *Let $K_1, K_2 > 1$, and let $\varphi_1, \varphi_2 : D \to \mathbb{R}^+$ be monotone functions of the same kind (i.e., either both are nondecreasing or both are nonincreasing) over 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)[2] 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.*

Proposition 6.3 (Calculus of K-approximation Sets of convex functions)**.** *Let $K_1, K_2 > 1$, let $\varphi_1, \varphi_2 : D \to \mathbb{Z}^+$ 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:*

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

We note that it can happen that $\arg\min(\varphi_1 + \varphi_2) \notin \{\arg\min \varphi_1, \arg\min \varphi_2\}$. We also 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

[2] Note from Nir: To show that an extra error is indeed obligatory

minimization of approximation sets since the minimum of two convex functions is not necessarily convex or even unimodal.
We conclude with the following simple proposition

Proposition 6.4 ([HKM+06])**.** *Let $K > 1$ and let φ be a convex function. Let W be a K-approximation set of φ, and $\hat{\varphi}$ be the approximation of φ induced by W. Then the convex extension of $\hat{\varphi}$ induced by W is a convex K-approximation of φ.*

Proof. This proposition is true because the convex extension of $\hat{\varphi}$ induced by W is the greatest convex function which does not lie above $\hat{\varphi}$, and the fact that φ itself is a convex function (i.e., the convex extension of $\hat{\varphi}$ induced by W is "sandwiched" between $\hat{\varphi}$ from above, and φ from below). □

7 From K-approximation sets and functions to dynamic programming

The following propositions link the notion of K-approximation sets with dynamic programming. The first one deals with Convex DP.

Proposition 7.1. *Suppose the DP formulation* (5) *is convex (so Condition 3(i) is satisfied). Let $1 \leq K'_1 \leq K'_2$, $I_t \in \mathcal{S}_t$ be fixed, and let $\tilde{g}_t(I_t, \cdot, d_{t,i})$ be a convex K'_1-approximation 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'_2-approximation of z_{t+1}, and let*

$$\tilde{G}_t(I_t, \cdot) := E_{D_t}\tilde{g}_t(I_t, \cdot, D_t), \quad \tilde{Z}F_{t+1}(I_t, \cdot) := E_{D_t}\tilde{z}_{t+1}(f_t(I_t, \cdot, D_t)).$$

Then

$$\bar{z}_t(I_t) := \min_{x_t \in \mathcal{A}(I_t)} \{\tilde{G}_t(I_t, x_t) + \tilde{Z}F_{t+1}(I_t, x_t)\} \tag{10}$$

is a K'_2-approximation of z_t that can be calculated in $O(\log(|\mathcal{A}(I_t)|)n_t(t_{\tilde{g}} + t_{\tilde{z}}))$ time for each value of I_t.

Proof. We apply Proposition 5.4 with $n = 2n_t$, $\varphi_i = g_t(I_t, \cdot, d_{t,i})$, $\psi(I_t, x_t) = x_t$, $K_i = K'_1$, $\forall i = 1, \dots, n_t$, and $\varphi_i = z_{t+1}$, $\psi_i = f_t(I_t, \cdot, d_{i-n_t})$, $K_j = K'_2$, $\forall\ i = n_t + 1, \dots, 2n_t$. Therefore, $\bar{z}_t(I_t)$ is a K'_2-approximation of z_t. Note that for fixed d_t, $f(I_t, \cdot, d_t)$ is a linear function with slope in $\{-1, 0, 1\}$. Therefore $\tilde{z}_{t+1}(f_t(I_t, \cdot, d_t))$ is a convex function. Moreover, since a convex combination of convex functions is such as well, we get that $\tilde{G}_t(I_t, \cdot) + \tilde{Z}F_{t+1}(I_t, \cdot)$ is a convex function, so its minimum can be found by binary search over the contiguous interval $\mathcal{A}(I_t)$. We conclude the proof by using linearity of expectation. □

The next proposition deals with monotone DP.

Proposition 7.2. *Suppose the DP formulation* (5) *is monotone (so either Condition 3(ii) or Condition 3(iii) is satisfied). Let $K'_1, L'_1, K'_2, L'_2 \geq 1$, with $K'_1 L'_1 \leq L'_2$, let $t \in [1, \dots, T]$, and $I_t \in \mathcal{S}_t$ be fixed. Let $g_t(I_t, \cdot, d_{t,i}) \equiv g^a_t(I_t, \cdot, d_{t,i}) + g^b_t(I_t, \cdot, d_{t,i})$. Let $\tilde{g}^b_t(I_t, \cdot, d_{t,i})$ be a monotone L'_1-approximation of $g^b_t(I_t, \cdot, d_{t,i})$ in the same direction as z_{t+1}, and $W^b_{t,i}$ be a K'_1-approximation set of it. Let $\tilde{g}^a_t(I_t, \cdot, d_{t,i})$ be a L'_1-approximation of $g^a_t(I_t, \cdot, d_{t,i})$ monotone in the other direction of z_{t+1}, and $W^a_{t,i}$ be a K'_1-approximation set of it. Let $\tilde{z}_{t+1}$ be a monotone L'_2-approximation of z_{t+1}, and W^c_t be a K'_2-approximation of it. Let*

$$\tilde{A}_t(I_t, \cdot) := E_{D_t}\tilde{g}^a_t(I_t, \cdot, D_t), \quad \tilde{B}_t(I_t, \cdot) := E_{D_t}\tilde{g}^b_t(I_t, \cdot, D_t), \quad \tilde{Z}F_{t+1}(I_t, \cdot) := E_{D_t}\tilde{z}_{t+1}(f_t(I_t, \cdot, D_t)),$$

and $W = \bigcup_{i=1}^{n_t}(W^a_{t,i} \cup W^b_{t,i} \cup f^{-1}(I_t, W^c_t(I_t), d_{t,i}))$. Then

$$\bar{z}_t(I_t) = \min_{x_t \in W} \{\tilde{A}_t(I_t, x_t) + \tilde{B}_t(I_t, x_t) + \tilde{Z}F_{t+1}(I_t, x_t)\} \tag{11}$$

is an L'_2-approximation of z_t that can be calculated in $O(|W|)$ time.

Proof. We apply Theorem 5.5 with $m = n_t,\ n = 3m,\ \varphi_i = g_t^a(I_t, \cdot, d_{t,i}),\ K_i = K_1',\ \forall i = 1, \dots, m,\ \varphi_i = g_t^b(I_t, \cdot, d_{t,i}),\ K_i = K_1',\ \forall i = m+1, \dots, 2m$, and $\varphi_i = z_{t+1},\ K_i = K_2',\ \forall i = 2m+1, \dots, n$. $\psi_i(I_t, x) = x,\ \forall\ i = 1, \dots, 2m$ and $\psi_i(I_t, x) = f_t(I_t, x, d_{i-2m})$ for $i = 2m+1, \dots, n$. Therefore, $\bar{z}_t(I_t)$ is an L_2'-approximation of z_t. The minimum can be found in the promised time by scanning all the elements of W. □

We note that the above propositions show that by the Calculus of K-approximation Sets and Functions approximating the stochastic DP recursion (5) is essentially as hard as approximating a deterministic DP recursion (i.e., where the random variables are constants with probability one), with an additional complexity factor of only n_t (the size of the support of the random variable).

This situation is substantially different from the one with the corresponding exact problems. For example, [HKM$^+$06] showed that the stochastic single-item inventory control problem (which we formulate in Section A.2.3 as a convex DP) is #P-hard, while it is known that the corresponding deterministic problem is in P and is easily solved by linear programming.[3].

In the case of monotone DP, we note that in many of our applications[4] the single-period cost function $g_t(I_t, x_t, D_t)$ can be expressed as the sum of a fixed number of monotone univariate functions, e.g., $g_t(I_t, x_t, D_t) = g_t^1(f_t^1(I_t, x_t, D_t)) + g_t^2(f_t^2(I_t, x_t, D_t))$, where $g_t^i, f_t^i(\cdot, x_t, D_t)$ and $f_t^i(I_t, \cdot, D_t)$ are all monotone. In these cases it is sufficient to ask in Proposition 7.2 to have K-approximation sets for each of these univariate functions (exactly as we have one such set W_t^c for z_{t+1}), instead of calculating such a set for each combination of I_t and $d_{t,i}$. This results in saving in the running time of the FPTAS we give below.

8 An FPTAS for Convex DP

In this section we develop an FPTAS for any dynamic program that satisfies Conditions 1-3(i). Our FPTAS is as follows

Procedure FPTASConvexDP
$K \leftarrow 1 + \frac{\epsilon}{2(T+1)},\quad W_{T+1} \leftarrow$**ApxSet**$(g_{T+1}, \mathcal{S}_{T+1}, K)$
Let $\hat{z}_{T+1}$ be the approximation of g_{T+1} induced by W_{T+1}
Let $\tilde{z}_{T+1}$ be the convex extension of $\hat{z}_{T+1}$ 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 (10) */
  $W_t \leftarrow$**ApxSet**$(-\lfloor M_t \bar{z}_t \rfloor, \mathcal{S}_t \cap \{x \mid x \le x^*\}, K) \bigcup$ **ApxSet**$(\lfloor M_t \bar{z}_t \rfloor, \mathcal{S}_t \cap \{x \mid x \ge x^*\}, K)$
  Let $\hat{z}_t$ be the approximation of $\frac{\lfloor M_t \bar{z}_t \rfloor}{M_t}$ induced by W_t
  Let $\tilde{z}_t$ be the convex extension of $\hat{z}_t$ induced by W_t
end for

Algorithm 2: FPTAS for Convex DP.

In order to prove that this is indeed an FPTAS, we need several properties to remain invariant throughout the execution of the algorithm.

Theorem 8.1 (Convexity Invariant)**.** *Suppose the DP formulation* (5) *is convex (i.e., the sets and functions involved satisfy Condition 3(i)). Then for every* $t = 1, \dots, T+1$*, function* z_t *in (5) and* $\bar{z}_t$ *in* (10) *are convex over* $\mathcal{S}_t$*.*

Proof. We give a proof for z_t. The proof for $\bar{z}_t$ is similar. Our proof follows the lines of, and generalizes a proof in [HKM$^+$06]. We use backward induction. We consider first the base case of $t = T+1$. Since g_{T+1}

[3] Note from Nir: To improve the above two paragraphs

[4] Note from Nir: to mention the applications where this actually happens

is convex by Condition 3(i), so is z_{T+1}. We assume by induction that z_{t+1} is convex and prove for z_t. It suffices to show that $2z_t(I) \leq z_t(I+1) + z_z(I-1)$ for all $I \in \mathcal{S}_t$. Consider now a fixed value of I. Since a convex combination of convex functions is a convex function, it is enough to prove for the case where the random variable D_t is fixed. For convenience we slightly abuse our notation and omit D_t from the domain of g_t and f_t. Hence the formula reads

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

Let us define $c = g_t^x, y = u_t + z_{t+1}$ and $f = f_t$. Note that c is a univariate convex function over the contiguous interval $\mathcal{A}_t(I)$ and y is a univariate convex function over the contiguous interval $\mathcal{S}_{t+1}$. Since g_t^I is convex, it suffices to prove that

$$z(I) = \min_{x \in \mathcal{A}_t(I)} \{c(x) + y(f(I,x))\}$$

is a convex function. We assume that $f(I,x) = aI + x$. The proof for $f(I,x) = aI - x$ and $f(I,x) = aI$ is similar. We choose $(I-1, x')$; $(I+1, x'') \in \mathcal{S}_t \otimes \mathcal{A}_t$, so that

$$z(I-1) = c(x') + y(aI - a + x'); \quad \text{and} \quad z(I+1) = c(x'') + y(aI + a + x'').$$

Let $x^* = \lfloor \frac{x'+x''}{2} \rfloor$. Let us assume for a moment that $x^* \in \mathcal{A}_t(I)$. This means that $z(I) \leq c(x^*) + y(aI + x^*)$. Therefore, $2z(I) - (z(I-1) + z(I+1)) \leq 2c(x^*) - (c(x') + c(x'')) + 2y(aI + x^*) - (y(aI - a + x') + y(aI + a + x'')) \leq 0$. The latter inequality holds by the convexity of c and y. This proves that z is indeed convex.

It remains to prove that $x^* \in \mathcal{A}(I)$, i.e., $(I, x^*) \in \mathcal{S}_t \otimes \mathcal{A}_t$. To see this, it suffices to prove that the minimal integrally-convex set S that contains $(I-1, x'), (I+1, x'')$, also contains (I, x^*). Without loss of generality, let us assume that $I = 1, x' = 0$ and $x'' \geq x'$. We distinguish between 4 cases (see Figure 3):

Case 1. $x'' = 0$. Clearly $S = \{(0,0);\ (1,0);\ (2,0)\}$ and we are done since $(I, x^*) = (1,0)$.

Case 2. $x'' = 1$. Recall that since S is an integrally-convex set, it is the intersection of a convex polytope consisting of edges with slopes in $\{-\infty, -1, 0, 1, \infty\}$, and the integer lattice. Therefore $S = \{(0,0);\ (1,0);\ (1,1);\ (2,1)\}$. Since $(I, x^*) = (1,0)$ we are done.

Case 3. $x'' = 2$. Clearly $S = \{(0,0);\ (1,1);\ (2,2)\}$ and we are done since $(I, x^*) = (1,1)$.

Case 4. $x'' \geq 3$. Then S is bounded below by the line $y(x) = x$, and is bounded above by the line $y(x) = x + x'' - 2$, so again $(I, x^*) \in S$.

□

Figure 3: The 4 cases in the proof of Theorem 8.1.

We note that if $\mathcal{S}_t \otimes \mathcal{A}_t$ were not integrally convex, then it may happen that $x^* \notin \mathcal{A}_t(I)$. For instance, let $\mathcal{S}_t \otimes \mathcal{A}_t$ be the middle set in Figure 2, and assume the bottom-left point is (0,0). Clearly, $x^* = 0$ but $(1,0) \notin \mathcal{S}_t \otimes \mathcal{A}_t$.

We note also that if we drop from Condition 3(i) the requirement that the coefficients of the second variable of f are -1,0 or 1, then the Convexity Invariant does not necessarily hold. Consider the following 1-time period example. $T = 1$, $\mathcal{S}_1 = [0, \dots, 10]$, $\mathcal{S}_2 = [-8, \dots, 10]$, $\mathcal{A}_1(I) = \{0, 1, 2\}$, $g_2(I) = |I|$, $g_1 \equiv 0$ and $f_1(I, x) = I - 4x$. ($\mathcal{S}_1 \otimes \mathcal{A}_1$ is integrally convex as the intersection of a rectangle with the integer lattice.) Then the global minima of $z_1(I)$ are at 0,4,8, with value 0 each, and its global maxima are at 2,6,10 with value 2 each.

Also, we note that neither the requirement that $\mathcal{A}_t(I)$ is a contiguous interval can be dropped. Consider by negation the following example. $T = 1$, $\mathcal{S}_1 = [0, \dots, 10]$, $\mathcal{S}_2 = [-8, \dots, 10]$, $\mathcal{A}_1(I) = [0, 4, 8]$, $g_2(I) = |I|$, $g_1 \equiv 0$ and $f_1(I, x) = I - x$. (This time $\mathcal{S}_1 \otimes \mathcal{A}_1$ is not integrally convex.) We get that $z_1(I)$ equals the one of the previous example.

We last note that even if we could somehow "bypass" the Convexity Invariant, it is unlikely that we could develop an FPTAS for these cases, because of the theorem below, which we prove in the Appendix.

Theorem 8.2. *A convex DP where either $b \notin \{-1, 0, 1\}$, or the action space is not a contiguous interval, does not admit an FPTAS, unless P=NP.*

Proposition 8.3 (Integrality Invariant)**.** *Suppose the DP formulation* (5) *is convex (i.e., the sets and functions involved satisfy Condition 3(i)). Then for every $t = 1, ..., T + 1$ and I_t, $M_t z_t(I_t)$ is a nonnegative integer.*

Proof. We prove the integrality of $M_t z_t(I_t)$ by induction. The base case holds since $z_{T+1} \equiv g_{T+1}$, and g_{T+1} is an integer by Condition 3(i). The induction hypothesis is that $M_{t+1} z_{t+1}$ is an integer, and we next prove that also is $M_t g_t$. Due to (5) and (6) we have

$$z_t(I_t) = \frac{1}{Q_t} \min_{x_t} \sum_{j=1}^{n_t} q_{t,j} \{g_t(I_t, x_t, d_{t,j}) + z_{t+1}(f_t(I_t, x_t, d_{t,j}))\}.$$

Since g_t is an integer-valued function, and since by the induction hypothesis $M_{t+1} z_{t+1}$ is an integer-valued functions, $M_t z_t$ is an integer-valued function as well. □

We are now ready to prove the section's main result.

Theorem 8.4 (FPTAS for Convex DP)**.** *Given a dynamic program that satisfies Conditions 1,2 and 3(i), and $1 > \epsilon > 0$, 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 9 in the last iteration of Algorithm 2. Moreover, Algorithm 2 runs in time polynomial in both $\frac{1}{\epsilon}$ and the (binary) input size.*

Proof. We first show that the algorithm is well defined. The Convexity Invariant, Theorem 8.1, assures that all the $\bar{z}_t$'s are convex nonnegative functions. Therefore, $-\bar{z}_t$ and $\bar{z}_t$ are nondecreasing over $\mathcal{S}_t \cap \{x \mid x \leq x^*\}$ and $\mathcal{S}_t \cap \{x \mid x \geq x^*\}$, respectively. Hence all calls to function *ApxSet* are well defined.

We next show the correctness of the algorithm. Let $K := 1 + \frac{\epsilon}{2(T+1)}$. We first show that it suffices to prove that $\tilde{z}_1$ is a K^{T+1}-approximation of z_1. By the DP (exact) optimality equation, Theorem 3.1, the optimal cost $z^*(I_1)$ is equal to $z_1(I_1)$. By the definition of K-approximation functions (Definition 4.2), $z^*(I_1) \leq z_1(I_1) \leq (1 + \frac{\epsilon}{2(T+1)})^{T+1} z^*(I_1)$. Thus, due to the inequality $(1 + \frac{x}{n})^n \leq 1 + 2x$, which holds for every $0 \leq x \leq 1$, we get that $z^*(I_1) \leq \tilde{z}_1(I_1) \leq (1+\epsilon) z^*(I_1)$.

We next prove by induction that for every $t = 1, \dots, T+1$, $\tilde{z}_t$ is a K^{T+2-t}-approximation of z_t. The base case of $t = T+1$ follows from Theorem 4.4. The induction hypothesis is that $\tilde{z}_{t+1}$ is a convex K^{T+1-t}-approximation of z_{t+1}. We next prove that $\tilde{z}_t$ is a convex K^{T+2-t}-approximation of z_t. By applying Proposition 7.1 with $K'_1 = 1$ and $K'_2 = K^{T+1-t}$, and using Theorem 8.1, we get that $\bar{z}_t$ is a convex K^{T+1-t}-approximation of z_t. In this way, finding x^* in Step 6 can be done efficiently by performing binary search.

Considering Step 7, and applying the Integrality Invariant (Proposition 8.3) together with Proposition 5.3, we get that $\frac{\lfloor M_t \bar{z}_t \rfloor}{M_t}$ is a unimodal K^{T+1-t}-approximation of z_t which is minimized at x^*. Due to Theorem 4.4 and the definition of W_t, $\hat{z}_t$ in Step 8 is a unimodal K^{T+2-t}-approximation of z_t that attains its minimum at x^*. (Note that by linearity of approximation sets (Proposition 6.1) W_t is a K-approximation set of $\frac{\lfloor M_t \bar{z}_t \rfloor}{M_t}$.) By Proposition 6.4, $\tilde{z}_t$ in Step 9 is a convex K^{T+2-t}-approximation of z_t that attains its minimum at x^*, as needed.

It remains to prove that the algorithm runs in time polynomial in both the input size and $\frac{1}{\epsilon}$. Recall that $U_D^{\mathcal{S}} = \max_{t=1,\dots,T+1} |\mathcal{S}_t|$ is the size of the maximal state space, $U_D^{\mathcal{A}} = \max_{t=1,\dots,T} |\mathcal{A}_t|$ is the size of the maximal action space, and $U_R = \max_t \max_{I \in I_t,\, x \in \mathcal{A}_t(I),\, d \in \mathcal{D}_t} g_t(I, x, d)$ is the maximal single-period cost value. By Conditions 1 and 2, $\log U^{\mathcal{S}}, \log U^{\mathcal{A}}$ and $\log U_R$ are all polynomially bounded by the input size. For ease of exposition let us assume that all these values are at least 2. By Theorem 4.4, Steps 2-4 of Algorithm 2 are executed in $O(t_g \log_K U_R \log U_D^{\mathcal{S}})$ time, where t_g is the time needed to evaluate g. Clearly, the "for loop" is executed T times. Considering Step 6, by the Convexity Invariant (Theorem 8.1) $\bar{z}_t$ is convex so we can apply binary search over the state space $U_D^{\mathcal{S}}$ in order to find the argmin, i.e., in $O(t_{\bar{z}_t} \log U_D^{\mathcal{S}})$ time, where $t_{\bar{z}_t}$ is the time needed to evaluate $\bar{z}_t$. By Theorem 4.4, Step 7 takes $O(t_{\bar{z}_t} \log_K U_R \log U_D^{\mathcal{S}})$ time. Therefore this step is the most time consuming in the for loop. By Proposition 7.1 each evaluation of $\bar{z}_t$ is done by performing binary search over the action space $U_D^{\mathcal{A}_t}$ in $O(n^*(t_g + t_f + t_{\tilde{z}_{t+1}}) \log U_D^{\mathcal{A}})$ time. As $\tilde{z}_{t+1}$ is succinctly stored in a sorted array in size not larger than the array for storing $\hat{z}_t$ which is by Theorem 4.4 $O(\log_K U_R)$, we conclude that the running time of each query of $\bar{z}_t$ is $O(n^*(t_g + t_f + \log\log_K U_R) \log U_D^{\mathcal{A}})$. Therefore, the running time of the entire algorithm totals to

$$O(Tn^*(t_g + t_f + \log\log_K U_R) \log_K U_R \log U_D^{\mathcal{S}} \log U_D^{\mathcal{A}}).$$

Without loss of generality we can assume that $\epsilon < 1$ which implies that $K < 2$, so $O(\log_K U_R) = O(\frac{\log U_R}{K-1})$. Replacing K with $1 + \frac{\epsilon}{2(T+1)}$, we conclude that the running time of the algorithm is

$$O((t_g + t_f + \log(\frac{T}{\epsilon} \log U_R)) \frac{n^* T^2}{\epsilon} \log U_R \log U_D^{\mathcal{S}} \log U_D^{\mathcal{A}}). \quad (12)$$

Since U_R depends linearly on the multiplying factor M_1, and since $\log U_D^{\mathcal{S}}, \log U_D^{\mathcal{A}}$ and $\log M_1$ are all polynomially bounded by the input size, the algorithm runs in polynomial time in both the input size and $\frac{1}{\epsilon}$. □

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

9 An FPTAS for Monotone DP

In this section we present an FPTAS for nondecreasing DP. The FPTAS for nonincreasing DP is analogous.

Procedure FPTASNondecreasingDP
$K \leftarrow 1 + \frac{\epsilon}{2(T+1)}$, $z_{T+1} \leftarrow g_{T+1}$ and $W_{T+1} \leftarrow$**ApxSet**$(z_{T+1}, \mathcal{S}_{T+1}, K)$
Let $\tilde{z}_{T+1} := \hat{z}_{T+1}$ be the approximation of z_{T+1} induced by W_{T+1}
for $t := T$ **downto** 1 **do**
    $W_t \leftarrow$**SubSet**$(\bar{z}_t, \mathcal{S}_t, K)$ /* $\bar{z}_t$ is as defined in (11) */
    Let $\tilde{z}_t$ be the maximal nondecreasing function that is bounded above by $\bar{z}$ on W_t
end for

Algorithm 3: FPTAS for Nondecreasing DP.

1: **Function SubSet**(φ, D, K)
2: $x \leftarrow D^{\max}$
3: $W \leftarrow \{D^{\min}, D^{\max}\}$
4: **while** $x > D^{\min}$ and $K\varphi(D^{\min}) < \varphi(x)$ **do**
5: $\quad x \leftarrow x' \mid K\varphi(x') < \varphi(x)$ and $K\varphi(\text{next}(x', D)) \geq \varphi(x)$
6: $\quad W \leftarrow W \cup \{x, \text{next}(x, D)\}$
7: **end while**
8: Return W

Algorithm 4: Constructing a subset of D for a function φ that approximates a nondecreasing function.

We give a few remarks about our FPTAS. The first remark is about Step 5 of Algorithm 3. Since $\bar{z}_t$ is not necessarily monotone, we cannot calculate for it a K-approximation set by calling $ApxSet(\bar{z}_t, \mathcal{S}_t, K)$ (it is not possible to perform Step 5 of Algorithm 1 in logarithmic time when φ is not nondecreasing). Instead, we call function *Subset* above. Step 6 of Algorithm 3 is easily performed by noticing that $\tilde{z}_t$ is a nondecreasing step function: We first set $\tilde{z}_t(\mathcal{S}_t^{\max}) \leftarrow \bar{z}_t(\mathcal{S}_t^{\max})$ and next scan W_t backwards, and set $\tilde{z}_t(x)$ for every pair of consecutive elements $x < y$ in W_t to be $\tilde{z}_t(x) \leftarrow \min\{\bar{z}_t(x), \tilde{z}_t(y)\}$.

Last remark is about how we efficiently perform Step 5 of Algorithm 4. We find x' that satisfies the condition in this step by performing binary search over the domain D. In the first iteration of the search, the scope is $D_1 := D \cap \{y \leq x \mid y \in D\}$, and the condition of the while loop implies that the invariant $K\varphi(D_1^{\min}) < \varphi(x)$ and $K\varphi(D_1^{\max}) \geq \varphi(x)$ holds (note that $D_1^{\max} = x$). We choose a middle element $m \in D_1$. If $K\varphi(m) < \varphi(x)$ we set $m' := \text{next}(m, D_1)$. If $K\varphi(m') \geq \varphi(x)$ the search is completed by assigning $x \leftarrow m$. Otherwise $(K\varphi(m), K\varphi(m') < \varphi(x))$, we reduce the new scope of the search to be $D_2 := D_1 \cap \{y \geq m' \mid y \in D\}$. On the other hand, if $K\varphi(m) \geq \varphi(x)$ we set $m' := \text{prev}(m, D)$. If $K\varphi(m') < \varphi(x)$ the search is completed by assigning $x \leftarrow m'$. Otherwise $(K\varphi(m), K\varphi(m') \geq \varphi(x))$, we reduce the new scope of the search to be $D_2 := D_1 \cap \{y \leq m' \mid y \in D\}$. In both cases where the search is not completed we get that the new domain D_2 is decreased by half and satisfies the invariant $K\varphi(D_2^{\min}) < \varphi(x)$ and $K\varphi(D_2^{\max}) \geq \varphi(x)$. We continue the search in the same way. Clearly, the search is well defined and is completed in $O(\log |D|)$ steps. Note that for each iteration of the while loop the ratio between the value of φ over the x before entering Step 5, and the value of φ over the x after leaving this step is at least K. This means that the while loop repeats $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$ times. We have just proved

Lemma 9.1. *Let $L \geq 1$ be arbitrary, $\psi : D \to \mathbb{R}^+$ be a nondecreasing function, and $\varphi : D \to \mathbb{R}^+$ be a (not necessarily monotone) L-approximation of ψ. Then for every $K > 1$, $W := Subset(\varphi, D, K)$ is a subset of D of cardinality $O(1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}})$ that is calculated in $O((1 + \log_K \frac{\varphi^{\max}}{\varphi^{\min}} \log |D|) t_\varphi)$ time and satisfies*

$$\varphi(next(x, W)) \leq K\varphi(x), \quad \forall D^{\max} > x \in W \text{ and } next(x, W) \neq next(x, D).$$

Proposition 9.2 (Nondecreasing Invariant)**.** *If Condition 3(ii) is satisfied, then for every $t = 1, \ldots, T$, function z_t in (5) is nondecreasing over $\mathcal{S}_t$.*

Proof. Due to Condition 3(ii) either z_t is nondecreasing for $t = 1, \ldots, T$ and there is nothing to prove, or $\forall I', I \in \mathcal{S}_t$ with $I' \leq I$, $\mathcal{A}_t(I) \subseteq \mathcal{A}_t(I')$. In the second case we use backward induction. z_{T+1} is nondecreasing since $z_{T+1} = g_{T+1}$ and g_{T+1} is such. Assuming that z_{t+1} in nondecreasing, we need to show that z_t is such as well. Since $f_t(\cdot, x_t, D_t)$ is nondecreasing, also does the composition $z_{t+1}(f_t(\cdot, x_t, D_t))$. Since $g_t(\cdot, x_t, D_t)$ is nondecreasing, also does the sum

$$g_t(\cdot, x_t, D_t) + z_{t+1}(f_t(\cdot, x_t, D_t)). \tag{13}$$

Last, since $\mathcal{A}_t(\cdot)$ is nonincreasing (by set containment) over $\mathcal{S}_t$, the minimization of (13) over $\mathcal{A}_t(\cdot)$ is nondecreasing over $\mathcal{S}_t$, and so does z_t. □

Finally, we are ready to prove our main result.

Theorem 9.3 (FPTAS for monotone DP)**.** *Given a dynamic program that satisfies Conditions 1,2 and either Condition 3(ii) or Condition 3(iii), and* $1 > \epsilon > 0$*, 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 give a proof for nondecreasing DP. The proof for nonincreasing DP is similar. It suffices to prove that for every $1 > \epsilon > 0$ and every initial state I_1, $\tilde{z}_1(I_1)$ given in Step 6 in the last iteration of Algorithm 3 is a $(1+\epsilon)$-approximation of the optimal cost $z^*(I_1)$, and that Algorithm 3 runs in time polynomial in both $\frac{1}{\epsilon}$ and the (binary) input size.

We first show that the algorithm is well defined. The Nondecreasing Invariant, Proposition 9.2, together with fact that the various $\tilde{z}_t$'s are nondecreasing by construction, assure us that all the functions for which the algorithm builds K-approximation sets are nonnegative and nondecreasing. In this way all the calls to functions ApxSet and Subset are well defined.

We next show the correctness of the algorithm. Let $K := 1 + \frac{\epsilon}{2(T+1)}$. We first show that it suffices to prove that for every $t = 1, \ldots, T+1$, $\tilde{z}_t$ is a K^{T+2-t}-approximation of z_t. By the DP optimality equation (5) in Theorem 3.1, the optimal cost $z^*(I_1)$ is equal to $z_1(I_1)$. By the definition of K-approximation functions (Definition 4.2), $z^*(I_1) \leq \tilde{z}_1(I_1) \leq (1+\frac{\epsilon}{2(T+1)})^{T+1} z^*(I_1)$. Thus, due to the inequality $(1+\frac{x}{n})^n \leq 1+2x$, which holds for every $0 \leq x \leq 1$, we get that $z^*(I_1) \leq \tilde{z}_1(I_1) \leq (1+\epsilon) z^*(I_1)$.

We next prove by induction that for every $t = 1, \ldots, T+1$, $\tilde{z}_t$ is a nondecreasing K^{T+2-t}-approximation of z_t and $\{\mathcal{S}_t^{\max}\} \cup \{x, \text{next}(x, \mathcal{S}_t) \mid \mathcal{S}_t^{\max} > x \in W_t\}$ is a 1-approximation set of it. The base case of $t = T+1$ follows from Theorem 4.4. The induction hypothesis is that $\tilde{z}_{t+1}$ is a nondecreasing K^{T+1-t}-approximation of z_{t+1} and $\{\mathcal{S}_{t+1}^{\max}\} \cup \{x, \text{next}(x, \mathcal{S}_{t+1}) \mid {\mathcal{S}_{t+1}}^{\max} > x \in W_{t+1}\}$ is a 1-approximation set of it. By applying Proposition 7.2 with $K_1' = K_2' = 1, L_1' = K$ and $L_2' = K^{T+1-t}$ we get that $\bar{z}_t$ is a (not necessarily monotone) K^{T+1-t}-approximation of z_t. By the definition of $\tilde{z}_t$ we get that it is a nondecreasing function that satisfies

$$\tilde{z}_t(\text{next}(x, W_t)) \leq K \tilde{z}_t(x), \quad \forall \mathcal{S}_t^{\max} > x \in W_t \text{ and } \text{next}(x, W_t) \neq \text{next}(x, \mathcal{S}_t). \tag{14}$$

To see this we distinguish between 3 cases. If $\tilde{z}_t$ equals $\bar{z}_t$ on both x and $\text{next}(x, W_t)$ then (14) holds by applying Lemma 9.1 with $\psi := z_t, L := K^{T+1-t}$ and $\varphi := \bar{z}_t$. If $\tilde{z}_t(x) \neq \bar{z}_t(x)$, then $\tilde{z}_t(x) = \tilde{z}_t(\text{next}(x, W_t))$, and (14) is again satisfied. Otherwise, $\tilde{z}_t(x) = \bar{z}_t(x)$ and $\tilde{z}_t(\text{next}(x, W_t)) \neq \bar{z}_t(\text{next}(x, W_t))$. Then we have $\tilde{z}_t(\text{next}(x, W_t)) < \bar{z}_t(\text{next}(x, W_t)) \leq K \bar{z}_t(x) = K \tilde{z}_t(x)$, where the second inequality is due to Lemma 9.1.

Note that by Proposition 4.3 W_t is a K-approximation set of $\tilde{z}_t$. Next we show that $\tilde{z}_t$ is a K^{T+1-t}-approximation of z_t restricted to W_t. Indeed, from the definition of $\tilde{z}_t$ we have

$$\tilde{z}_t(x) \leq \bar{z}_t(x) \leq K^{T+1-t} z_t(x), \quad \forall x \in W_t. \tag{15}$$

On the other hand, for every $D^{\max} > x \in W_t$ and $\text{next}(x, W_t) \neq \text{next}(x, \mathcal{S}_t)$ we have

$$\tilde{z}_t(x) = \min\{\bar{z}_t(x), \tilde{z}_t(\text{next}(x, W_t))\} \geq \min\{z_t(x), z_t(\text{next}(x, W_t))\} = z_t(x), \tag{16}$$

where the inequality is due to the fact that $\bar{z}_t$ is an approximation function of z_t, and $\tilde{z}_t(\text{next}(x, W_t)) = \bar{z}_t(y)$ for some $y \geq \text{next}(x, W_t)$ so the monotonicity of z_t implies $\tilde{z}_t(\text{next}(x, W_t))\} = z_t(y) \geq z_t(\text{next}(x, W_t))$. We apply now approximation of approximation sets (Proposition 6.2) with $\varphi_1 = \tilde{z}_t, K_1 = K, \varphi_2 = z_t$ and $K_2 = K^{T+1-t}$ and get that $\hat{\tilde{z}}_t = \tilde{z}_t$ is a K^{T+2-t}-approximation of z_t as needed.

It remains to analyze the running time of the algorithm. Recall that $U_D^{\mathcal{S}} = \max_{t=1,\ldots,T+1} |\mathcal{S}_t|$ is the size of the maximal state space, $U_D^{\mathcal{A}} = \max_{t=1,\ldots,T} |\mathcal{A}_t|$ is the size of the maximal action space, and $U_R = \max_t \max_{I \in I_t,\, x \in \mathcal{A}_t(I),\, d \in \mathcal{D}_t} g_t(I, x, d)$ is the maximal single-period cost value. Due to Conditions 1 and 2 $\log U^{\mathcal{S}}, \log U^{\mathcal{A}}$ and $\log U_R$ are all polynomially bounded by the input size. For ease of exposition, let us assume that all these values are at least 2. By Theorem 4.4, Steps 2-3 of Algorithm 3 are executed in $O(t_g \log_K U_R \log U_D^{\mathcal{S}})$ time, where t_g is the time needed to evaluate g. Clearly, the "for loop" is executed T

times and so is function Subset. By Lemma 9.1 each execution of function Subset takes $O(\log_K U_R \log U_D^{\mathcal{S}} t_{\bar{z}_t})$ time, where $t_{\bar{z}_t}$ is the time needed to evaluate $\bar{z}_t$ in (11). We now apply Proposition 7.2 with $L_1' = K_1' = K_2' = K$ and $L_2' = K^{T+1-t}$. W_t^c is simply W_{t+1} from the previous iteration of the for loop. By Theorem 4.4 we build W_t^a and W_t^b of size $O(\log_K U_R)$ in $O(\log_K U_R \log U_D^{\mathcal{A}} t_g)$ time. Therefore $t_{\bar{z}_t} = O(n^* \log_K U_R(\log U_D^{\mathcal{A}} t_g + t_{\tilde{z}_{t+1}}))$, where $t_{\tilde{z}_{t+1}}$ is the time needed to evaluate $\tilde{z}_{t+1}$ from the previous iteration of the for loop. Since $\tilde{z}_{t+1}$ is stored succinctly in a sorted array of size $|W_{t+1}|$, we get that $t_{\tilde{z}_{t+1}} = O(\log\log_K U_R)$. We conclude that the running time of the algorithm is $O((t_g + t_f + \log(\log_K U_R))n^* T \log_K^2 U_R \log U_D^{\mathcal{S}} \log U_D^{\mathcal{A}})$. Since $\epsilon < 1$, we get that $K < 2$, so $O(\log_K U_R) = O(\frac{1}{K-1} \log U_R)$. By replacing K with $1 + \frac{\epsilon}{2T}$, we conclude that the running time of the algorithm is

$$O((t_g + t_f + \log(\frac{T}{\epsilon} \log U_R))\frac{n^* T^3}{\epsilon^2} \log^2 U_R \log U_D^{\mathcal{S}} \log U_D^{\mathcal{A}}). \tag{17}$$

Since U_R depends linearly on the multiplying factor, which is at most M_1, and since $\log U_D^{\mathcal{S}}, \log U_D^{\mathcal{A}}$ and $\log M_1$ are all polynomially bounded by the input size, the algorithm runs in polynomial time in both the input size and $\frac{1}{\epsilon}$. □

Remarks: The dependency of the running time of the algorithm on T is at most $T^3 \log T$ and on ϵ is at most $\frac{1}{\epsilon^2} \log \frac{1}{\epsilon}$. Note that if the single-period cost function g_t can be expressed as the sum of univariate functions as explained in the end of Section 7, then by computing a single K-set of it for each time period, the running time of the algorithm is improved to $O((t_f + \log(\frac{T}{\epsilon} \log U_R))\frac{n^* T^3}{\epsilon^2} \log^2 U_R \log U_D^{\mathcal{S}} + t_g \frac{n^* T^2}{\epsilon} \log U_D^{\mathcal{A}})$. Note also that the numbers computed throughout the algorithm are polynomially bounded by the input size: they are all bounded by $M_1 T U_R$.

10 Extensions

10.1 Maximization problems

Not surprisingly, our framework applies for maximization problems as well, as summarized in Theorem 3.3. We give here details how to apply the framework for maximization problems. So far we have considered one-sided approximations, where for $K \geq 1$, the approximated function $\tilde{\varphi}$ is "sandwiched" between the original function φ and $K\varphi$ (see Definition 4.1). This was done so the approximated value found by our FPTAS is "sandwiched" between the optimal value and K times of it. For maximization problems we need a different one-sided approximation, one with $K \leq 1$ (e.g., $K = 1 - \epsilon$). We apply the standard transformation $\max \varphi = -\min -\varphi$. It is easy to see that when expanding all the definitions in Sections 4-9 to non-positive functions and to K's and L's with positive values not exceeding 1, then all our results carry over.

10.2 Random vectors

Until now we have assumed that the D_t are independent random one-dimensional variables. Dealing with multi-dimensional random variable, i.e., *random vectors*, is straightforward. Consider, for example, a more general version of the stochastic adaptive ordered knapsack problem studied in Section A.1, where not only the volume V_t is a random variable, but also the profit Π_t. In this case the input includes the mutual distribution of $(V_t, \Pi_t) = D_t$ (for every t we allow V_t and Π_t to be non-independent). The domain of the single-period cost function g_t and the transition function f_t is then 4-dimensional, where $g_t(I_t, x_t, V_t, \Pi_t) = x_t \Pi_t \delta_{V_t \leq I_t}$ and $f_t(I_t, x_t, V_t, \Pi_t) = (I_t - x_t V_t)^+$. Another example concerns supply processes. Suppose that in addition there is a possibility that the order to include an item into the knapsack may not be fulfilled, depending on the instantiation of an additional *binary* random variable O_t. Then the domain of the single-period cost function g_t and the transition function f_t is augmented to be 5-dimensional, where $g_t(I_t, x_t, V_t, \Pi_t, O_t) = O_t x_t \Pi_t \delta_{V_t \leq I_t}$ and $f_t(I_t, x_t, V_t, \Pi_t) = (I_t - O_t x_t V_t)^+$. An example of a binary random process in inventory control theory is given in [PWG95]. *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 [YL95] and the references therein). Consider, for example, a random yield version of the batch disposal problem studied in Section A.2.2, where 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 random yield random variable O_t (i.e., the random vector is $D_t = (G_t, O_t)$). In this case $g_t(I_t, x_t, G_t, O_t) = O_t c_t x_t + K_t \delta_{O_t x_t > 0} + h_t(I_t + G_t - O_t x_t)$, and $f_t(I_t, x_t, G_t, O_t) = (I_t + G_t - O_t x_t)^+$.

10.3 Non independence

Dealing with non-independent random vectors requires more attention. Let us consider the well-studied model of "World-Driven Demand" ("Markov-Modulated Demand") (see, for example, [CSL04, IK62, KF59, SZ93] and pages 415-420 in [Zip00]). 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 $W_t = w$. This means that the D_t are no longer independent, they are influenced by the W_t, and the Markovian dependence among the W_t induces dependence in the D_t. We even allow D_t and the next world state W_{t+1} to be driven by common events, so D_t and W_{t+1} may be dependent. We assume, however, that these are the only sources of dependence, i.e., conditional on W_t, the pair (D_t, W_{t+1}) is independent of all past events. In a generalization of the cash management problem [HW01], the world may represent the economy. In a generalization of the batch disposal problem where the goods are ashes of fireplaces, the world may represent the weather.

Let the world state consist of the n states $[1, \dots, n]$, and be represented by a transition probability matrix $(W_{i,j})$. Note that the classical model of i.i.d. D_t is contained in our model by choosing the world to consist of a single state. Also note that the basic model presented in Section 3 is contained in our model: we simply set the number of states in the world to be $n = T$, and choose the matrix $(W_{i,j})$ to be a $T \times T$ stochastic matrix with $w_{i,i+1} = 1$ for $i = 1, \dots, T-1$, $w_{T,T} = 1$, and 0 elsewhere.

In the World-Driven Demand model the domain of z_t is $[1, \dots, n] \times \mathcal{S}_t$. Thus instead of (4) we have

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

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 (5) we have

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

$$z_t(w_t, I_t) = \min_{x_t \in \mathcal{A}_t(I_t)} E_{D_t \mid W_t = w_t}\{g_t(w_t, I_t, x_t, D_t) + E_{W_{t+1} \mid W_t = w_t} z_{t+1}(W_{t+1}, f_t(w_t, I_t, x_t, D_t))\}, \quad t = 1, \dots, T,$$

and instead of (6) we have

$$E_{D \mid W_t = w_t}\{g_t(w_t, I_t, x_t, D_t) + E_{W_{t+1} \mid W_t = w_t} z_{t+1}(W_{t+1}, f_t(w_t, I_t, x_t, D_t))\} =$$
$$= \sum_{j=1}^{n_{w_t}} p_{w_t,j} \left(g_t(w_t, I_t, x_t, d_{w_t,j}) + \sum_{i=1}^{n} w_{w_t,i} z_{t+1}(i, f_t(w_t, I_t, x_t, d_{w_t,j})) \right).$$

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.

It is easy to generalize our framework to give FPTASs for any constant number of Markov-modulated processes, where 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 [GH04], then the domain of $z_t(w_t^D, w_t^S, I_t)$ becomes 3-dimensional, and for every pair of states $(w_t^D, w_t^S) \in [1, \dots, n^D] \times [1, \dots, n^S]$ we compute for it K-approximating sets and functions.

It is also possible to handle another case of non-independence. 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 in period t, d_t, is dependent only on the observed state in period $t-1$, d_{t-1}.

We last consider a non-Markov-modulated process, where the world state at time t, w_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, 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 admits an FPTAS as well.

We conclude this section by considering Convex/Monotone DP with general non-independent random variables.

Theorem 10.1. *The stochastic ordered adaptive knapsack problem in which the weights of each period are non independent is APX-hard.*

Since in Section A.1 we formulated the stochastic ordered adaptive knapsack problem as a maximization nondecreasing DP we get the following corollary:

Corollary 10.2. *The Convex/Monotone DP framework presented in this paper cannot be extended to deal with general non-independent random variables, unless $P = NP$.*

The proof of Theorem 10.1 makes a transformation from the Max k-cover problem, which is known not to be approximated within a factor of $1 - \frac{1}{e}$ [Fei98]).

Problem: Max k-cover (MkC)
Instance: *m subsets $S_1, S_2, \ldots, S_\ell \subset S = \{1, 2, \ldots, m\}$ and an integer $k \le m$.*
Question: *What is the maximum number of elements of S that can be covered by exactly k subsets?*

Proof. We transform MkC, for the case that $|S_j| = 3$ for each j, into the stochastic ordered adaptive knapsack problem (SKP) (19). The proof below will carry through even if we permitted the sizes of the subsets to vary. Let $S_j = \{s_{j1}, s_{j2}, s_{j3}\}$ for each $j = 1, \ldots, \ell$.

1. $n = \ell + 2m$ items, one for each subset and two for each element of S.
2. $B = 8\ell k + 7k$.
3. For periods $j = 1$ to ℓ of the stochastic knapsack problem, we have the following:
   (a) $\pi_j = 0$.
   (b) $8\ell \le v_j \le 8\ell + 7$. Each occurs with probability $\frac{1}{8}$. We will discuss dependencies soon.
4. For the remaining $2m$ periods, induced by indices $\ell + 2j - 1$ and $\ell + 2j$ for $j = 1, \ldots, m$, we have the following:
   (a) $\pi_{\ell+2j-1} = \pi_{\ell+2j} = 1$.
   (b) Either $v_{\ell+2j-1} = 0$ and $v_{\ell+2j} = M$ or else $v_{\ell+2j-1} = M$ and $v_{\ell+2j} = 0$, where $M = 9\ell k$. Each possibility occurs with probability $\frac{1}{2}$. Thus they are dependent. We refer to one of these elements as *light* and one as *heavy*.
5. Define a binary function $f : S \rightarrow \{0, 1\}$ as follows. $f(j) = 0$ if $v_{\ell+2j-1} = 0$ and $v_{\ell+2j} = M$, and let $f(j) = 1$ otherwise.

6. Also for $j = 1, \ldots, \ell$ let $v_j = f(s_{j1} + 2f(s_{j2}) + 4f(s_{j3}) + 8\ell$. This introduces lots of dependencies between variables. We refer to elements 1 to ℓ as *revealing*. Each of these elements reveals the location of 3 light elements.

It suffices to prove that if one can approximate SKP to within a factor $\frac{1}{2}$ then one can approximate MkC within a constant greater than $1 - \frac{1}{e}$.

Without loss of generality, an optimum solution to this instance of SKP consists of selecting k of the first ℓ items, and then selecting as many as possible light elements of the remaining 2ℓ items.

Suppose first that there is a feasible solution $\{S_j \mid j \in J\}$ for MkC that covers K items. Then the optimal solution to the stochastic adaptive knapsack problem has value between K and $K+2$. To see this, let $x_j = 1$ for $j \in J$. The value v_j is a code that reveals the location of three light elements. The values $\{v_j \mid j \in J\}$ reveal the location of K distinct light elements. Other items can be guessed, with an expected additional value of at most $\sum_{j=1}^{m} \frac{j}{2^j} < 2$.

Suppose next that the optimal solution to SKP is at least K. The most that can be obtained by guessing unrevealed light elements is a total value of 2. Therefore, the initial k items selected must have revealed the location of at least $K-2$ items, and so the solution to the max cover problem is at least $K-2$.

Hence, the relative approximation ratio is at least $\frac{\frac{1}{2}K-2}{K} = \frac{1}{2} - \frac{2}{K}$. We conclude the proof by noting that approximating MkC within a constant $1 - \frac{1}{e} + \epsilon$ is NP-hard even for $K > \frac{4e}{2-e}$. □

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 problems in our framework. Recall that a real-valued function $f : \mathbb{R}^d \to \mathbb{R}^+$ is *V-shaped* on its first variable x if it is linear with non-positive slope for $x < 0$, and linear with nonnegative slope for $x > 0$. Also recall that a *limit policy* (r, s) $(-\infty \leq r \leq s \leq \infty)$ is a policy under which whenever the state I falls below r we augment it to r by ordering $r - I$ units, and whenever the state I exceeds s we decrease it to s by discarding of $I - s$ units. (When the state I is between r and s we do nothing.)

Theorem 10.3. *Suppose a given convex DP satisfies the following. For every time period t and fixed D_t the transition cost function $f_t(\cdot, \cdot, D_t)$ is linear and the absolute values of the coefficients for both its variables are the same. Moreover, the single-period cost function 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 on x_t, and u_t is convex. Then this convex DP admits an optimal limit policy (r_t, s_t) and Algorithm 2 computes such an approximated limit policy $(\hat{r}_t, \hat{s}_t)$.*

Proof. Let us fix t. For the ease of terminology we refer to the state I as inventory level, and the action x as production/disposal quantity (the former for positive values of x, the later for negative values). Due to Condition 3(i), the coefficients are either -1,0 or 1. From the optimality equation (5) and linearity of expectation, the optimal policy chooses x^* that minimizes

$$E_{D_t} g_t(I_t, \cdot, D_t) + E_{D_t} z_{t+1}(f_t(I_t, \cdot, D_t)). \tag{18}$$

The Convexity Invariant (Theorem 8.1) implies that z_{t+1} is convex, so since $f_t(I_t, \cdot, D_t)$ is linear in x, we get that $z_{t+1}(f_t(I_t, \cdot, D_t))$ is convex as well. Moreover, since a convex combination of convex functions is such as well, we have that (18) is convex. Let

$$a_t(I_t, x_t) := E_{D_t} v_t(x_t, D_t) \qquad \text{and} \qquad b_t(I_t, x_t) := E_{D_t} u_t(f_t(I_t, x_t, D_t), D_t) + E_{D_t} z_{t+1}(f_t(I_t, x_t, D_t)).$$

Then we have

$$z_t(I_t) = \min_{x \in \mathcal{A}_t} a_t(I_t, x_t) + b_t(I_t, x_t),$$

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

If both coefficients are 0, then $f_t(\cdot, \cdot, D_t)$ is indifferent to the action x_t, and therefore also is b_t, so the best strategy is to minimize the V-shaped function a_t, i.e., do nothing. Hence in this case $r_t = -\infty$ and $s_t = \infty$.

We now analyze the case where both coefficients are 1. Let $R_t := \{I \in \mathbb{Z} \mid b_t(I-1, 0) - b_t(I, 0) > a_t^+\}$ be all the inventory levels for which production of 1 unit is beneficial. Let $S_t := \{I \in \mathbb{Z} \mid b_t(I+1, 0) - b_t(I, 0) > |a_t^-|\}$ be all the inventory levels for which disposal of 1 unit is beneficial. Note that here we make use of the case assertion that the coefficients for both first variables of $f_t(\cdot, \cdot, D_t)$ are equal. We next show that the levels of the limit policy are

$$r_t := \max R_t, \qquad s_t := \min S_t.$$

(In case that a set is empty, we define its maximum (minimum) to be $-\infty$ (∞), respectively.) Note that the convexity of function a_t implies that the set R_t is connected over $\mathbb{Z}$, so if the current inventory is $I_t \in R_t$ then producing $r_t - I_t$ units is beneficial. Similarly, if the current inventory is $I_t \in S_t$ then due to the convexity of function a_t, disposing of $I_t - s_t$ units is beneficial. This completes the proof for the case where both first coefficients of $f_t(I_t, \cdot, D_t)$ are 1.

If the coefficient of I_t is 1 and of x_t is -1, then negative x's mean production and positive x's mean disposal (graphically, function a_t is rotated along the $x = 0$ plane). Hence we define for this case $R_t' := \{I \in \mathbb{Z} \mid b_t(I-1, 0) - b_t(I, 0) > a_t^-\}$ to be all the inventory levels for which production of 1 unit is beneficial, and $S_t' := \{I \in \mathbb{Z} \mid b_t(I+1, 0) - b_t(I, 0) > |a_t^+|\}$ to be all the inventory levels for which disposal of 1 unit is beneficial. The rest of the analysis for this case is identical to the former case.

We note that the case where both coefficients are -1 is completely analogue to the first case, and the case where the coefficient of I_t is -1 and of x_t is 1 is completely analogue to the second case.

It remains to analyze the structure of the policy exercised by Algorithm 2. Note that 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 production quantities output by the algorithm follow the optimal limit policy for these (approximated) piecewise-linear convex functions, and these quantities serve as *approximated* quantities for the *exact* convex functions. Note that it is easy to change the algorithm so it will output the approximated limit policy levels for each time period. □

The cash management problem studied in Section A.4 satisfies the conditions of the theorem above.

10.5 Non-exact evaluation of cost functions

In Condition 2, we require that g_t be evaluated in polynomial time. We can weaken this condition by requiring that there exists an FPTAS for evaluating g_t.

Assumption 10.4. *For every $\delta \geq 0$ and time period t there exist a function $\tilde{g}_t^\delta$ such that*

$$\frac{|\tilde{g}_t^\delta(x) - g_t(x)|}{g_t(x)} \leq \delta,$$

for every x, and this function can be evaluated in polynomial time in the input size and $1/\delta$.

This assumption is equivalent to the statement that for every $K > 1$, g_t has a two-sided K-approximation.

Definition 10.5. *Let $K > 1$ and let $\varphi : D \to \mathbb{R}$ be a function. We say that $\tilde{\varphi} : D \to \mathbb{R}$ is a* two-sided K-approximation *of φ if for all $x \in D$ we have $\varphi(x)/K \leq \tilde{\varphi}(x) \leq K\varphi(x)$.*

Proposition 10.6. *Let $K > 1$ and let $\varphi : D \to \mathbb{R}$ be a function. 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.*

Suppose first that either Condition 3(ii) or Condition 3(iii) is satisfied, and that there exists an FPTAS for evaluating each of g_t^a, g_t^b. In order that our framework applies, we need build for each of these functions a monotone K-approximation function. We do this as follows (for brevity we deal only with g_t^a and assume it is nondecreasing. The case of g_t^b is analoguous). For simplicity we omit the superscript a and subscript t from the notation g_t^a. Let $I \in \mathcal{S}_t$ and $d \in \mathcal{D}_t$ be fixed. Let $\bar{g}(I, \cdot, d) \leftarrow \sqrt[3]{K} g^{\sqrt[3]{K}}(I, \cdot, d)$ be a one-sided (not necessarily monotone) $\sqrt[3]{K^2}$-approximation of $g(I, \cdot, d)$. Let $W(I, d) \leftarrow$**Subset**$(\bar{g}(I, \cdot, d), \mathcal{A}, \sqrt[3]{K})$. Last, let $\tilde{g}(I, \cdot, d)$ be the maximal nondecreasing function that is bounded above by $\bar{g}(I, \cdot, d)$ on $W(I, d)$. By the analysis in we get that $\tilde{g}(I, \cdot, d)$ is a nondecreasing K-approximation of $g(I, \cdot, d)$. We apply Proposition 7.2 in the proof of Theorem 9.3 with $K_1' = K$ instead of $K_1' = 1$.

For the convex case we need to have a stronger assumption that there exists a *convex* FPTAS for g_t. In this case, in the proof of Theorem 8.4 we simply use $\sqrt{K} g_t^{\sqrt{K}}(I, \cdot, d)$ as a one-sided convex K-approximation of g_t, and apply Proposition 7.1 with $K_1' = K$ instead of $K_1' = 1$.

10.6 Multivariate functions

A natural question is as follows. Can we generalize Theorem 4.4 to multivariate functions? Unfortunately, due to the counting argument given below, this is not possible for the natural extension of this theorem. To see this, we first observe that directly from the definition of K-approximation functions we have

Proposition 10.7. *For every $K > 1, d \in \mathbb{N}$ and binary function $\varphi : [1, U]^d \to [0, 1]$, φ coincides with any of its K-approximation functions.*

So approximating φ instead of exactly calculating φ does not reduce the complexity of the problem.

We next count the number of "special" binary functions over $[1, U] \times [1, U]$. Clearly, there are 2^{U^2} distinct binary functions over $[1, U] \times [1, U]$. Therefore, we need $\Omega(\log 2^{U^2}) = \Omega(U^2)$ space to represent such a function, i.e., more than polylogarithmic in U.

Proposition 10.8. *There are $\Omega(2^{\sqrt{U}})$ monotone convex binary functions over $[1, U] \times [1, U]$.*

Proof. Due to the monotonicity of the function, the number of (consecutive) 0's in the (beginning of the) rows decreases in a convex way as we proceed with the rows. For every row $i \geq 2$ we need to determine $x_i :=$ the number of more 1's that appear in this row relatively to row $i - 1$. The convexity implies that $x_2 \leq x_3 \leq \cdots \leq x_U$. We count the special case where row U consists of 1's only. In this case $\sum_{i=2}^{U} x_i = U$. Each such combination of x_i's is a partition of U into at most $U - 1$ positive integers. Let $P(U)$ be the number of partitions of U (including the partition of U into U 1's). The number of combinations in our special case is hence $P(U) - 1$. The proposition follows from the fact that $P(U) > 2^{\sqrt{U}}$ [HR18]. □

So we need $\Omega(\sqrt{U})$ space to represent a convex-nondecreasing binary function, again too much. Since in principle it is impossible to represent binary functions over domain D in $O(\log^k |D|)$ space, for any fixed k, we get the following theorem

Theorem 10.9 (Non existence of succinct approximations for multivariate functions)**.** *For any $K > 1$, a bivariate convex and monotone function does not necessarily admit a succinct K-approximation, regardless of the scheme used to represent the function.*

11 Concluding remarks and future research

In this paper we have presented a framework for obtaining FPTASs for stochastic monotone or convex dynamic programs. 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. We still use the standard optimality equation or recursion; however, we consider only polynomially many states. Our algorithm relies on either convexity or monotonicity of the value function.

Previous works tried to determine sufficient and necessary conditions for a dynamic program to admit an FPTAS. While most of these works are not constructive, the one of Woeginger [Woe00] gives sufficient conditions for a deterministic dynamic program to admit an FPTAS, and states a clear FPTAS for such a DP. He demonstrates a number of examples that fit into his framework, where all of those examples have already (other) known FPTASs. We note that none of the problems discussed in this paper appears to fit into his framework, either because the problem is stochastic, or it does not satisfy his Condition C4(*ii*).

It is interesting to relax any of Conditions 1-3. Regarding Condition 1, we would have liked to extend our framework to deal with multi-variate DP, i.e., to allow fixed-dimensional state and action spaces. Theorem 10.9 tells us that our approach does not extend to multi-variate functions. The question is therefore, does there exist FPTASs for multi-variate monotone DP? Unfortunately, the answer is is unlikely to be positive, since it is known that the existence of an FPTAS for the 2-dimensional 0/1 knapsack problem (which can be formulated as a 2-dimensional nondecreasing DP) would imply $P = NP$ (see p. 252 in [KPP04] and the references therein).

Our method appears quite general. Is it possible to extend it to other recursive structures rather than finite horizon DPs (which can be reduced to problems on path-like graphs)? The answer is positive. In [HLS08] a subset of the authors apply K-approximation sets and functions to design an FPTAS for time-cost tradeoff project scheduling problems over series-parallel graphs. We can also address optimization problems over trees. It is interesting to formulate a general recursive structure that admits FPTASs using our method.

In Section 10.5 we deal explain how to get FPTAS in the case where we have only indirect access to the cost functions via FPTASs. We solve the monotone DP case by transforming a (not necessarily monotone) FPTAS for a monotone function φ into a monotone K-approximation of φ. In the convex DP case we assume that our FPTAS is convex. It is interesting to relax this assumption. A possible way to achieve this is by showing how to convert a (not necessarily convex) FPTAS for a convex function φ into a convex K-approximation of φ.

We conclude with two complexity remarks. First, Alekhnovich *et al.* [ABB$^+$05] present a model for backtracking and dynamic programming. They prove several upper and lower bounds on the capabilities of algorithms in their model, and show that it captures the simple dynamic programming framework of Woeginger [Woe00]. In their paper they question whether their model captures other dynamic programming algorithms. It would be interesting to check the capabilities of our framework in this context.

Second, many #P-complete problems exhibit fully polynomial randomized approximation schemes (FPRASs), for example, counting Hamiltonian cycles in dense graphs [DFJ98], counting knapsack solutions [Dye03], counting Eulerian orientations of a directed graph [MW95], counting perfect matchings in a bipartite graph [JS89], and computing the permanent [JSV04]. To the best of our knowledge, the only deterministic FPTASs for #P-hard problems known up-to-date and published in the literature are found in the very recent works of [Wei06, BG07, BGK$^+$07, GD07], and are developed by applying methods from statistical physics. Our FPTAS, which uses different methods, is another rare example in the literature of a (deterministic) FPTAS for #P-hard problems. Dyer *et al.* [DGGJ03] investigate classes of counting problems that are interreducible under approximation-preserving reductions. One of these classes is the class of counting problems that admit (randomized) FPRASs. It is therefore interesting in this context to investigate the class of counting problems that admit FPTASs.

Acknowledgments. The first author would like to thank Oded Goldreich, Sudipto Guha and Asaf Levin for inspiring discussions, Leslie Ann Goldberg for pointing out selected references, and Nimrod Megiddo for bringing to his attention the book of Murota and the references therein [Mur03].

References

[ABB+05] M. Alekhnovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi. Towards a model for backtracking and dynamic programming. In *Proceedings of the 20th IEEE Conference on Computational Complexity*, 2005.

[AC03] J. Adda and R. Cooper. *Dynamic Economics: Quantitive Methods and Applications*. MIT Press, 2003.

[ACG+99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation*. Springer, 1999.

[ACW04] H. Almeida, M. Campello, and M. Weisbach. The cash flow sensitivity of cash. *Journal of Finance*, 59:1777–1804, 2004.

[Bel57] R. Bellman. *Applied Dynamic Programming*. Princeton University MIT Press, Princeton, NJ, 1957.

[Ber95] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.

[BG07] A. Bandyopadhyay and D. Gamarnik. Counting without sampling. new algorithms for enumeration problems using statistical physics. *To Appear in Random Structures and Algorithms*, 2007.

[BGK+07] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali. Simple deterministic approximation algorithms for counting matchings. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007.

[BS02] K.M. Bretthauer and B. Shetty. The nonlinear knapsack problem – algorithms and applications. *European Journal of Operational Research*, 138:459–472, 2002.

[CCLL98] 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 Research Logistics*, 45:67–82, 1998.

[CKP06] S. Chubanov, M.Y. Kovalyov, and E. Pesch. An FPTAS for a single-item capacitated economic lot-sizing problem with monotone cost structure. *Mathematical Programming*, 106:453–466, 2006.

[CSL04] X. Chen and D. Simchi-Levi. Coordinating inventory control and pricing strategies with random demand and fixed ordering costs: the finite horizon case. *Operations Research*, 52:887–896, 2004.

[CSW93] 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 Research Logistics*, 40:161–173, 1993.

[DFJ98] M. Dyer, A. Frieze, and M. Jerrum. Approximately counting Hamilton paths and cycles in dense graphs. *SIAM Journal on Computing*, 27:1262–1272, 1998.

[DGGJ03] M. Dyer, L.A. Goldberg, C. Greenhill, and M. Jerrum. The relative complexity of approximate counting problems. *Algorithmica*, 38:471–500, 2003.

[DGV04] B.C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: the benefit of adaptivity. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.

[DL77] S.E. Dreyfus and A.M. Law. *The Art and Theory of Dynamic Programming*. Academic Press, New York, 1977.

[Dye03] M. Dyer. Approximate counting by dynamic programming. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing*, pages 693–699, 2003.

[EF69] G.D. Eppen and E.F. Fama. Cash ballance and symple dynamic portfolio problems with proportional costs. *International Economic Review*, 10(2):119–133, 1969.

[Faa81] B.H. Faaland. The multiperiod knapsack problem. *Operations Research*, 29:612–616, 1981.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

[FLR80] M. Florian, J.K. Lenstra, and A.H.G. Rinnooy Kan. Deterministic production planning: algorithms and complexity. *Management Science*, 26:669–679, 1980.

[FT90] P. Favati and F. Tardella. Convexity in nonlinear integer programming. *Ricerca Operativa*, 53:3–44, 1990.

[FW06] M. Faulkender and R. Wang. Corporate financial policy and the value of cash. *Journal of Finance*, 61:1959–1990, 2006.

[GD07] D. Gamarnik and D.Katz. Correlation decay and deterministic fptas for counting list-colorings of a graph. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[GH04] G. Gallego and H. Hu. Optimal policies for production/inventory systems with finite capacity and Markov-modulated demand and supply processes. *Annals of Operations Research*, 126:21–41, 2004.

[GI99] A. Goel and P. Indyk. Stochastic load balancing and related problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 579–586, 1999.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W. H. Freeman, 1979.

[Hen90] M. Henig. Risk criteria in a stochastic knapsack problem. *Operations Research*, 38(5):820–825, 1990.

[HKL+08] 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 Nineteenth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 700–709, 2008.

[HKM+06] 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. Technical report, Massachusetts Institute of Technology, 2006. Submitted for journal publication.

[HLS08] N. Halman, C.-L. Li, and D. Simchi-Levi. Fully polynomial time approximation schemes for time-cost tradeoff problems in series-parallel project networks. *To appear in APPROX*, 2008.

[Hoc95] D.S. Hochbaum. A nonlinear knapsack problem. *Operations Research Letters*, 17:103–110, 1995.

[Hoc97] D.S. Hochbaum. Various notions of approximations: Good, better, best, and more. In D.S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 346–398. PWS Publishing company, 1997.

[HR18] G.H. Hardy and S. Ramanujan. Asymptotic formulae in combinatory analysis. *Proc. London Math. Soc.*, 17:75–115, 1918.

[HS76] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, 23:317–327, 1976.

[HW01] K. Hinderer and K.-H. Waldmann. Cash management in a randomly varying environment. *European Journal of Operational Research*, 130:468–485, 2001.

[IK62] D. Iglehart and S. Karlin. Optimal policy for dynamic inventory with nonstationary stochastic demands. In K.J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in Applied Probability and Management Science*, chapter 8. Stanford University Press, Stanford, CA, 1962.

[IK70] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches.* The MIT Press, Cambridge, MA, 1970.

[IK75] O.H. Ibarra and C.E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22:463–468, 1975.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.

[JSV04] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004.

[Kar72] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[KF59] S. Karlin and A. Fabens. The (s,S) inventory model under Markovian demand process. In K.J. Arrow, S. Karlin, and P. Suppes, editors, *Mathematical Methods in the Social Sciences*. Stanford University Press, Stanford, CA, 1959.

[KMS98] C.-S. Kim, D.C. Mauer, and A.E. Sherman. The determinants of corporate liquidity: Theory and evidence. *Journal of Finacial and Quantitative Analysis*, 33:335–359, 1998.

[KN09] S. Kameshwaran and Y. Narahari. Nonconvex piecewise linear knapsack problems. *European Journal of Operational Research*, 192:56–68, 2009.

[KP01] A.J. Kleywegt and J.D. Papastavro. The dynamic and stochastic knapsack problem with random sized items. *Operations Research*, 49(1):26–41, 2001.

[KPP04] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springler-Verlag, 2004.

[KRT97] J. Kleinberg, Y. Rabani, and E. Tardos. Allocating bandwidth for bursty connections. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 664–673, 1997.

[Law79] E.L. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, 4:339–356, 1979.

[LLRS93] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. Sequencing and scheduling: algorithms and complexity. In S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, editors, *Handbooks in Operations Research and Management Science, Logistics of Production and Inventory*, volume 4, pages 445–522. North-Holland, Amsterdam, 1993.

[LPRS05] R. Levi, M. Pal, R. Roundy, and D.B. Shmoys. Approximation algorithms for stochastic inventory control models. In *Proceedings of the 11th MPS Conference on Integer Programming and Combinatorial Optimization*, pages 306–320, Berlin, Germany, 2005.

[LRS05] R. Levi, R. Roundy, and D.B. Shmoys. Provably near-optimal sampling-based policies for stochastic inventory control models. Technical report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 2005.

[LTRS05] R. Levi, V.A. Truong, R. Roundy, and D.B. Shmoys. Approximation algorithms for capacitated stochastic inventory control problems. Technical report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 2005.

[Mil70] B.L. Miller. On minimizing nonseparable functions defined on the integers with an inventory application. *SIAM Journal on Applied Mathematics*, 21:166–185, 1970.

[Mur03] K. Murota. *Discrete Convex Analysis*. SIAM, 2003.

[MW95] M. Mihail and P. Winkler. On the number of Eulerian orientations of a graph. *Algorithmica*, 16:402–414, 1995.

[Phe62] E. Phelps. The accumulation of risky capital: A sequential utility analysis. *Econometrica*, 30:729–743, 1962.

[Por02] E.L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, 2002.

[Pow07] W.B. Powell. *Approximate Dynamic Programming*. John Wiley and Sons, 2007.

[PP03] K.P. Papadaki and W.B. Powell. An adaptive dynamic programming algorithm for stochastic multiproduct batch dispatch problem. *Naval Research Logistics*, 50:742–769, 2003.

[PRK96] J.D. Papastavrou, S. Rajagopalan, and A.J. Kleywegt. The dynamic and stochastic knapsack problem with deadlines. *Management Science*, 42(12):1706–1718, 1996.

[PWG95] M. Parlar, Y. Wang, and Y. Gerchak. Periodic review inventory model with Markovian supply availability. *International Journal of Production Economics*, 42:131–136, 1995.

[Sah76] S. Sahni. Algorithms for scheduling independent tasks. *Journal of the ACM*, 23:116–127, 1976.

[San95] I. Saniee. An efficient algorithm for the multiperiod capacity expansion of one location in telecommunications. *Operations Research*, 43:187–190, 1995.

[SCB05] D. Simchi-Levi, X. Chen, and J. Bramel. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management.* Springer-Verlag, second edition, 2005.

[Sni80] M. Sniedovich. Preference order stochastic knapsack problems: methological issues. *Journal of the Operational Research Society*, 31(11):1025–1032, 1980.

[SO95] H. Safer and J. Orlin. Fast approximation schemes for multi-criteria flow, knapsack, and scheduling problems. Sloan school working paper 3757-95, Massachusetts Institute of Technology, 1995.

[SP79] M. Sniedovich and M.S. Parks. A preference order dynamic program for a knapsack problem with stochastic rewards. *Journal of the Operational Research Society*, 30(2):141–147, 1979.

[SZ93] J.-S. Song and P.H. Zipkin. Inventory control in a fluctuating demand environment. *Operations Research*, 41:351–370, 1993.

[Vaz01] V.J. Vazirani. *Approximation Algorithms.* Springler-Verlag, 2001.

[VW01] C.P.M. Van Hoesel and A.P.M. Wagelmans. Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. *Mathematics of Operations Research*, 26:339–357, 2001.

[Wei06] D. Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[Whi67] W.D. Whisler. A stochastic inventory model for rented equipment. *Management Science*, 13:640–647, 1967.

[Woe00] G. J. Woeginger. When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing*, 12:57–75, 2000.

[WW58] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.

[Yan06] X. Yan. The determinants and implications of mutual fund cash holdings: Theory and evidence. *Financial Management*, 35:67–91, 2006.

[YL95] C.A. Yano and H.L. Lee. Lot sizing with random yields: A rerview. *Operations Research*, 43:311–334, 1995.

[Zip00] P.H. Zipkin. *Foundations of Inventory Management.* McGraw-Hill, 2000.

Appendix

A Applications of the model

In this section we demonstrate the application of our framework to various dynamic problems.

A.1 Knapsack-related problems

The classical knapsack problem can be formulated as follows. We are given n items, each is associated with integer profit π_i and volume v_i, $i = 1, \dots, n$. We are also given an integer knapsack volume B. The goal is to select a subset of this items with maximum profit without exceeding the knapsack volume. Formally, the problem is

$$\begin{array}{lll} \max & \sum_{i=1}^{n} \pi_i x_i & \\ \text{subject to} & \sum_{i=1}^{n} v_i x_i \leq B & \\ & x_i \in \{0,1\} & i = 1, \dots, n. \end{array} \tag{19}$$

In this formulation, x_i indicates the selected items, π_i is the profit generated from selecting item i, v_i is the space consumption of item i, and $B > 0$ is the capacity of the knapsack.

This problem is also known as the 0/1 knapsack problem, since each item can appear in the knapsack 0 or 1 times. The 0/1 knapsack problem is NP-hard [Kar72] and a first FPTAS for it was developed by Ibarra and Kim [IK75].

A.1.1 The stochastic ordered adaptive knapsack problem

Many stochastic variants of the knapsack problem have been studied in the literature. We consider the ordered adaptive model which is discussed in [DGV04]. In this model we are given a sequence (i.e., ordered set) of n items. While the profit of item i, π_i, is a constant, its volume, v_i, is a random discrete variable with a known distribution. The distribution of v_i is given as the one of D_t in Section 3. The problem is about which of the items to place in the knapsack, and the decision must be made in the order of the items. The actual volume of an item is unknown until we instantiate it by attempting to place it in the knapsack. The goal is to maximize the expected profit from items successfully placed in the knapsack. In the ordered adaptive model the decision whether to put item i in the knapsack is made after knowing the available capacity of the knapsack after executing the previous $i-1$ decisions. In the ordered nonadaptive model all n decisions are made in advance. In [DGV04] the authors give a polynomial time algorithm for the stochastic ordered adaptive knapsack problem. For every $\epsilon > 0$ their algorithm gives a solution whose value is at least the optimal value, at the expense of a slight loss in terms of feasibility, i.e., 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.

Several variants of the stochastic knapsack problem discussed above were considered in the past. Knapsack problems with deterministic volumes and random profits were studied in [CSW93, Hen90, Sni80, SP79]. Another somewhat related variant, known as the stochastic and dynamic knapsack problem, involves items that arrive on-line according to some stochastic process [KP01, PRK96]. Two recent papers due to Kleinberg *et al.* [KRT97] and Goel and Indyk [GI99] consider a stochastic knapsack problem with "chance" constraints. Like our model, they consider items with deterministic profits and random volumes. However, their objective is to find a maximum-profit set of items whose probability of overflowing the knapsack is at most some specified value.

We now present a dynamic program for the problem. Let $z_t(I_t)$ be the expected profit when considering only items t to n, where the remaining available volume in the knapsack is I_t. The recurrence relation is (recall that $x^+ = \max\{x, 0\}$)

$$z_t(I_t) = \max\{E_{v_T}\{\pi_t \delta_{v_t \le I_t} + z_{t+1}((I_t - v_t)^+)\}, z_{t+1}(I_t)\}, \tag{20}$$

for $I_t = 1, \ldots, B$ and $t = 1, \ldots, n$. The boundary conditions are $z_{n+1} \equiv 0$ and $z_t(0) = 0$ for $t = 1, \ldots, n$. The optimal solution is $z_1(B)$.

We note that the first term in the set in (20) is the outcome of placing item t in the knapsack, and the second term is the outcome of not doing so. In order to show that this program fits in our framework we need to reformulate (20) as a maximization over a function of the action space. It is easy to see that (21) below is equivalent to (20), and that it is indeed a maximization over a function of the action space:

$$z_t(I_t) = \max_{x_t=0,1} E_{v_t}\{x_t \pi_t \delta_{v_t \le I_t} + z_{t+1}((I_t - x_t v_t)^+)\}. \tag{21}$$

We next show that program (21) is a maximization nondecreasing dynamic program which fits in our framework, i.e., a dynamic program satisfying Conditions 1, 2 and 4(iii) (recall that the last condition is the counterpart of Condition 3 for maximization problems). We define $D_t = v_t$, $\mathcal{S}_t = [0, \ldots, B]$, $\mathcal{A}_t = \mathcal{A}_t(I_t) = [0, 1]$ for all I_t and $t = 1, \ldots, n$. We define the system dynamics function to be $f_t(I_t, x_t, D_t) = (I_t - x_t D_t)^+, \quad t =$

$1,\dots,n$, and the single-period cost functions to be $g_{n+1}(I_{n+1}) = 0$ and $g_t(I_t, x_t, D_t) = x_t \pi_t \delta_{D_t \le I_t}$, $t = 1,\dots,n$.

We need to show that Conditions 1, 2 and 4(ii) are satisfied. Let us fix the time period t. Clearly, $\mathcal{D}_t, \mathcal{S}_t, \mathcal{A}_t \in \mathbb{Z}$. The logarithm of the cardinality of the maximum element in $\mathcal{S}_t$ is linear in the input size since the input size of B is $\log B$. Hence Condition 1 is satisfied. Clearly, Condition 2 is satisfied as well. As for Condition 4(ii), note that both g_t and f_t are nondecreasing in I_t, f_t is nonincreasing in x_t, g_t is nondecreasing in x_t, and $\mathcal{A}_t(I_t)$ does not depend on I_t. Therefore, this condition is satisfied as well.

A.1.2 The nonlinear knapsack problem

Consider a nonlinear knapsack problem with a separable nondecreasing objective function, a separable nondecreasing "packing constraint", and integer variables. The problem can be formulated as follows:

$$
\begin{array}{lll}
\max & \sum_{i=1}^{n} \pi_i(x_i) & \\
\text{subject to} & \sum_{i=1}^{n} v_i(x_i) \le B & \\
 & 0 \le x_i \le u_i & i = 1,\dots,n \\
 & x_i \in \mathbb{Z}^+ & i = 1,\dots,n.
\end{array} \tag{22}
$$

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, $B > 0$ is the capacity of the knapsack, and $u_i \ge 0$ is an upper bound requirement of x_i. We assume that $\pi_i : \mathbb{Z}^+ \to \mathbb{Z}^+$ and $v_i : \mathbb{Z}^+ \to \mathbb{Z}^+$ are nondecreasing functions that are evaluated in polynomial time, and which satisfy $\pi_i(0) = v_i(0) = 0$. Define $\rho_i : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that $\rho_i(y) := \max\{x \mid v_i(x) \le y\}$ for any $y \in \mathbb{Z}^+$. Note that function ρ_i is nondecreasing and can be evaluated in logarithmic time and calls to function π_i using binary search. Define $R_i := \rho_i(B)$ to be the maximum number of units of item i that can be stored in the knapsack.

Many versions of the nonlinear knapsack problem have been addressed in the literature, but the setting of our problem is quite general because functions π_i and v_i are not restricted to be linear, convex, or concave. For recent surveys on the nonlinear knapsack problem, see [BS02, KPP04, KN09]. When $\rho_i(x_i) = x_i$ for all $i = 1,\dots,n$, the problem is usually referred to as the "resource allocation problem". For detailed discussions of the resource allocation problem, see [IK70].

Problem (22) is a generalization of the classical 0/1 knapsack problem (i.e., when all π_i and v_i are linear, and all $u_i = 1$). Since the classical 0/1 knapsack problem is known to be NP-hard [Kar72], problem (22) is NP-hard as well. As we mentioned above, an FPTAS for the 0/1 knapsack and integer knapsack problems was first developed by Ibarra and Kim [IK75]. Lawler [Law79] has improved the efficiency of Ibarra and Kim's FPTAS and has discussed its extension to the nonlinear case. However, Lawler's approximation scheme is no longer polynomial when it is applied to the nonlinear knapsack problem. Hochbaum [Hoc95] has demonstrated that Lawler's approximation scheme is implementable in polynomial time when π_i is concave and v_i is convex for $i = 1,\dots,n$. Safer and Orlin[5] give a more has given an FPTAS to our problem in a technical report (see pages 26-29 in [SO95]). They deal with the case where π_i is (general) nondecreasing and v_i is linear. This is without loss of generality, since (22) can be formulated in this setting. Very recently, and independently to our work, Kameshwaran and Narahari [KN09] give an FPTAS to a special case where the cost function is a piecewise-linear monotone function that is represented explicitly by tuples of breakpoints, slopes, and costs at breakpoints. Their algorithm is not simple and makes $O(n^3)$ calls to linear programs. Our solution assumes that the cost function is given imlicitely as an oracle, is much simpler, and does not involve linear programs solving.

Next, we present a dynamic program for problem (22). Define $z_t(I)$ as the maximum total profit from

[5] Note from Nir: Jim, could you please summarize your work in a better way?

items $t, \ldots, n$ given that the available knapsack space is I. The recurrence relation is

$$z_t(I_t) := \max_{x_t \in [0,\ldots,\min\{u_t, \rho_t(I_t)\}]} \{\pi_t(x_t) + z_{t+1}(I_t - v_t(x_t))\},$$

for $I_t = 1, \ldots, B$ and $t = 1, \ldots, n$. The boundary conditions are $z_{T+1}(I_t) := 0$, for $I_t \geq 0$. The optimal solution value is $z_1(B)$.

We show next that problem (22) is a maximization nondecreasing DP that fits into our framework, where π and v_i are general nondecreasing integer-valued functions. Since the problem is deterministic, we omit the discrete random variables from our functions. We define $T := n, g_{T+1} \equiv 0$, $g_t(I_t, x_t) := \pi_t(x_t)$ and $f_t(I_t, x_t) := I_t - v_t(x_t)$. The problem is to find $z^*(B)$ in (4). The state space $\mathcal{S}_t$ describes the capacity of the knapsack after storing in it items of type $t, \ldots, T$, hence $\mathcal{S}_t \subseteq [0, \ldots, B]$. We define the boundary state space $\mathcal{S}_{T+1}$ to be $\mathcal{S}_T$. The action space $\mathcal{A}_t$ describes the number of units of item t to put in the knapsack. Clearly $\mathcal{A}_t \subseteq [0, \ldots, R_i]$. We note that $\mathcal{A}_t(I_t) = [0, \ldots, \min\{u_t, \rho_t(I_t)\}]$. Since v_t are computable in polynomial time we get that Condition 1 holds, and since π_t are computable in polynomial time we get that Condition 2 is satisfied. We conclude by noting that it is straightforward to check that Condition $4(ii)$ holds as well.

Note that the nonlinear knapsack problem has numerous applications in areas such as capacity planning, production planning, and capital budgeting (see, for example, [IK70] and [BS02]). The problem with nonconcave and nonconvex functions is particularly important, because elements such as fixed charges may result in a nonconcave profit function π_i, while economies of scales may result in a nonconvex packing function v_i.

Note also that in a similar way we can provide an FPTAS for the corresponding nonlinear minimization knapsack problem

$$\begin{array}{lll} \min & \sum_{i=1}^n \pi_i(x_i) & \\ \text{subject to} & \sum_{i=1}^n v_i(x_i) \geq B & \\ & 0 \leq x_i \leq u_i & i = 1, \ldots, n \\ & x_i \in \mathbb{Z}^+ & i = 1, \ldots, n. \end{array} \tag{23}$$

A.1.3 Dynamic Capacity Expansion

We consider a multi-period capacity expansion problem in telecommunication network planning. The problem is defined as follows: 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:

$$\begin{array}{lll} \min & \sum_{t=1}^T \sum_{i=1}^n \pi_{t,i}(x_{t,i}) & \\ \text{subject to} & \sum_{j=1}^t \sum_{i=1}^n v_i x_{j,i} \geq C_t & t = 1, \ldots, T \\ & x_{t,i} \in \mathbb{Z}^+ & t = 1, \ldots, T;\ i = 1, \ldots, n. \end{array} \tag{24}$$

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, D_t is the added demand requirement (expansion) in period t, and $C_t := \sum_{j=1}^t D_j$ is the accumulated demand over time periods $1, \ldots, t$, where $v_i > 0$ and $D_t \geq 0$. We assume that v_i and D_t are integers for $i = 1, \ldots, n$ and $t = 1, \ldots, T$. The quantity $\pi_{t,i}(x_{t,i})$ is the present value of the monetary resources spent on technology i in time period t, where $\pi_{t,i} : \mathbb{Z}^+ \to \mathbb{Z}^+$ is a nondecreasing function. Note that when $T = 1$, problem (24) becomes a nonlinear minimization knapsack problem (see [KPP04]), which is known to be NP-hard.

Saniee [San95] has studied this multi-period 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 unit cost of technology i and γ is a constant discount factor $(0 < \gamma < 1)$. He has developed a pseudo-polynomial time 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 (24) by modifying Saniee's algorithm and making use of our framework. First, we consider a single time period t, and let $\Pi_t(k)$ be the optimal cost to meet k units of demand in that period (assuming that there is no capacity carried over from the previous period). To obtain the value of $\Pi_t(k)$, we need to solve the following nonlinear minimization knapsack problem:

$$\begin{array}{lll} \min & \sum_{i=1}^{n} \pi_{t,i}(x_{t,i}) & \\ \text{subject to} & \sum_{i=1}^{n} v_i x_{t,i} \geq k & \\ & x_{t,i} = 0, \ldots, \bar{x}_{t,i} \quad i = 1, \ldots, n. & \end{array} \tag{25}$$

Here, $\bar{x}_{t,i}$ represents an upper bound on $x_{t,i}$. For example, we may set $\bar{x}_{t,i} := \lceil \frac{k}{v_i} \rceil$. In this way problem (25) is an instance of problem (23). Clearly, Π_t is a nondecreasing function, and therefore by Theorem 4.4 we can provide for it a K-approximation set and a corresponding K-approximation function in time which is polynomial in the input size, for every $t = 1, \ldots, T$.

Next, we present a dynamic program for problem (24). Define $z_t(I)$ as the minimum total cost to meet the demands of periods $t, \ldots, T$, given that there are already I units of accumulated capacity available from period $t-1$ (i.e., $I = \sum_{j=1}^{t-1} \sum_{i=1}^{n} v_i x_{j,i}$), for $t = 1, \ldots, T+1$ and $I = 0, \ldots, C_T$. The recurrence relation is

$$z_t(I_t) = \min_{x_t = \max\{0, C_t - I_t\}, \ldots, C_T} \{\Pi_t(x_t) + z_{t+1}(\min\{I_t + x_t, C_T\})\}$$

for $I_t = 0, \ldots, C_T$ and $t = 1, \ldots, T$. The boundary conditions are $z_{T+1}(I_t) := 0, \quad I_t \geq 0$. The optimal solution value is $z_1(0)$.

In this paragraph we show that problem (24) is a nonincreasing DP that fits into our framework. Since the problem is deterministic the random variable accepts one value, D_t, with probability 1. We define $g_{T+1} \equiv 0$, $g_t(I_t, x_t, D_t) := \Pi_t(x_t)$ and $f_t(I_t, x_t, D_t) := I_t + x_t$. Since Π_t is nondecreasing, $g_t(I_t, x_t, D_t)$ is constant in its first variable and is nondecreasing in its second variable. Clearly $f_t(I_t, x_t, D_t)$ is nondecreasing in both its two first variables. The problem is to find $z^*(0)$ in (4). The state space $\mathcal{S}_t$ describes the overall capacity of the network in time period t. In an optimal solution it suffices the consider $\mathcal{S}_t = [0, \ldots, C_T]$ for $t = 1, \ldots, T+1$. The action space $\mathcal{A}_t$ describes the amount of capacity expansion from time period $t-1$ to t. Here $\mathcal{A}_t = [0, \ldots, C_T]$ and $\mathcal{A}_t(I_t) = [(C_t - I_t)^+, \ldots, C_T]$ for $t = 1, \ldots, T$ and $I_t = 0, \ldots, C_T$. Directly from the definitions of $\mathcal{D}_t, \mathcal{S}_t, \mathcal{A}_t$ and g_t we get that Conditions 1, 2 and 3(iii) are satisfied. The "catch" is that we cannot compute g_t exactly, and as explained above we compute a K-approximation of it instead. Due to approximation of approximation (property 7 in Proposition 5.1), the accumulated error remains under control.

We note that our FPTAS trivially extends to the non-linear capacities case, where the capacity of $x_{t,i}$ items of technology i in time period t is a nondecreasing function $v_i : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$. We also note that our FPTAS trivially extends to the case of stochastic added demand.

It is not difficult to check that we can solve in an analogous way the following maximization version of problem (24):

$$\begin{array}{lll} \max & \sum_{t=1}^{T} \sum_{i=1}^{n} \pi_{t,i}(x_{t,i}) & \\ \text{subject to} & \sum_{j=1}^{t} \sum_{i=1}^{n} v_i x_{j,i} \leq C_t & t = 1, \ldots, T \\ & x_{t,i} \in \mathbb{Z}^+ & t = 1, \ldots, T;\ i = 1, \ldots, n. \end{array} \tag{26}$$

Note that problem (26) is a generalization of the (linear) multi-period knapsack problem [Faa81]. Thus, our FPTAS is also applicable to the multi-period knapsack problem.

A.2 Logistics and operations management

A.2.1 Machine scheduling

Consider the following machine scheduling problem: There is a 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. We assume that function ρ_j can be evaluated in polynomial time. The objective is to determine the job processing times and to schedule the jobs onto 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. 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 resource 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, polynomially bounded by the input size.

Note that the special case in which all job compressions are prohibitively expensive (denoted as $1||\sum w_j \delta_j$ in the machine scheduling literature) is already NP-hard [LLRS93]. Thus, our problem is also NP-hard. Cheng *et al.* [CCLL98] 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 we consider a general nonincreasing resource consumption function.

We first present a pseudo-polynomial-time dynamic program for our problem: We renumber the jobs such that $d_1 \geq d_2 \geq \cdots \geq d_n$. (Note: There exists an optimal solution in which all on-time jobs are arranged in nondecreasing order of due dates.) Let $z_j(I_t)$ be the minimum total cost of a partial schedule containing $J_j, \ldots, J_n$, given that the total processing time of the on-time jobs is no greater than I_t. The recurrence relation is

$$z_t(I_t) = \min\{\min_{x_t=0,\ldots,I_t}\{z_{t+1}(\min\{I_t - x_t, d_{t+1}\}) + \rho_t(x_t)\}, z_{t+1}(\min\{I_t, d_{t+1}\}) + w_t\}, \tag{27}$$

for $t = 1, \ldots, n$ and $I_t = 0, \ldots, d_t$, and the boundary condition is $z_{T+1}(I_{T+1}) = 0$, for $I_{T+1} = 0, \ldots, d_1$. The optimal solution value is $z_1(d_1)$. It is easy to see that we can replace (27) with

$$z_t(I_t) = \min_{x_t=0,\ldots,I_t}\{z_{t+1}(\min\{I_t - x_t, d_{t+1}\}) + \min\{\rho_t(x_t), w_t\}\}. \tag{28}$$

We show next that problem (28) is a nonincreasing DP that fits into our framework. Since the problem is deterministic, d_t is the only realization of D_t. We define $T := n, g_{T+1} \equiv 0, g_t(I_t, x_t, D_t) := \min\{\rho_t(x_t), w_t\}$ and $f_t(I_t, x_t, D_t) := \min\{I_t - x_t, D_t\}$. The problem is to find $z^*(d_1)$ in (4). The state space $\mathcal{S}_t$ describes the maximal processing time of the on-time jobs amongst jobs $J_t, \ldots, J_T$, hence $\mathcal{S}_t = [0, \ldots, d_t]$. We define the boundary state space $\mathcal{S}_{T+1}$ to be $\mathcal{S}_T$. The action space $\mathcal{A}_t$ describes the processing time of job J_t. Clearly in an optimal solution $\mathcal{A}_t \subseteq [0, \ldots, d_t]$. We note that $\mathcal{A}_t(I_t) = [0, \ldots, I_t]$. Clearly Condition 1 holds, and since ρ_t are computable in polynomial time we get that Condition 2 is satisfied as well. We conclude by noting that it is straightforward to check that Condition 3(*iii*) holds as well.

A.2.2 Batch disposal

Consider managing a warehouse where a single truck of capacity Q is available to dispatch the goods. The goods are received randomly based on distribution D_t in time period t. D_t is given as explained in Section 3. The question is whether we dispatch the truck and if yes, what is its load. If we use the truck, a fixed cost of K_t is incurred in addition to the per-unit cost of c_t. The goods that remain in the warehouse incur the per-unit holding cost h_t. This problem is studied in [Pow07] and [PP03]. No approximation is known for this problem, nor a complexity result. In this part we provide an FPTAS for this problem and show in Section C.2.1 in the Appendix that this problem is #P-hard. Let I_t be the number of assets at time period

t. Recall that the maximum disposal quantity is Q. Our objective is:

$$\min_{x_1,\dots,x_T} E_D \left\{ h_{T+1}(I_{T+1}) + \sum_{t=1}^{T} (c_t x_t + K_t \delta_{x_t>0} + h_t(I_t + D_t - x_t)) \right\}, \quad (29)$$

subject to the system dynamics $I_{t+1} = I_t + D_t - x_t,\ \ t = 1,\dots,T$, where x_t is the truck load in time period t. Denote by $z_t(I_t)$ the total optimal cost starting at time period t with inventory I_t until time period T. A formulation of this problem as a dynamic program is:

$$z_t(I_t) := \min_{x_t=0,\dots,\min\{Q,I_t\}} E_D \{c_t x_t + K_t \delta_{x_t>0} + h_t(I_t + D_t - x_t) + z_{t+1}(I_t + D_t - x_t)\}$$

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

We show next that Problem (29) is a nondecreasing DP that fits into our framework. We define $g_{T+1} = h_{T+1}$, $g_t(I_t, x_t, D_t) = c_t x_t + K_t \delta_{x_t>0} + h_t(I_t + D_t - x_t)$, and $f_t(I_t, x_t, D_t) = (I_t + D_t - x_t)^+$. (If the truck load in time period t exceeds the inventory at the end of that time period, then the inventory at the beginning of the next time period is 0.) The problem is to find $z^*(0)$ in (4). The state space $\mathcal{S}_t$ describes the number of assets in the beginning of time period t. In an optimal solution it suffices to consider $\mathcal{S}_t = [0,\dots,D^*]$ for $t = 1,\dots,T+1$. The action space $\mathcal{A}_t$ describes the number of assets to be disposed of at time period t. Clearly, $\mathcal{A}_t(I_t) = [0,\dots,Q]$. Directly from the definitions of $\mathcal{D}_t, \mathcal{S}_t, \mathcal{A}_t$ and g_t we get that Conditions 1 and 2 hold. Considering Condition 3(ii), we note that g_{T+1} is increasing, g_t and f_t are increasing in their first variable, and f_t is decreasing in its second variable. Although $g_t(I_t, x_t, D_t)$ is not monotone in its second variable, it is the sum of $c_t x_t + K_t \delta(x_t)$ and $h_t(I_t + D_t - x_t)$, where the first function is increasing in x_t and the second one is decreasing in x_t, so Condition 3(ii) holds.

We note that our FPTAS trivially extends to the non-linear case where the disposal costs and holding costs are nondecreasing functions.

A.2.3 The single-item stochastic inventory control problem

The single-item inventory control problem is one of the most widely studied problems in inventory theory and has more than 50 years long history. It has received considerable attention due to its practical implications and analytical results. Maybe the most well-known result in this area is the Wagner Whitin model [WW58]. We refer the reader to books devoted exclusively to inventory theory and logistics for an in depth coverage of the topic (see for example [SCB05, Por02, Zip00]).

The complexity of the deterministic single-item inventory control problem is well understood. In the classical paper of [FLR80], the authors give complexity results for various variants of the deterministic problem. As for algorithmic results, the problem draws attention until nowadays. The first FPTAS to any of the NP-hard variants of this problem was given only in 2001 by Van Hoesel and Wagelmans [VW01]. It is in the very recent paper of Chubanov, Kovalyov and Pesch [CKP06] that the last NP-hard variant described in [WW58] was given an FPTAS.

Recently there is a growing interest in approximation algorithms for stochastic inventory control problems [LPRS05, LTRS05, LRS05]. However, none of these algorithms is an FPTAS. In a previous work of ours [HKM$^+$06], we give the first FPTAS for several variants of the stochastic single-item inventory control problem. We also show there that the stochastic versions of any of the 3 polynomially solvable variants presented in [FLR80] are #P-hard. While the FPTASs given in [HKM$^+$06] are all ad-hoc algorithms tailored to the specific model studied, we show in this section how to formulate these problems into our framework as Convex DP, and therefore the presented FPTAS applies.

We next describe the stochastic inventory control problem. Let T be the length of the planning horizon. At the beginning of a time period, we first observe the inventory level, and next a replenishment decision is made. If we place an order, it arrives immediately, i.e., we assume there is no lead time. If we allow backlogging, the inventory level can be negative. In this case its absolute value corresponds to the number of units on backlog. If the inventory level is positive, then a holding cost is charged; otherwise, a backlogging cost is occurred. For ease of discussion, we call both of these components the holding cost. The holding cost is accounted for at the end of the time period. Our objective is:

$$\min_{x_1,\ldots,x_T} E_D \left\{ \sum_{t=1}^{T} (c_t(x_t) + h_t(I_t + x_t - D_t)) \right\}, \tag{30}$$

subject to the system dynamics $I_{t+1} = I_t + x_t - D_t, \quad t = 1, \ldots, T$, where x_t is the procurement quantity in time period t, I_t is the inventory level at the beginning of time period t, $c_t(x_t)$ is the procurement cost in time period t, given an order of size x_t, and $h_t(y)$ is the holding cost in time period t, given that the inventory level at the end of time period t is y. We define I_1 to be 0, or in other words, we start with no inventory. We assume the procurement and holding costs are convex for all time period.

The input data for the problem consists of the number of time periods T, and for each time period $t = 1, \ldots, T$ an oracle that computes functions c_t, h_t, and a discrete random variable D_t describing the demand in time period t. D_t is given as explained in Section 3. All demand, procurement and inventory levels are integral. Moreover, the demand and procurement levels are nonnegative. For every $t = 1, \ldots, T$, the procurement cost function c_t and the holding cost function h_t are each functions whose values are nonnegative rational numbers, and the binary size of any of their values is polynomially bounded by the (binary) input size.

We next formulate the problem as a convex DP. Clearly, in an optimal solution it suffices to consider procurements x_t restricted to take values between 0 and D^* where D^* is the total demand over the entire time horizon. Note that $\log D^*$ is polynomially bounded by the input size. We define $\mathcal{S}_t := [-D^*, \ldots, D^*]$, $\mathcal{A}_t := [0, \ldots, D^*]$ and $\mathcal{A}_t(I_t) := [0, \ldots, \min\{D^* - I_t, D^*\}]$ for all I_t and $t = 1, \ldots, T$. We define the system dynamic function to be $f_t(I_t, x_t, D_t) := I_t + x_t - D_t, \quad t = 1, \ldots, T$, and the single-period cost functions to be $g_{T+1}(I_{T+1}) := 0$ and $g_t(I_t, x_t, D_t) := c_t(x_t) + h_t(I_t + x_t - D_t)$, $t = 1, \ldots, T$.

It remains to show that Conditions 1, 2 and 3(i) are satisfied. Let us fix the time period t. Clearly, $\mathcal{D}_t, \mathcal{S}_t, \mathcal{A}_t \in \mathbb{Z}$. The cardinality of $\mathcal{D}_t$ is at most linear in the input size since $d_{t,1}, \ldots, d_{t,n_t}$ are given explicitly. Following our definitions of $\mathcal{S}_t$ and of $\mathcal{A}_t(I)$, their cardinalities are bounded by $2D^*$ for every I, and their kth largest element is found in constant time, for every k. Therefore Condition 1 is satisfied. The properties of the procurement cost function c_t and holding cost function h_t assure that Condition 2 is satisfied as well. As for Condition 3(i), the definitions of g_t and f_t directly imply that this condition is fulfilled as well.

It is easy to see that also the capacitated, discounted, and disposal at a cost versions of the problems can be formulated as convex DPs in a similar way.

A.3 Economics and mathematical finance

In this section we consider a few applications in economics and mathematical finance.

A.3.1 Lifetime Consumption Strategy with Risk Exposure

The following problem is studied in [Phe62]. There is a single consumer who must manage her capital in discrete time periods. She can spend some amount of capital, which is governed by the underlying utility function. The remaining capital yields a stochastic return rate, and, in addition, she receives a fixed amount of wealth in each time period. The problem is to find an optimal consumption strategy.

To formalize this problem, let I_t be the amount of capital on hand at the beginning of time period $t = 1, \ldots, T$. In period t the consumer selects an amount x_t of capital to consume. The utility u_t of consuming x_t units is a nonnegative, nondecreasing, and concave function. Unconsumed capital $I_t - x_t$

grows with respect to an exogenous stochastic process defined by random discrete variable D_t with finite support. In addition to this stochastic growth, the consumer receives a_t units of capital at the end of time period t. The consumer would like to maximize the expected total utility in periods $1, \dots, T$ by making consumption decisions $x_1, \dots, x_t$.

This is a maximization problem with cost function

$$g_t(I_t, x_t, d_t) = u_t(x_t),$$

and transition function

$$f_t(I_t, x_t, d_t) = d_t(I_t - x_t) + a_t \ .$$

For an initial capital I_1 it suffices to set the state space $\mathcal{S}_t$ to be $\{0, \dots, D^t_{\max}(I_1 + \sum_{i=1}^t a_i)\}$, where $D_{\max} = \max_t\{d_{t,n_t}\}$ is the maximal value the random variable D_t can get throughout the time horizon. The action space is $\mathcal{A}_t(I_t) = \{x_t \mid 0 \leq x_t \leq I_t\}$. Clearly, the problem satisfies Conditions 1, 2 and 4(ii), and therefore the presented FPTAS applies.

We prove in Section C.2.2 in the Appendix that this problem is #P-hard. To the best of our knowledge no FPTAS is known for this problem.

A.3.2 Deterministic and stochastic growth models

Growth models have drawn substantial interest in the past due to their importance in understanding economic growth, see, for example, Chapter 5 in [AC03] for a recent treatment of growth models.

The standard growth model considers a single consumer with capital I_t on hand at the beginning of time period $t = 1, \dots, T$, which can attain only a finite number of values. In time period t the capital grows based on a nondecreasing, nonnegative, and strictly concave production function p_t, and there is a rate of return $1 - \delta$ for initial capital I_t $(0 < \delta < 1)$. The consumer has a utility function u_t, which is nonnegative, nondecreasing, and concave. The goal is to find optimal capital consumptions in order to maximize total utility. Deterministic and stochastic versions of the problem have been studied. In the stochastic version the production in time t is subject to a random production shock D_t which is a discrete random variable with finite support. In many cases this production shock depends on the previous shock realization, which requires modeling with two state variables. In this subsection we deal with the special case where D_t is independent of the previous realization. In section 10.3 we deal with the general case.

The action corresponds to the amount of capital x_t to consume. We have

$$g_t(I_t, x_t, d_t) = u_t(x_t) \ ,$$

$$f_t(I_t, x_t, d_t) = d_t p_t(I_t) + (1 - \delta)I_t - x_t \ .$$

With $p_t(I_t)$ strictly concave, there exist a maximal level of capital achievable by this economy given by $\bar{I}_t$ where $\bar{I}_t = \max_I\{d_{t,n_t} p_t(I) + (1 - \delta)I > I\}$. (Recall that d_{t,n_t} is the maximal value D_t can have with positive probability.) Hence, in an optimal policy it suffices to consider state space $\mathcal{S}_t = \{0, \dots, \bar{I}_t\}$. Since we cannot consume more than we have (borrowing is not considered), the action space is $\mathcal{A}_t(I_t) = \{x_t \mid 0 \leq x_t \leq (1 - \delta)I_t + d_{t,1} p_t(I_t)\}$. (Recall that $d_{t,1}$ is the minimal value D_t can have with positive probability.) It is easy to see that Conditions 1, 2, and 4(ii) are satisfied, and therefore the problem exhibits an FPTAS.

We prove in Section C.2.3 in the Appendix that this problem is #P-hard. To the best of our knowledge no FPTAS is known even for the deterministic case.

A.4 Cash management

Consider the following cash management problem based on a problem stated in Dreyfus and Law (pp. 154-155 in [DL77]). A mutual fund would like to decide how much cash to have in its bank account for each of the next T time periods. At the beginning of period t, the cash balance can be increased by selling stocks (at a cost of s_t per dollar value of the stocks), can be decreased by buying stocks (at a cost of b_t per dollar value of the stocks), or can be left constant. Assume that the amount of time required to implement the decision is negligible. During the period (after the decision has been implemented) the mutual fund receives demands for cash from customers redeeming their mutual fund shares and cash inflows from the sale of shares. Let D_t be a discrete integer random variable describing the amount of cash withdrawals due to customers during period t. D_t is given as described in Section 3. Note that D_t may be positive or negative, where the later case means that there are deposits in the mutual fund. If during a period the cash balance falls below zero, then the bank will automatically lend the fund the additional amount. However, the fund must pay the bank an interest charge of k_t per dollar per period (based on the cash balance at the end of period t). Conversely, if the fund has a positive cash balance at the end of period t, then it incurs a cost of l_t per dollar per period due to the fact that the fund's money could have been invested elsewhere. The cash balance at the beginning of period 1 is I_1. We assume that there is a constant rational discount factor $0 < \alpha = \frac{\alpha'}{q} \leq 1,\ \alpha', q \in \mathbb{N}$ and there are no terminal costs. Let D^* be the maximal aggregated positive change throughout the time horizon multiplied by q^T, and let D_* be the minimal aggregated negative change throughout the time horizon multiplied by q^T. We assume w.l.o.g. that $D^* > |D_*|$. By using α' as the discount factor instead of α we get that the cash balance in any period must be an integer between $-D^*$ and D^*. We would like to determine the cash balance in each period so as to minimize the expected total discounted cost for operating the fund.

The first works on the mutual fund cash management problem are of Whisler [Whi67] and Eppen and Fama [EF69], are dated back in the Sixties, and seem to be done concurrently and independently (the problem in the former paper was called as an inventory model for rented equipment problem). Both works characterize the structure of the optimal policy for either the finite or infinite time periods cases Surprisingly, it seems that these works have been forgotten later on, since they are not cited in any of the newer results which we survey next. Yan [Yan06] develops a simplified static model for the problem and only deals with qualitative aspects. Hinderer and Waldmann [HW01] propose a dynamic programming model for the infinite planning horizon and prove several structural properties associated with the corresponding optimal policy. They also consider in short the finite time-horizon and several scenarios of non-independent random variables D_t. In order to compute the optimal value functions, they propose a value iteration algorithm that exploits their structural properties. None of these papers deals with either the computational or hardness issues of this problem. There are also papers on the related problem of corporate cash holdings [KMS98, ACW04, FW06], all of which are concerned with qualitative analysis. [ACW04] also analyze cash flow sensitivity.

A pseudo-polynomial-time dynamic program is given as follows (see [DL77], pp. 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. The recurrence relation is

$$z_t(I_t) = \min E_D \begin{bmatrix} \min_{y > I_t} \{s_t(y - I_t) + L_t(y) + \alpha' \sum_{k=1}^{n_t} z_{t+1}(y - d_{t,k}) p_{t,k}\} \\ L_t(I_t) + \alpha \sum_{k=1}^{n_t} z_{t+1}(y - d_{t,k}) p_{t,k} \\ \min_{y < I_t} \{b_t(I_t - y) + L_t(y) + \alpha' \sum_{k=1}^{n_t} z_{t+1}(y - d_{t,k}) p_{t,k}\} \end{bmatrix} \tag{31}$$

for $t = 1, \ldots, T$ and $I_t \in [\min\{-D^*, I_1\}, \ldots, \max\{D^*, I_1\}]$, where

$$E_D L_t(y) = \sum_{d_{t,i} \in \mathcal{D}_t \mid d_{t,i} \leq y} l_t(y - d_{t,i}) p_{t,i} + \sum_{d_{t,i} \in \mathcal{D}_t \mid d_{t,i} > y} k_t(d_{t,i} - y) p_{t,i} \tag{32}$$

for $y \in [\min\{-D^*, I_1\}, \ldots, \max\{D^*, I_1\}]$. The boundary condition is $z_{T+1}(I_{T+1}) = 0$ for any integer I_{T+1}. The optimal solution value is $\frac{z_1(I_1)}{q^T}$.

We prove in Section C.2.4 in the Appendix that this problem is #P-hard. To the best of our knowledge no FPTAS is known for this problem, even for the special case of static costs (i.e., $s_i = s_j, b_i = b_j, k_i = k_j, l_i = l_j$, for every i, j), stated in [DL77].

An alternative formulation for this problem is as follows. Let $c_t : \mathbb{Z} \to \mathbb{Q}$ be the per dollar transaction cost. For every $x \in \mathbb{Z}^+$ the per dollar value buying stocks cost at time t is $c_t(x) = b_t x$, and for every $x \in \mathbb{Z}^-$ the per dollar value selling stocks cost at time t is $c_t(x) = -s_t x$. Note that c_t is a V-shape nonnegative convex function. Let $h_t : \mathbb{Z} \to \mathbb{Q}$ be the per dollar holding cost. For every $x \in \mathbb{Z}^+$ the per dollar value cost for having positive cash balance at the end of period t is $h_t(x) = l_t x$, and for every $x \in \mathbb{Z}^-$ the per dollar value interest rate paid for negative cash balance at the end of period t is $h_t(x) = -k_t x$. Note that also h_t is a V-shape nonnegative convex function. Using this new notation we can substitute (31) and (32) with

$$z_t(I_t) := \min_{x_t = I_t + \min\{-D^*, I_1\}, \ldots, I_t + \max\{D^*, I_1\}} E_D \left\{ (\alpha')^{T-t}(c_t(x_t) + h_t(I_t - x_t - D_t)) + z_{t+1}(I_t - x_t - D_t) \right\} \tag{33}$$

for $t = 1, \ldots, T$ and $I_t \in [\min\{-D^*, I_1\}, \ldots, \max\{D^*, I_1\}]$, where the boundary condition is $z_{T+1}(I_{T+1}) = 0$ for every integer I_{T+1}.

We next show that problem (33) is a convex DP that fits in our framework. We define $g_{T+1} \equiv 0$, $g_t(I_t, x_t, D_t) := (\alpha')^{T-t}(c_t(x_t) + h_t(I_t - x_t - D_t))$ and $f_t(I_t, x_t, D_t) := I_t - x_t - D_t$. The problem is to find $z^*(I_1)$ in (4). The state space $\mathcal{S}_t$ describes the cash balance in the fund at period t, hence in an optimal solution $\mathcal{S}_t = [\min\{-D^*, I_1\}, \ldots, \max\{D^*, I_1\}]$, for $t = 1, \ldots, T+1$. The action space $\mathcal{A}_t$ describes the value of transactions in period t, where negative values mean selling stocks. Clearly, in an optimal solution $\mathcal{A}_t \subseteq [\min\{I_1 - D^*, -D^*\}, \ldots, \max\{I_1 + D^*, D^*\}]$. We note that $\mathcal{A}_t(I_t) = [I_t + \min\{-D^*, I_1\}, \ldots, I_t + \max\{D^*, I_1\}]$ for $t = 1, \ldots, T$.

Since D^* is a sum of at most T numbers from the input, Condition 1 holds. Clearly Condition 2 is satisfied as well. We conclude by noting that since c_t and h_t are convex functions Condition 3(i) holds as well.

B Proofs for Section 6

Proposition 6.1 (Calculus of K-approximation Sets of unimodal functions) *Let $K_1, K_2 \geq 1$, let $\varphi_1, \varphi_2 : D \to \mathbb{R}^+$ be unimodal functions over a finite domain D of real numbers. Let W_i be a K_i-approximation set of φ_i, for $i = 1, 2$. Let $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ be a (strictly) monotone function, and let $\alpha, \beta \in \mathbb{R}^+$. The following properties hold:*

1. *(monotonicity of approximation sets) every superset $W_1 \subset W' \subset D$ is a K_1-approximation set of φ_1,*
2. *(composition of approximation sets) $\psi^{-1}(W_1) = \{i \mid \psi(i) \in W_1\}$ is a K_1-approximation set of $\varphi_1(\psi)$,*
3. *(linearity of approximation sets) W_1 is a K_1-approximation set of $\alpha + \beta\varphi_1$.*
4. *(maximization of approximation sets) $W_1 \cup W_2$ is a $\max\{K_1, K_2\}$-approximation set of $\max\{\varphi_1, \varphi_2\}$.*

Proof. Properties 2 and 3 follow directly from the definitions of K-approximation sets and functions. We give here a proof for monotonicity of approximation sets and for maximization of approximation sets.

We first prove monotonicity of approximation sets. Since $W' \supset W_1$, we have $D^{\min}, D^{\max} \in W'$ and the first property of K-approximation sets is satisfied. As for boundness, suppose that $x \in W'$ and $x \neq D^{\max}$ and $\mathrm{next}(x, W') \neq \mathrm{next}(x, D)$. We need to prove that $\max\{\varphi(x), \varphi(\mathrm{next}(x, W'))\} \leq K \min\{\varphi(x), \varphi(\mathrm{next}(x, W'))\}$. If $x, \mathrm{next}(x, W') \in W_1$ (i.e., $\mathrm{next}(x, W_1) = \mathrm{next}(x, W')$) then this indeed holds due to the boundness of W_1. Suppose now without loss of generality that $x \notin W_1$. We have

$$\max\{\varphi(x), \varphi(\mathrm{next}(x, W'))\} \leq \max\{\varphi(\mathrm{prev}(x, W_1)), \varphi(\mathrm{next}(x, W_1))\} \leq K\varphi(x),$$

where the first inequality is due to unimodality of φ and the second inequality is due to locality of W_1. If also $\text{next}(x, W') \notin W_1$, then we similarly get

$$\max\{\varphi(x), \varphi(\text{next}(x, W'))\} \leq \max\{\varphi(\text{prev}(x, W_1)), \varphi(\text{next}(x, W_1))\} \leq K\varphi(\text{next}(x, W')), \tag{34}$$

and the boundness of W' follows. If, on the other hand, $\text{next}(x, W') \in W_1$ then (34) still holds because the second inequality is now due to boundness of W_1, and again the boundness of W' follows. It remains to show locality of W'. Let $x \in D \setminus W'$. We have

$$\max\{\varphi(\text{prev}(x, W')), \varphi(\text{next}(x, W'))\} \leq \max\{\varphi(\text{prev}(x, W_1)), \varphi(\text{next}(x, W_1))\} \leq K\varphi(x),$$

where the first inequality is due to unimodality of φ and the second one is due to the locality of W_1. This proves monotonicity of approximation sets.

We next prove maximization of approximation sets. Let $\varphi_{\max}(x) = \max_{i=1,2} \varphi_i(x),\ \forall x$, and $W_{12} = W_1 \cup W_2$. We first note that unimodality is closed under maximization, so $\varphi_{\max}$ is a unimodal function, and a K-approximation set for it is well defined. Clearly, $D^{\min}, D^{\max} \in W_{12}$ and the first property of K-approximation sets is satisfied. We next prove boundness of $\varphi_{\max}$. Due to monotonicity of approximation sets, W_{12} is a K_i-approximation set for $\varphi_i,\ i = 1, 2$. Let $x \in W_{12}$. Due to boundness of φ_1, φ_2 we have

$$\varphi_j(\text{next}(x, W_{12})) \leq \max\{\varphi_j(x), \varphi_j(\text{next}(x, W_{12}))\} \leq K_j \min\{\varphi_j(x), \varphi_j(\text{next}(x, W_{12}))\} \leq K_j\varphi_j(x), \quad j = 1, 2. \tag{35}$$

If there exists $j \in \{1, 2\}$ such that $\varphi_{\max}(x) = \varphi_j(x)$ and $\varphi_{\max}(\text{next}(x, W_{12})) = \varphi_j(\text{next}(x, W_{12}))$ then boundness of $\varphi_{\max}$ follows directly from the one of φ_j. Otherwise, suppose without loss of generality that $\varphi_{\max}(x) = \varphi_1(x)$, $\varphi_{\max}(\text{next}(x, W_{12})) = \varphi_2(\text{next}(x, W_{12}))$ and $\max\{\varphi_{\max}(x), \varphi_{\max}(\text{next}(x, W_{12}))\} = \varphi_{\max}(\text{next}(x, W_{12}))$. Then, due to (35) we have

$$\begin{aligned} \max\{\varphi_{\max}(x), \varphi_{\max}(\text{next}(x, W_{12}))\} &= \varphi_{\max}(\text{next}(x, W_{12})) = \varphi_2(\text{next}(x, W_{12})) \leq K_2\varphi_2(x) \leq \\ &\leq K_2\varphi_{\max}(x) = K_2 \min\{\varphi_{\max}(x), \varphi_{\max}(\text{next}(x, W_{12}))\}, \end{aligned}$$

and the boundness of $\varphi_{\max}$ follows.

It remains to prove the locality of $\varphi_{\max}$. Let $x \notin W_{12}$. If there exists $j \in \{1, 2\}$ such that $\varphi_{\max}(\text{prev}(x, W_{12})) = \varphi_j(\text{prev}(x, W_{12}))$ and $\varphi_{\max}(\text{next}(x, W_{12})) = \varphi_j(\text{next}(x, W_{12}))$, then the locality of $\varphi_{\max}$ follows directly from the one of φ_j. Otherwise, suppose without loss of generality $\varphi_{\max}(\text{prev}(x, W_{12})) = \varphi_1(\text{prev}(x, W_{12}))$, $\varphi_{\max}(\text{next}(x, W_{12}))$ $\varphi_2(\text{next}(x, W_{12}))$ and $\max\{\varphi_{\max}(\text{prev}(x, W_{12})), \varphi_{\max}(\text{next}(x, W_{12}))\} = \varphi_{\max}(\text{next}(x, W_{12}))$. Then, the locality of φ_1, φ_2 implies

$$\begin{aligned} &\max\{\varphi_{\max}(\text{prev}(x, W_{12})), \varphi_{\max}(\text{next}(x, W_{12}))\} = \max\{\varphi_1(\text{prev}(x, W_{12})), \varphi_2(\text{next}(x, W_{12}))\} \leq \\ &\leq \max\{(\max\{\varphi_1(\text{prev}(x, W_{12})), \varphi_1(\text{next}(x, W_{12}))\}), (\max\{\varphi_2(\text{prev}(x, W_{12})), \varphi_2(\text{next}(x, W_{12}))\})\} \leq \\ &\leq \max\{K_1\varphi_1(x), K_2\varphi_2(x)\} \leq \max\{K_1, K_2\}\varphi_{\max}(x), \end{aligned}$$

as needed. □

Proposition 6.2 (Calculus of K-approximation Sets of monotone functions) *Let $K_1, K_2 > 1$, and let $\varphi_1, \varphi_2, \varphi_3 : D \rightarrow \mathbb{R}^+$ be monotone functions of the same kind (i.e., either all are nondecreasing or all are nonincreasing) over 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.*

Proof. Let $W_{12} = W_1 \cup W_2$. Without loss of generality we assume that φ_1, φ_2 are both nondecreasing. We first prove summation of approximation sets. Let $\varphi_{\sum}(x) = \varphi_1(x) + \varphi_2(x)$, $\forall x$. We first note that $\varphi_{\sum}$ is also nondecreasing, so a K-approximation set for it is well defined. Since $W_{12} \supset W$, we have $D^{\min}, D^{\max} \in W_{12}$ so the first property of K-approximation sets is satisfied. As for boundness, let $x \in W_{12}$. By monotonicity of approximation sets W_{12} is a K_i-approximation set of φ_i, $i = 1, 2$. Since $\varphi_{\sum}$ is nondecreasing we get

$$\begin{array}{rclcll} \max\{\varphi_{\sum}(x), \varphi_{\sum}(\text{next}(x, W_{12}))\} & = & \varphi_{\sum}(\text{next}(x, W_{12})) & = & \sum_{j=1,2} \varphi_j(\text{next}(x, W_{12})) & \leq \\ & \leq & \sum_{j=1,2} K_j \varphi_j(x) & \leq & \max\{K_1, K_2\} \varphi_{\sum}(x) & = \\ & & & = & \max\{K_1, K_2\} \min\{\varphi_{\sum}(x), \varphi_{\sum}(\text{next}(x, W_{12}))\}, & \end{array}$$

where the first inequality is due to (35). Therefore, boundness of $\varphi_{\max}$ over W_{12} holds. Locality is satisfied by observing that the monotonicity of $\varphi_{\sum}$ implies that $\arg\min \varphi_{\sum} \in \{D^{\min}, D^{\max}\} \subseteq W_{12}$ and applying Proposition 4.3.

We next prove minimization of approximation sets. Let $\varphi_{\min}(x) = \min\{\varphi_1(x), \varphi_2(x)\}$, $\forall x$. We first note that $\varphi_{\min}$ is also nondecreasing, so a K-approximation set for it is well defined. Since $W_{12} \supset W$, we have $D^{\min}, D^{\max} \in W_{12}$ and the first property of K-approximation sets is satisfied. As for boundness, let $x \in W_{12}$. By monotonicity of approximation sets W_{12} is a K_i-approximation set of φ_i, $i = 1, 2$. We have

$$\begin{array}{rclcl} \max\{\varphi_{\min}(x), \varphi_{\min}(\text{next}(x, W_{12}))\} & = & \varphi_{\min}(\text{next}(x, W_{12})) & = & \min_{j=1,2} \varphi_j(\text{next}(x, W_{12})) \leq \\ & \leq & \min_{j=1,2} K_j \varphi_j(x) & \leq & \max\{K_1, K_2\} \min_{j=1,2} \varphi_j(x) = \\ & = & \max\{K_1, K_2\} \varphi_{\min}(x) & = & \max\{K_1, K_2\} \min\{\varphi_{\min}(x), \varphi_{\min}(\text{next}(x, W_{12}))\}, \end{array}$$

where the first inequality is due to (35). Therefore, boundness of $\varphi_{\min}$ over W_{12} holds. We prove locality of $\varphi_{\min}$ over W_{12} similarly to the way we proved it for summation of approximation sets.

We last prove approximation of approximation sets. The fact that φ_1 is a K_2-approximation of the restriction of φ_2 over W_1 is equivalent to

$$\varphi_2(z) \leq \varphi_1(z) \leq K_2 \varphi_2(z), \quad \forall z \in W_1. \tag{36}$$

We have

$$\hat{\varphi}_1(z) = \varphi_1(\text{next}(z, W_1)) \geq \varphi_2(\text{next}(z, W_1)) \geq \varphi_2(z), \quad \forall z \in D \setminus W_1, \tag{37}$$

where the equality is due to the definition of the approximation induced by an approximation set, (**??**) implies the first inequality, and the second inequality is true because φ is nondecreasing. We also have

$$\hat{\varphi}_1(z) = \varphi_1(\text{next}(z, W_1)) \leq K_1 \varphi_1(\text{prev}(z, W_1)) \leq K_1 K_2 \varphi_2(\text{prev}(z, W_1)) \leq K_1 K_2 \varphi_2(z), \quad \forall z \in D \setminus W_1, \tag{38}$$

where the first inequality is due to locality of approximation sets, (**??**) implies the second inequality, and the last one is due to the monotonicity of φ_2. We conclude from (36)-(38) that $\hat{\varphi}_1$ is a K_1K_2-approximation of φ_2 over D. □

Proposition 6.3 (Calculus of K-approximation Sets of convex functions) *Let $K_1, K_2 > 1$, let $\varphi_1, \varphi_2 : D \to \mathbb{Z}^+$ 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:*

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

Proof. Let $W_{12} = W_1 \cup W_2$ and let $\varphi_{\sum}(x) = \varphi_1(x) + \varphi_2(x),\ \forall x$. Since the sum of two convex functions is a convex function, we get that $\varphi_{\sum}$ is a convex function, so a K-approximation set for it is well defined. We prove the first two properties similarly to the way we proved them for summation of approximation for monotone functions.

As for the locality of K-approximation sets, let $x \notin W_{12}$. We have

$$\begin{aligned}
&\max\{\varphi_{\sum}(\text{prev}(x, W_{12})), \varphi_{\sum}(\text{next}(x, W_{12}))\} = \\
&= \max\{\varphi_1(\text{prev}(x, W_{12})) + \varphi_2(\text{prev}(x, W_{12})), \varphi_1(\text{next}(x, W_{12})) + \varphi_2(\text{next}(x, W_{12}))\} \leq \\
&\leq \max\{\varphi_1(\text{prev}(x, W_{12})), \varphi_1(\text{next}(x, W_{12}))\} + \max\{\varphi_2(\text{prev}(x, W_{12})), \varphi_2(\text{next}(x, W_{12}))\} \leq \\
&\leq \textstyle\sum_{j=1,2} K_j \varphi_j(x) \leq \max\{K_1, K_2\}\varphi_{\sum}(x),
\end{aligned}$$

where the second inequality is due to the locality applied on each of φ_1, φ_2 over W_{12}. □

C Hardness results

C.1 Proofs of results in Section 8

Theorem 8.2. *A convex DP where either $b \notin \{-1, 0, 1\}$, or the action space is not a contiguous interval, does not admit an FPTAS unless P=NP.*

The proof makes a transformation from the partition problem (Problem SP12 in [GJ79]).

Partition

Instance: Finite set V of nonnegative integer numbers.
Question: Is there a subset $V' \subseteq V$ such that $\sum_{v \in V'} v = \sum_{v \in V \setminus V'} v$?

Proof. Let $M = \frac{\sum_{v \in V} v}{2}$. We prove first the case where the coefficients of the second variable of f are not restricted to be -1,0 or 1. Given an instance of the partition problem where the cardinality of V is T, let us consider the following deterministic DP (for simplicity, we slightly abuse notation by omitting the random variable D_t from the equations): $\mathcal{S}_t = [-M, \ldots, M],\ t = 1, \ldots, T+1,\ \mathcal{A}_t(I_t) = \mathcal{A}_t = \{0, 1\},\ t = 1, \ldots, T$. $g_t(I_t, x_t) = v_t x,\ f_t(I_t, x_t) = I_t - v_t x,\ t = 1, \ldots, T$, and finally $G_{T+1}(I_{T+1}) = M|I_{T+1}|$. Clearly, this program is a convex DP except for the requirement that the coefficient of the second variable of f_t is neither -1, 0, or 1.

We note that an optimal solution starting with state M, $z_1(M)$ as in (5), is either M, or at least $2M - 1$. Hence, we can distinguish between these cases by calculating a 1.5 approximation for $z_1(M)$. We conclude the proof by noting that there exists a solution to the partition problem if and only if $z_1(M) = M$.

The proof for the case where the action space is not contiguous is similar. □

C.2 Proofs of results in Section A

In this section we show that hardness of several stochastic problems by making a transformation from the Kth largest subset problem, which is known to be #P-hard (see for example problem SP20 in page 225 in [GJ79]).

Problem: Kth largest subset

Instance: *A finite set $A = \{a_1, \ldots, a_n\}$ of nonnegative integers and positive integer numbers $K \leq 2^n$ and B.*
Question: *Are there K or more distinct subsets $A' \subseteq A$ for which $\sum_{a \in A'} \leq B$?*

Remark: This problem is named the "Kth largest subset problem", but it actually looks for the Kth smallest subset of A.

C.2.1 Batch disposal

Theorem C.1. *The stochastic batch disposal problem is #P-hard even in the case of linear procurement and holding costs.*

Proof. Given an instance of the Kth largest subset problem, we transform it into the following instance of the stochastic batch disposal problem. Let $M = 2^n \max\{2^n, \sum_{a \in A} a\}$. We have

- $T = n + 3$ time periods, labeled $1, \ldots, n$.
- Maximum disposal quantity is $Q = \sum_{a \in A} a$.
- Demand in time period 1 is $\sum_{a \in A} a$ with probability 1. The demand in time period $t = n+2, n+3$ is 0 with probability 1. The demand in time period $n + 1 \geq t > 1$ is $\begin{cases} 0 & \text{with probability } \frac{1}{2} \\ a_{t-1} & \text{with probability } \frac{1}{2}. \end{cases}$
- Fixed disposal cost K_t is 0 for all time periods.
- Variable disposal cost in time period t is $c_t(x) = \begin{cases} (2^n - K)x & \text{for } t = 1; \\ Mx & \text{for } t = 2, \ldots, n+1; \\ 2^n x & \text{for } t = n+2; \\ 0 & \text{for } t = n+3. \end{cases}$
- Holding cost in time period t is $h_t(x) = \begin{cases} 0 & \text{for } t = 1, \ldots, n+3; \\ Mx & \text{for } t = n+4; \end{cases}$

Since $c_t(x)$ is much larger than each $c_1(x), c_{n+2}(x)$ and c_{n+3}, for $t = 2, \ldots, n+1$, it is clear that the optimal policy is to order nothing in time periods $2, \ldots, n+1$, order $\sum_{a \in A} a$ in time period $n+3$, order some number S in time period $n+3$ and order the difference between the realized demand and $S + \sum_{a \in A} a$ in time period $n+2$ (note that since the demand in time periods $n+2$ and $n+3$ is 0, the realized demand is known already at the end of time period $n+1$). Let D be the realized demand in periods $2, \ldots, n+1$. If S is larger than D, then the inventory at hand in the end of time period $n+1$ does not exceed $\sum_{a \in A} a$. In this case the cost is $(2^n - K)S$, due to batch disposal cost in time period 1, and free disposal in time period $n+3$. If S is smaller than the realized demand D, then $D - S$ units will be disposed in time period $n+2$. In this case the cost is $(2^n - K)S + 2^n(D - S)$, due to the disposal cost in time periods $1, n+2$ and $n+3$.

This reduces to the newsvendor problem, where the unit cost of ordering one item too many is $m = 2^n - K$, and the cost of ordering one item too few is $l = 2^n - (2^n - K) = K$. Therefore, the optimal decision is to produce the minimum amount S such that $\text{Prob}(D \leq S) \geq \frac{l}{m+l} = \frac{K}{2^n}$, where D is a random variable describing the demand (see, e.g., [SCB05], Section 8.2.1). Note that in our case we have

$$\text{Prob}(D \leq S) = \frac{\text{number of subsets } I \subseteq \{1, \ldots, n\} \text{ such that } \sum_{i \in I} a_i \leq S}{2^n}.$$

Therefore, S equals to the sum of the Kth smallest subset. We conclude the proof by noting that the answer for the given instance of the Kth largest subset problem is affirmative if and only if $S \leq B$. □

C.2.2 Lifetime consumption strategy with risk exposure

Theorem C.2. *The lifetime consumption strategy with risk exposure problem is #P-hard.*

Proof. We transform the Kth largest subset problem into our problem. Given an arbitrary instance of the Kth largest subset problem, we define $\bar{a}_j = a_j + \frac{1}{2^j}$ $(j = 1, \ldots, n)$ and $\bar{B} = B + 1 - \frac{1}{2^n}$. Clearly, for any $U \subseteq \{1, \ldots, n\}$, $\sum_{j \in U} a_j \leq B$ if and only if $\sum_{j \in U} \bar{a}_j \leq \bar{B}$. Thus, the Kth largest subset problem is equivalent to the problem of determining whether there exist at least K distinct subsets $A' \subseteq \{\bar{a}_1, \ldots, \bar{a}_n\}$ such that $\sum_{\bar{a} \in A'} \bar{a} \leq \bar{B}$. 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 $U \subseteq \{1, \ldots, n\}$,

$$1 + \frac{\sum_{j \in U} \bar{a}_j}{M} \leq \prod_{j \in U} \left(1 + \frac{\bar{a}_j}{M}\right) \leq 1 + \frac{\sum_{j \in U} \bar{a}_j}{M} + \frac{2^n \bar{a}_{\max}^2}{M^2},$$

which implies that

$$1 + \frac{\sum_{j \in U} \bar{a}_j}{M} \leq \prod_{j \in U} \left(1 + \frac{\bar{a}_j}{M}\right) \leq 1 + \frac{\sum_{j \in U} \bar{a}_j}{M} + \frac{1}{2^{n+1}M}.$$

Let $\hat{B} = 1 + \frac{\bar{B}}{M} + \frac{1}{2^{n+1}M}$. Hence, for any $U \subseteq \{1, \ldots, n\}$, $\sum_{j \in U} \bar{a}_j \leq \bar{B}$ if and only if $\prod_{j \in U}(1 + \frac{\bar{a}_j}{M}) \leq \hat{B}$ (note that if $\prod_{j \in U}(1 + \frac{\bar{a}_j}{M}) \leq \hat{B}$, then $\sum_{j \in U} \bar{a}_j \leq \bar{B} + \frac{1}{2^{n+1}}$, which implies that $\sum_{j \in U} \bar{a}_j \leq \bar{B}$, because $\bar{B}, \bar{a}_1, \ldots, \bar{a}_n$ are all multiples of $\frac{1}{2^n}$). Hence, for any $U \subseteq \{1, \ldots, n\}$, $\sum_{j \in U} a_j \leq B$ if and only if $\prod_{j \in U}(1 + \frac{\bar{a}_j}{M}) \leq \hat{B}$.

We now construct the following instance of our problem:

- Number of time periods, $T = n + 1$.
- Initial capital on hand, $I_1 = M$.
- Capital received by the consumer at the end of period t, $a_t = 0$ for $t = 1, \ldots, T$.
- Utility function $u_t(x_t) = \begin{cases} x_t, & \text{if } t = 1; \\ 0, & \text{if } t = 2, \ldots, T-1; \\ \min\left\{\frac{2^n}{K} x_t,\, M\right\}, & \text{if } t = T. \end{cases}$
- $\text{Prob}(D_t = 1) = \text{Prob}(D_t = 1 + \frac{\bar{a}_t}{M}) = \frac{1}{2}$ for $t = 1, \ldots, T-1$.

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. Since the utility is zero in periods $2, \ldots, T-1$, the optimal decision is to make consumption in periods 1 and T only, with $x_T = I_T$. The decision is to select a consumption amount x_1 in period 1. Let $S = \left\{t \mid D_t = 1 + \frac{\bar{a}_t}{M}; 1 \leq t \leq T-1\right\}$. Hence, $H = \prod_{t \in S}\left(1 + \frac{\bar{a}_t}{M}\right)$. Note that

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

Therefore,

$$1 \leq H \leq 1 + \frac{1}{2^n}. \tag{39}$$

If x_1 is less than $I_1 - \frac{M}{H}$, then $I_T = (I_1 - x_1)H > M$ and the utility in period T is M, regardless of how much x_1 is below $I_1 - \frac{M}{H}$. 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 $I_1 - \frac{M}{H}$, then the utility in period T is $\frac{2^n}{K}(I_1 - x_1)H$. Thus, the unit cost of setting x_1 too high is $\frac{2^n}{K}H - 1$. By (39), this unit cost is at least $\frac{2^n-K}{K}$ and at most $\frac{2^n-K+1}{K}$.

Consider the scenario in which the unit cost of setting x_1 too high is $\frac{2^n-K}{K}$. This scenario reduces to a newsvendor problem, where the optimal decision is to select the smallest value of x_1, denoted as x_1', such that

$$\text{Prob}\left(H \leq \frac{M}{I_1 - x_1'}\right) = \frac{1}{1 + \frac{2^n-K}{K}} = \frac{K}{2^n},$$

where

$$\text{Prob}\left(H \leq \frac{M}{I_1 - x_1'}\right) = \frac{\text{number of subsets } U \subseteq \{1, \ldots, n\} \text{ such that } \prod_{t \in U}(1 + \frac{\bar{a}_t}{M}) \leq \frac{M}{I_1 - x_1'}}{2^n}.$$

Next, consider the scenario in which the unit cost of setting x_1 too high is $\frac{2^n-K+1}{K}$. Note that $\frac{K-1}{2^n} < 1/\left(1 + \frac{2^n-K+1}{K}\right) < \frac{K}{2^n}$ (i.e., the critical fractile is in between $\frac{K-1}{2^n}$ and $\frac{K}{2^n}$), and that

$$\text{Prob}\left(H \leq \frac{M}{I_1 - x_1}\right) = \frac{\text{number of subsets } U \subseteq \{1, \ldots, n\} \text{ such that } \prod_{t \in U}(1 + \frac{\bar{a}_t}{M}) \leq \frac{M}{I_1 - x_1}}{2^n},$$

which is a multiple of $\frac{1}{2^n}$. Hence, this scenario again reduces to a newsvendor problem, where the optimal decision is to select the smallest value of x_1, denoted as x_1'', such that

$$\text{Prob}\left(H \leq \frac{M}{I_1 - x_1''}\right) = \frac{K}{2^n}.$$

The above analysis of the two scenarios implies that $x_1' = x_1''$. Thus, in the constructed problem, the optimal decision is to select the smallest x_1 such that there are K subsets $U \subseteq \{1, \ldots, n\}$ satisfying $\prod_{t \in U}(1 + \frac{\bar{a}_t}{M}) \leq \frac{M}{I_1 - x_1}$. This is equivalent to selecting the smallest x_1, denoted as x_1^*, such that there are K subsets $U \subseteq \{1, \ldots, n\}$ satisfying

$$\sum_{t \in U} a_t \leq \frac{M^2}{I_1 - x_1} - M - \frac{1}{2^{n+1}} + \frac{1}{2^n} - 1. \tag{40}$$

Since x_1^* is the smallest x_1 satisfying condition (40), we have $\sum_{t \in U} a_t = \frac{M^2}{I_1 - x_1^*} - M - \frac{1}{2^{n+1}} + \frac{1}{2^n} - 1$. Therefore, if we solve the constructed problem instance optimally, then we can determine the answer to the Kth largest subset problem as follows: There exist at least K subsets $A' \subseteq A$ with $\sum_{a \in A'} a \leq B$ if and only if $\frac{M^2}{I_1 - x_1^*} - M - \frac{1}{2^{n+1}} + \frac{1}{2^n} - 1 \leq B$. ◻

C.2.3 Stochastic growth model

Theorem C.3. *The stochastic growth problem is #P-hard.*

Proof. The proof is similar to the one of Theorem C.2. Let $\bar{a}_j$ and M be as defined in the proof of Theorem C.2. Given an arbitrary instance of the Kth largest subset problem, we transform it into the following instance of our problem:

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

- Production function, $p_t(I_t) = I_t$, for $I_t \geq 0$ and $t = 1, \ldots, T$.
- Rate of return of capital, $1 - \delta = 0$.
- Utility function $u_t(x_t) = \begin{cases} x_t, & \text{if } t = 1; \\ 0, & \text{if } t = 2, \ldots, T-1; \\ \min\left\{\frac{2^n}{K} x_t,\, M\right\}, & \text{if } t = T. \end{cases}$
- $\text{Prob}(D_t = 1) = \text{Prob}(D_t = 1 + \frac{\bar{a}_t}{M}) = \frac{1}{2}$ for $t = 1, \ldots, T-1$.
- $\text{Prob}(D_T = 1) = 1$.

We conclude the proof by following the proof of Theorem C.2. □

C.2.4 Cash management

In this section we show that the cash management problem is NP-hard. We make a transformation from the Kth largest subset problem, which is known to be NP-hard.

Theorem C.4. *The cash management problem is #P-hard.*

The proof of this theorem is essentially identical to the hardness proof we give in [HKM$^+$06] for the stochastic single-item inventory control problem. The equivalence here for production cost is transaction cost.