Improved Approximation Results for Stochastic Knapsack Problems

Anand Bhalgat [*]　　　Ashish Goel [†]　　　Sanjeev Khanna [‡]

Abstract

In the *stochastic knapsack* problem, we are given a set of items each associated with a probability distribution on sizes and a profit, and a knapsack of unit capacity. The size of an item is revealed as soon as it is inserted into the knapsack, and the goal is to design a *policy* that maximizes the expected profit of items that are successfully inserted into the knapsack. The stochastic knapsack problem is a natural generalization of the classical knapsack problem, and arises in many applications, including bandwidth allocation, budgeted learning, and scheduling.

An *adaptive* policy for stochastic knapsack specifies the next item to be inserted based on observed sizes of the items inserted thus far. The adaptive policy can have an exponentially large explicit description and is known to be PSPACE-hard to compute. The best known approximation for this problem is a $(3 + \epsilon)$-approximation for any $\epsilon > 0$. Our first main result is a relaxed PTAS (Polynomial Time Approximation Scheme) for the adaptive policy, that is, for any $\epsilon > 0$, we present a poly-time computable $(1+\epsilon)$-approximate adaptive policy when knapsack capacity is relaxed to $1+\epsilon$. At a high-level, the proof is based on transforming an arbitrary collection of item size distributions to *canonical* item size distributions that admit a compact description. We then establish a *coupling* that shows a $(1 + \epsilon)$-approximation can be achieved for the original problem by a canonical policy that makes decisions at each step by observing events drawn from the sample space of canonical size distributions. Finally, we give a mechanism for approximating the optimal canonical policy.

Our second main result is an $(8/3 + \epsilon)$-approximate adaptive policy for any $\epsilon > 0$ without relaxing the knapsack capacity, improving the earlier $(3 + \epsilon)$-approximation result. Interestingly, we obtain this result by using the PTAS described above. We establish an existential result that the optimal policy for the knapsack with capacity 1 can be *folded* to get a policy with expected profit 3OPT/8 for a knapsack with capacity $(1 - \epsilon)$, with capacity relaxed to 1 *only* for the first item inserted. We then use our PTAS result to compute the $(1 + \epsilon)$-approximation to such policy.

Our techniques also yield a relaxed PTAS for *non-adaptive* policies. Finally, we also show that our ideas can be extended to yield improved approximation guarantees for
multi-dimensional and fixed set variants of the stochastic knapsack problem.

1 Introduction

The knapsack problem is a fundamental and widely studied optimization problem [GJ79, KK82]. We are given a set of items each associated with a size and a profit, and the objective is to find a maximum profit set of items with total size at most 1. When the item sizes are not fixed and are instead given by a probability distribution, the problem is broadly referred to as the *stochastic knapsack* problem. The size of an item is revealed as soon as it is inserted into the knapsack. If the knapsack capacity is not violated by the inserted item, the insertion is referred to as a *successful insertion*. No item insertions are allowed after an unsuccessful insertion, and the goal is to maximize the expected profit of successfully inserted items. The stochastic knapsack problem and its variants naturally arise in many scenarios, including bandwidth allocation, budgeted learning, and scheduling. For instance, a prototypical example is deadline-scheduling of a maximum value subset of jobs on a single machine. Even though the underlying distribution of processing times for each job is known, the actual processing time is revealed only when the job finishes execution.

Unlike the deterministic knapsack problem, the order in which the items are chosen for insertion plays a crucial role in determining the expected profit. Thus a solution to the stochastic knapsack problem corresponds to a *policy* that specifies the next item to be inserted. A *non-adaptive* policy specifies a fixed permutation of items for insertion into knapsack. An *adaptive* policy chooses the next item to be inserted based on *realized* sizes of the items inserted thus far. It is easy to see that the stochastic knapsack problem is NP-hard in both adaptive and non-adaptive settings since either setting generalizes the classical knapsack problem. We note that in the adaptive version of the problem, just the description of an optimal policy can be exponentially large. Indeed, Dean, Goemans, and Vondrák [DGV08] show that several variants of the adaptive stochastic knapsack problem are PSPACE-hard.

Much work on this problem has thus focused on

polynomial-time approximation algorithms. The current best known approximation ratios for stochastic knapsack are due to Dean et al [DGV08]; they give a $(3 + \epsilon)$-approximation to the optimal adaptive and non-adaptive policies for any $\epsilon > 0$. When the knapsack capacity is relaxed to $(1 + \epsilon)$ for some $0 < \epsilon < 1$, the techniques of [DGV08] imply a $(3 - O(\epsilon))$-approximation. There are no better approximation results known for even for commonly studied distributions such as Bernoulli or exponential, even when the knapsack capacity is relaxed to $1 + \epsilon$. More generally, we are not aware of any approximation schemes for adaptive policies for scheduling and packing problems when an unbounded number of adaptive decision points are allowed, as in the stochastic knapsack problem.

1.1 Our Results and Techniques

Stochastic Knapsack with $(1 + \epsilon)$ **Space:** Our first main result is an approximation scheme for the stochastic knapsack problem, in both adaptive and non-adaptive settings for *arbitrary* size distributions of items, provided the knapsack is allowed an arbitrarily small extra space.

THEOREM 1.1. *For any $\epsilon > 0$, there is a poly-time computable $(1 + \epsilon)$-approximate adaptive policy for the adaptive stochastic knapsack problem when knapsack capacity is relaxed to $1 + \epsilon$. A similar result holds for non-adaptive policies.*

The starting point for this result is a reduction of the given arbitrary collection of size distributions to a small number of distributions. Since an optimal policy realization may potentially insert all n items, a $(1 + \epsilon)$-approximate solution would seem to require $(1 + \epsilon/n)$-approximate representation of probabilities. This yields $\Omega(n/\epsilon)$ groups just for one size and thus $n^{\Omega(\log n/\epsilon)}$ different size distributions overall; too large for our purposes. Our first technical contribution is a sequence of transformations that discretize the given distributions into $2^{poly(1/\epsilon)} \log n$ different *canonical distributions*; the size realized by an item under a canonical distribution is referred to as its *canonical size*. Observe that an optimal adaptive policy on canonical distributions may behave completely differently from an optimal adaptive policy on real size distributions. We establish a coupling that relates events drawn from canonical size distributions to actual size realizations. Note that any discretization errors need to be analyzed over all execution paths. Towards this end, we frequently utilize the following paradigm: for each source of discretization error, we identify a *good event* (an event that fills the knapsack without error) that occurs at a much higher rate than the discretization error. Thus any profit lost due to discretization error can be charged to the good event. We also derive and use a tail bound for an adaptive sequence of random variables when both the sum of expectations of random variables in any realization and the maximum realized value of a random variable are bounded.

We next focus on computation of a near-optimal adaptive policy using canonical size distributions. It is worth noting that even for canonical size distributions, there are instances where just the description of the optimal adaptive policy is super-polynomial. We overcome this difficulty by introducing the notion of *block-adaptive* policies, that allow us to reduce the number of adaptive decision points. Broadly speaking, a block-adaptive policy makes adaptive decisions only after playing a subset of items whose profit is $\Omega(\text{poly}(\epsilon)\text{OPT})$. Our second technical contribution is to show that there exists a $(1 + O(\epsilon))$-approximate block adaptive policy when we are allowed $O(\epsilon)$ extra space. We then show that we can enumerate over all block adaptive policies in polynomial time to find a $(1 + O(\epsilon))$-approximate policy. In implementing this step, we use a technique of Chekuri and Khanna [CK00] to reduce quasi-polynomially many possibilities to polynomially many possibilities by exploiting their inter-dependence.

We note here that it is known [DGV08] that there exists an absolute constant $\epsilon_0 > 0$ such that no non-adaptive policy can approximate an adaptive policy to better than a factor of $(1 + \epsilon_0)$ even when the knapsack is allowed $(1 + \epsilon_0)$ space. Thus even with $(1 + \epsilon)$ space, design of an approximation scheme inherently requires us to explicitly consider adaptive policies.

Stochastic Knapsack without Extra Space: Our second main result is an improvement of the $(3 + \epsilon)$-approximation result of Dean et al [DGV08] when knapsack capacity is not relaxed.

THEOREM 1.2. *For any $\epsilon > 0$, there is a poly-time computable $(\frac{8}{3} + \epsilon)$-approximate adaptive policy for the adaptive stochastic knapsack problem.*

Somewhat surprisingly, we obtain this result by using Theorem 1.1. We consider the set of adaptive policies for a knapsack of capacity $1 - \epsilon$ with the following special property: if the first item inserted by the policy realizes to a size between $(1 - \epsilon)$ and 1, then the policy does not treat it as an overflow, but it can not insert any more items. We show that the optimal policy under this setting achieves an $8/3$ approximation to an optimal policy for the original knapsack capacity of 1. Given this existential result, we can use Theorem 1.1 to compute a $(1 + \epsilon)$-approximation to such a policy.

The total space used by such a policy will be at most $1 - \epsilon + \epsilon = 1$. The only modification we need to make to the dynamic program is to increase the profit of the first item placed in the knapsack to also include profit from the event where this item realizes to a size between $(1 - \epsilon)$ and 1.

The existential proof is based on a conceptual experiment where we simulate the optimal policy σ for a knapsack of capacity 1 using two knapsacks of capacity $1 - \epsilon$ by charging each item placed by the optimal policy to one of the knapsacks. We transition between knapsacks to determine which knapsack the item gets charged. The transition events between the knapsacks are dynamically determined based on the realized sizes of the inserted items. We refer to this simulation as the *two knapsack experiment*, and the output of this experiment is a pair of adaptive policies, one for each knapsack. By using suitable choice of parameters and transition points, we show that better of the two policies captures a $(\frac{3}{8})$-fraction of the optimal profit.

Multi-Dimensional Stochastic Knapsack Problem: In the multidimensional stochastic knapsack problem with d dimensions, each item is associated with a size distribution in each of d dimensions. The objective is to maximize the expected profit of items successfully inserted subject to the capacity constraint in each dimension. Dean et al [DGV05] note that this problem is a stochastic variant of the *Packing Integer Program*, namely, the problem of finding maximum value 0/1 vector satisfying $Ax \leq b$ with A and b non-negative.

The best known approximation guarantee for d-dimensional stochastic knapsack problem is a $(6d + 1)$-approximation [DGV05]. Even in the deterministic setting, this problem is known to be hard to approximate within a factor of $O(d^{1-\epsilon})$[DGV05]. Our techniques can be extended to obtain a relaxed PTAS for the multi-dimensional generalization of stochastic knapsack problem.

THEOREM 1.3. *For any fixed d, there is a PTAS for the adaptive d-dimensional stochastic knapsack problem when capacity is relaxed to $1 + \epsilon$ in each dimension. With no relaxation in capacity, there is a poly-time computable $(2d + 1 + \epsilon)$-approximate adaptive policy. A similar result holds in the non-adaptive setting.*

Fixed Set Models of Stochastic Knapsack Problem: In the *fixed set* models of the stochastic knapsack problem, the goal is to output a suitable set of items as opposed to a policy. We consider two such models.

In the *bounded overflow probability* model, the objective is to find a maximum profit subset of items such that $\Pr\left(\sum_{1 \leq i \leq |S|} X_i > 1\right) \leq \gamma$, where $X_1, X_2, ..., X_{|S|}$ are random variables corresponding to the sizes of items in S, and γ is an input parameter, referred to as the *overflow probability*. This variant is motivated by the problem of allocating bandwidth to bursty connections [KRT97]. Kleinberg et al [KRT97] gave a $\log(1/\gamma)$-factor approximation algorithms when items sizes have Bernoulli size distribution. Later, Goel et al [GI99] gave PTAS and QPTAS results for restricted classes of distributions, namely Bernoulli, Exponential and Poisson distribution. These results require the knapsack capacity and the overflow probability to be relaxed by a $(1 + \epsilon)$ factor, and the algorithm requires γ to be a constant for Bernoulli distributions. There are no results known for this problem with arbitrary size distributions, even when the knapsack capacity and the overflow probability are relaxed. We give a PTAS for arbitrary size distributions in this model.

THEOREM 1.4. *There is a PTAS for the bounded overflow probability model when the knapsack capacity and the overflow probability are relaxed by a $(1 + \epsilon)$ factor.*

In the *All-or-None* model [DGV08], we get profit for all items in the output subset S when all items in S fit in the knapsack, and otherwise we get zero profit. Dean et al [DGV08] gave a factor 9.5 approximation for this model. Our techniques can be used to get a PTAS for this model when knapsack capacity is relaxed.

THEOREM 1.5. *There is a PTAS for for the All-or-None model when knapsack capacity is increased to $1 + \epsilon$.*

1.2 Other Related Work We briefly highlight some other related work on variations of the stochastic knapsack problem and adaptivity in stochastic optimization. The stochastic knapsack problem with deterministic sizes and random profits has been studied in [CSW93], [Hen90], [Sne80], [SP79]; the objective is to find a set which maximizes the probability of achieving some threshold profit value. These works primarily focus on heuristic approaches for this problem. Derman *et al.* [DLR78] consider the *knapsack cover* problem, where multiple copies of the same item can be used and the objective is to fill the knapsack with minimum cost. They show that when item sizes follow exponential distributions, a greedy scheduling strategy solves the problem optimally.

The stochastic knapsack has also been studied in an *ordered adaptive model* where we are given a fixed sequence of items. Each item in the sequence is associated with a size distribution. For each item in the sequence, we have a choice of either inserting or neglecting the item based on previous realizations of items. The objective is to maximize the expected

profit. Dean et al [DGV08] show that optimum profit can be achieved when the capacity is relaxed to $1 + o(1)$. Halman et al [HKLOS08] show that a $(1 + \epsilon)$ approximation to the optimum can be achieved with a strict knapsack capacity constraint.

Adaptivity has been also studied in the framework of two-stage stochastic optimization (see, for instance, [IKMM04], [RS04], [Shm04]). In this setting, there is only a single level of adaptive recourse in contrast to the adaptive stochastic knapsack problem where an adaptive choice can be made after every item insertion.

1.3 Organization Section 2 presents notation, definitions, and some useful properties of $(1 + \epsilon)$-approximate policies. In Section 3, we sketch the discretization of size distributions of items; complete details are deferred to Appendix A. The transformations performed in this section allow us to restrict our search space to canonical policies that use $O(\epsilon)$ extra space. In Section 4, we show how to exploit the structure of canonical policies to get an approximation scheme for the adaptive stochastic knapsack problem. In Section 5, we show how the techniques developed in Section 3 and 4 can be used to design a $(8/3 + \epsilon)$-approximate policy under a strict capacity constraint.

In Section 6, we show that our techniques also yield an approximation scheme for non-adaptive stochastic knapsack when the capacity is relaxed to $(1 + \epsilon)$. Finally, Sections 7 and 8, describe our results for the multidimensional and the fixed set variants of the stochastic knapsack problem respectively.

2 Preliminaries

Let $\mathbb{B} = \{b_1, b_2, ..., b_n\}$ be the set of items where the profit of item b_i is denoted by p_i. We will denote by π^* the original vector of size distributions for items in $\mathbb{B}$ with size distribution of item b_i being π_i^*. We assume that the input size distributions of items are arbitrary.

We will perform a sequence of discretization steps that will transform π^* to a vector of *canonical* size distributions, that we denote by π^c. In general, given any vector of size distributions π, the size distribution of item b_i is indicated by π_i, the *expected size* μ_i of an item b_i is defined to be $\mu_i = E_{X_i \sim \pi_i}[X_i]$, and *profit density* of item b_i is defined to be $\frac{p_i}{\mu_i}$.

We use the notation $\mathbb{P}(\sigma, \pi, \mathbb{C})$ to indicate the expected profit of the policy σ with knapsack of capacity $\mathbb{C}$ and distributions on items π. We will denote by OPT the profit of an optimal policy given size distributions π^* and knapsack capacity 1. For any given $\alpha > 1$, a policy is called α-*approximate* if its expected profit is at least OPT/α. In course of performing various transformations for our algorithm, we will increase the

knapsack capacity by $O(\epsilon)$ a constant number of times. Let the final knapsack capacity be $\mathbb{C}_{max} = 1 + O(\epsilon)$; we assume w.l.o.g. that ϵ is small enough so that $\mathbb{C}_{max} \leq 2$.

Adaptive Policy and Its Representation: An adaptive policy is a function $\sigma : 2^{[n]} \times \mathbb{R} \to [n]$ that maps the set of available items and the remaining knapsack capacity to next item to be inserted into the knapsack. An adaptive policy may be equivalently viewed as a decision tree which encodes all possible execution paths in different realizations of the policy. The root node corresponds to the first item to be played by the policy, every other node corresponds to an item to be played given the path of size realizations from the root to the node. An edge from a node ν to a child node may be viewed as labeled with the size realized when the item at node ν is played. Since a path from root to any node implicitly encodes both the available items and the residual knapsack capacity, each node in the decision tree needs to be only labeled with an item to be played. A leaf in the decision indicates the end of the policy, and sum of sizes of items on any root leaf path (other than potentially the last item) is no more than the knapsack capacity.

Structured Near-Optimal Policies: We sketch here a few simple transformations that allow us to restrict our attention to near-optimal policies with certain useful structural properties. The first transformation below allows us to bound the maximum possible size of any item. Given size distribution π_i^* of an item b_i, its truncated distribution $\pi_i^{'}$ is:

$$\Pr_{X_i^{'} \sim \pi_i^{'}} (X_i^{'} = s) = \Pr_{X_i \sim \pi_i^*} (X_i = s), \quad \forall s \leq 1 \quad \text{and}$$

$$\Pr_{X_i^{'} \sim \pi_i^{'}} (X_i^{'} = \mathbb{C}_{max} + \epsilon) = \Pr_{X_i \sim \pi_i^*} (X_i > 1)$$

Clearly, the truncation does not affect the expected profit of the optimal policy. With the truncated size distribution, the maximum size to which an item can realize is $\mathbb{C}_{max} + \epsilon$. *Here onwards, we thus assume truncated size distributions.* The lemma below holds for truncated size distributions and its proof is similar to the Lemma 3.1 in [DGV08].

LEMMA 2.1. *For a stochastic knapsack of capacity $\mathbb{C}$ where $1 \leq C \leq C_{max}$ and any adaptive policy, let S denote the (random) set of items the policy attempts to insert. Then $E_S[\sum_{b_i \in S} \mu_i] \leq \mathbb{C} + \mathbb{C}_{max} + \epsilon = O(1)$.*

Next we show that item profits can be assumed to be bounded by OPT/ϵ. We define an item to be *a huge profit item* if it has profit greater than or equal to OPT/ϵ. Given the initial vector of size distributions π^* on items, for any huge profit item b_i, we define a new

size distribution π_i' as follows:

$$\Pr_{X_i' \sim \pi_i'}(X_i' = s) = \left(\Pr_{X_i \sim \pi_i^*}(X_i = s)\frac{\epsilon p_i}{\text{OPT}}\right), \quad \forall s \leq 1$$

and

$$\Pr_{X_i' \sim \pi_i'}(X_i' = \mathbb{C}_{max} + \epsilon) =$$
$$1 - \sum_{s \leq 1}\left(\Pr_{X_i \sim \pi_i^*}(X_i = s)\frac{\epsilon p_i}{\text{OPT}}\right)$$

We now *scale the profit* of the item b_i by $\text{OPT}/(\epsilon p_i)$; thus the profit of b_i is OPT/ϵ after scaling. In Lemma 2.2 and 2.3, we show that that this transformation can be performed with only an $O(\epsilon)$ loss in the optimal profit.

LEMMA 2.2. *There is a $(1+\epsilon)$-approximate policy that plays at most one huge profit item in any realization, and that also always at the end of the policy.*

Proof. Let E be the event where an optimal policy successfully inserts a huge profit item. Clearly, $\Pr(E) \leq \epsilon$. If S is the (random) set of items which an optimal policy attempts to inserts after successfully inserting the first huge item, then the expected profit contributed by items in S conditioned on the event E is at most OPT, otherwise policy is not optimal. Hence the expected profit contributed by items in S is at most $\Pr(E) \cdot \text{OPT} = \epsilon\text{OPT}$. □

LEMMA 2.3. *Restricted to the set of policies that play at most one huge profit item in any realization and that also at the end, the expected profit before and after the profit and size scaling operation for huge profit items is the same.*

Proof. Conditioned on the event that the policy does not attempt to insert any huge item, the expected profit of the policy remains unaffected by scaling. We will analyze the profit of the policy conditioned on the event that the policy attempts to insert b_i at the end, where b_i is a huge profit item. The proof follows if we show that the claim holds true for every choice of b_i.

With unscaled size distributions on items, let $I(X_i, c)$ be the indicator variable of the event that before b_i is inserted, amount of space left in the knapsack is c and b_i realizes to size $X_i \leq c$. Similarly we define $I'(X_i, c)$ for the scaled distributions. Since there is no other huge item which the policy has already inserted into the knapsack and there is no change in the distributions of the non-huge items, we get,

$$E[I(X_i, c)] = \frac{\text{OPT} \cdot E[I'(X_i, c)]}{\epsilon p_i}$$

With unscaled distributions on items, the expected profit contributed by b_i conditioned on the event that b_i is the huge profit item that policy has attempted to insert, is

$$p_i E[I(X_i, c)].$$

With scaled distributions, the expected profit contributed by b_i conditioned on the event that b_i is the huge profit item that the policy has attempted to insert, is

$$\frac{\text{OPT} \cdot E[I'(X_i, c)]}{\epsilon} = p_i E[I(X_i, c)].$$

This completes the proof. □

Here onwards, all references to π^* will refer to the distributions with the above two properties. The preceding transformations allow us to establish the useful lemma below.

LEMMA 2.4. *Given a size distribution vector π^* and knapsack capacity 1, there is a $(1 + O(\epsilon))$-approximate policy σ that satisfies the following properties: (1) in any realization, σ achieves profit $O(\text{OPT}/\epsilon)$, (2) in any realization, the sum of expected sizes of items that σ inserts is $O(1/\epsilon)$, and moreover, (3) σ never plays an item with profit density less than ϵOPT.*

Proof. To prove the first property, let E_1 be the event where the policy successfully inserts items with profit $O(\text{OPT}/\epsilon)$. Clearly $\Pr(E_1) \leq \epsilon$. Let S_1 be the (random) set of items which an optimal policy attempts to inserts after successfully inserting items with total profit OPT/ϵ. Then the expected profit contributed by items in S_1 conditioned on the event E_1 is at most OPT (otherwise the original policy is not optimal). Hence expected profit contributed by items in S_1 is at most $\Pr(E_1) \cdot \text{OPT} = \epsilon\text{OPT}$, which is the profit lost if the policy stops in the event E_1.

To prove the second property, let E_2 be the event where the policy successfully inserts items with the sum of expected sizes $\Omega(1/\epsilon)$. Clearly, $\Pr(E_2) \leq O(\epsilon)$ by Lemma 2.1. Let S_2 be the (random) set of items which the optimal policy attempts to insert, after successfully inserting items with sum of expected sizes $3/\epsilon$. Then the expected profit contributed by items in S_2 conditioned on the event E_2 is at most OPT (otherwise the original policy is not optimal). Hence expected profit contributed by items in S_2 is at most $\Pr(E_2) \cdot \text{OPT} = O(\epsilon\text{OPT})$. Now, there is one more claim that we have to make to complete the proof. The maximum size to which an item can realize is $O(1)$, and hence the very last item that gets inserted does not violate the guarantee in the lemma.

Using Lemma 2.1, for any policy, the expected profit contributed by items with profit density less than ϵOPT is $O(\epsilon\text{OPT})$, this proves the third property in the lemma. $\qquad\square$

Lemma 2.4 is crucial to the analysis of the discretization step, where the hard upper bounds on the maximum possible profit in any realization, and on the sum of expected sizes of items any policy attempts to insert in any realization are repeatedly used to bound the loss in expected profit because of error events.

We note that even though some steps in our algorithm, e.g. the scaling operation on huge profit items, assume knowledge of OPT, their correctness does not rely on exact knowledge of OPT. In particular, it suffices to use the $(3 + \epsilon)$-approximate estimate of OPT given by [DGV08]. In the final approximation ratio of $(1 + O(\epsilon))$, this affects the multiplier of ϵ by a constant factor.

3 Discretization of Distributions: A Sketch

We perform a number of transformations to discretize the size distributions of items, while ensuring that there remains a $(1 + \epsilon)$-approximate adaptive policy that uses only $O(\epsilon)$ extra space. We provide here a high-level sketch, with a view towards explaining how these transformations relate to each other and to the rest of the algorithm, and to highlight the novel elements in our discretization scheme. Details of these transformations and full proofs are deferred to Appendix A.

We first divide the size distribution of each item into two regions, a *large size region* where sizes $\geq \epsilon^5$ and a *small size region* which corresponds to sizes $< \epsilon^5$. With reference to size distribution π_i for an item b_i, we will use notation $\pi_i(large)$, $\pi_i(small)$, $\mu_i(small)$ to indicate the probability that an item b_i realizes to a large size, probability that b_i realizes to a small size, and the expected size of b_i conditioned on the event that b_i realizes to a small size respectively. For any size $s \geq 0$, we will use notation $\pi_i(s)$ to indicate $\Pr_{X_i \sim \pi_i}(X_i = s)$. Also, given two size distributions π_i and π_i' for an item b_i, we will use notation $\Delta(\pi_i, \pi_i')$ to denote their total *variation distance*, i.e. $\Delta(\pi_i, \pi_i') = \sum_s |\pi_i(s) - \pi_i'(s)|$.

We now describe the discretization steps. At the end of the discretization, we reduce the possible size distributions to $2^{poly(1/\epsilon)} \log n$ while ensuring that there is still a $(1 + O(\epsilon))$-approximate adaptive policy when knapsack capacity is relaxed to $(1 + O(\epsilon))$. Proofs of these steps crucially depends the fact that policy in consideration satisfies the properties given in Lemma 2.4.

Step 1: Discretizing large sizes: Since the size of any item is $O(1)$ after truncation, the range of large sizes is $[\epsilon^5, O(1)]$. We discretize these sizes to powers of $(1+\epsilon)$, resulting in $q' = poly(1/\epsilon)$ sizes and in no loss of profit for an optimal policy if we increase the knapsack capacity by a factor of $(1 + \epsilon)$.

Step 2: Coupled discretization of large-size probabilities: This step is quite involved, and shows that not just the sizes of the items, but the *entire large-size distribution* can be discretized into a small number of classes. Observe that each individual probability value $\pi_i(s)$ can legitimately lie in the range $[\Theta(\epsilon^2/n), O(1)]$ as the policy can potentially insert n items.

We use a coupling argument to show that for any change in the distributions of items in the large size region such that variation distance for any item b_i is $O(\epsilon^8 \pi_i(large))$, the expected profit reduces by only $O(\epsilon\text{OPT})$. Thus, we can discretize the probability values for a single size to the powers of $(1 + poly(\epsilon))$. If we store discretized individual probability values, we would get $\Omega((\log n)^{poly(1/\epsilon)})$ different size distributions. To get around this problem, we represent $\pi_i(s)$ for any large size s as being *relative* to $\pi_i(large)$, so that it has only $O(poly(1/\epsilon))$ values. We show that this can be achieved while keeping the total variation distance between the original and new distributions small.

After the first two steps, the number of possible large-size distributions is as desired, i.e., $2^{poly(1/\epsilon)} \log n$. We next show that the number of size-distributions can be similarly bounded even if we take the small sizes into account. For this, we will show (in steps 3 and 4) that for any item, either we need to store only the large-size distribution, or only the small-size distribution, or $\mu_i(small)$ is within a $poly(1/\epsilon)$ factor of $\pi_i(large)$.

Step 3: Upper-bounding $\mu_i(small)$. We show that if $\mu_i(small)$ is larger than $2\pi_i(large)/\epsilon^3$ then for such items, the small size distribution stochastically dominates the large size distribution and we can ignore the large size distribution while reducing the optimal profit by only $O(\epsilon\text{OPT})$.

Step 4: Lower-bounding $\mu_i(small)$. We show that if $\mu_i(small)$ is smaller than $\epsilon^9 \pi_i(large)$, then by relaxing the knapsack capacity by ϵ, we can ignore the small-size distribution of such items while reducing the optimal profit by only $O(\epsilon\text{OPT})$.

Step 5: Dealing with the small-size distribution. This is the most critical step in the discretization process since we are dealing with adaptive policies. We replace the small size distribution of an item b_i by a single size $\mu_i(small)$ and the corresponding probability $\pi_i(small)$. Let π and π' be the vectors of size distributions before and after the discretization change in this step. Observe that, this is the only step where the real size of an item and the corresponding discretized size

are not within $[1/(1 + \epsilon), 1]$ times of each other and we need to show the existence of an adaptive policy σ^c such that $\mathbb{P}(\sigma^c, \pi', 1 + O(\epsilon)) \geq (1 - O(\epsilon))\text{OPT}$. Furthermore, we also need to analyze the performance of such a policy on items from π.

To begin with, we note that the *canonical size* associated with a size realization s of an item b_i is (a) $\mu_i(small)$, if s is in the small size region, and (b) the discretized size in large size region in the range $(s/(1 + \epsilon), s)$, if s is in the large size region.

We now introduce the notion of a *canonical policy* to handle the discretized item sizes. Given a knapsack of capacity $(1 + O(\epsilon))\mathbb{C}$, a policy is said to be a canonical policy if (a) for any item b_i inserted into the knapsack, the policy observes its canonical size; and (b) the policy stops if either the total canonical size of items inserted exceeds $\mathbb{C}$ or the total real size exceeds $(1 + O(\epsilon))\mathbb{C}$.

To summarize, for an adaptive canonical policy, the decisions are purely based on observed canonical sizes. At the end of the policy, it observes the real sizes of all items inserted and if the difference between the sum of real and canonical sizes of items inserted is more than $O(\epsilon)$, it discards the entire profit of all items inserted. We now show two important properties of the canonical policies.

LEMMA 3.1. *There exists an adaptive canonical policy σ^c such that $\mathbb{P}(\sigma^c, \pi', 1 + O(\epsilon)) \geq (1 - O(\epsilon))\text{OPT}$. Moreover, for any adaptive canonical policy σ^c, $\mathbb{P}(\sigma^c, \pi, (1 + O(\epsilon))\mathbb{C}) \geq \mathbb{P}(\sigma^c, \pi', \mathbb{C}) - O(\epsilon)\text{OPT}$ where $\mathbb{C} = 1 + O(\epsilon)$.*

We briefly outline the proof. The detailed proof can be found in Appendix A.5. We first consider the existence of a canonical policy σ^c for items in π' whose expected profit is $(1 - O(\epsilon))\text{OPT}$. Let $\mathbb{T}$ be the decision tree of the optimal policy for items with size distributions π (for the knapsack of capacity 1). Each root-leaf path in $\mathbb{T}$ is associated with a (fixed) set of items and their (fixed) realized sizes. Thus for a given leaf in $\mathbb{T}$, the difference between the sum of realized and expected sizes (in the small size region) of items realizing to small size is fixed quantity. We call a leaf to be a *good leaf* if this difference is less than $O(\epsilon)$, and a *bad leaf otherwise*. Given distribution π, each leaf is associated with a probability of reaching that leaf, that we refer to as its *weight*. We show that at least $(1 - O(\epsilon^2))$-fraction of the leaves (by weight) are good leaves. Note that, standard tail bounds like Chernoff's bound or Azuma's inequality do not suffice to establish this, due to adaptive nature of the policy.

This alone does not imply the existence of $(1 + O(\epsilon))$-approximate policy for items with size distributions π' with $O(\epsilon)$ relaxation in then knapsack capacity.

We then show that $\mathbb{T}$ can be transformed to create a policy (i.e. the corresponding decision tree) for items in π' such that with extra $O(\epsilon)$ space, the expected profit of the policy is $(1 - O(\epsilon))\text{OPT}$. This proves the first claim in the lemma.

The second part of the lemma follows by decoupling the discretized and real distributions and use of standard tail bounds. Note that, in contrast, for non-adaptive policies, the proofs of both claims in Lemma 3.1 immediately follow by a careful application of Chernoff's bound.

Number of Size Distributions: Now we count the number of distinct size distributions. First we observe that when we have to remember both the small-size and large-size distributions for an item, $\pi_i(small)$ is inherent in the large-size distribution (since $\pi_i(small) = 1 - \pi_i(large)$) and need not be explicitly remembered. Next, we observe that when we are storing the small-size distribution alone, $\pi_i(small)$ is implicitly 1, and need not be explicitly remembered. Similarly, when we are remembering both distributions, $\mu_i(small)$ is within $poly(1/\epsilon)$ of $\pi_i(large)$ and increases the number of possible distributions by only $poly(1/\epsilon)$ when appropriately discretized. When we are only storing the small-size distribution, $\mu_i(small)$ can be easily discretized into $O(poly(1/\epsilon) \log n)$ values. Thus, we can discretize the entire large-size distribution as well as the values $\pi_i(small)$ and $\mu_i(small)$ into a total of $2^{poly(1/\epsilon)} \log n$ different values.

LEMMA 3.2. *Total number of different size distributions is $2^{poly(1/\epsilon)} \log n$.*

4 An Approximation Scheme for Adaptive Stochastic Knapsack Problem

By the discretization scheme of Section 3, we know that there exists an adaptive canonical policy σ^c with expected profit $(1 - O(\epsilon))\text{OPT}$ that uses $\mathbb{C} = (1 + O(\epsilon))$ space and satisfies the constraints of Lemma 2.4. We aim to guess the decision tree of such policy by enumerating over decision trees of all canonical policies. This cannot be done in polynomial time as even with canonical distributions, just the description of a single canonical policy i.e. the its decision tree is of super-polynomial size. Hence we restrict to the set of canonical policies with weak adaptivity, where the policies insert a *block* of items with total profit $\Omega(poly(\epsilon)\text{OPT})$ together and take the decision based on the sum of canonical size realization of all items in the *block*. We refer to such policies as *block-adaptive policies*. In Lemma 4.1, we show that to find a policy with expected profit $(1 - O(\epsilon))\text{OPT}$, it is sufficient to restrict the enumeration of all policies to the set of block-adaptive policies. We then show

that the set of all block-adaptive can be enumerated in polynomial time. The best policy found by this process is guaranteed to be a $(1 + O(\epsilon))$-approximation to the adaptive optimum.

LEMMA 4.1. *An optimal canonical adaptive policy σ^c can be transformed into a block-adaptive policy with expected profit $(1 - O(\epsilon))\mathrm{OPT}$ when the capacity constraint is further relaxed by $\epsilon\mathbb{C}$.*

In the remainder of section, we prove Lemma 4.1. We will denote by $\widehat{\sigma}$ the block-adaptive policy obtained by suitably transforming σ^c. We highlight two keys issues that arise in transforming σ^c to $\widehat{\sigma}$. One, since the decision tree of σ^c can have an arbitrarily complex structure, at any node in the decision tree, the remainder of the policy may be completely different based on two different size realizations of the corresponding item. Thus if we inserting a block of items together, the policy $\widehat{\sigma}$ may end up inserting items that σ^c would not have inserted for some size realizations for items in the block. Second, we need to identify a block of items to be played together even when decision branches emanating from a node in σ^c might insert distinctly different sets of items.

We now describe a transformation of σ^c into a $\widehat{\sigma}$ that addresses both issues above using $\epsilon\mathbb{C}$ space. For the purpose of analysis, it will be convenient to express $\widehat{\sigma}$ as a policy that uses two knapsacks, a *main knapsack* of capacity $\mathbb{C}$ and an *auxiliary knapsack* of capacity $\epsilon\mathbb{C}$. Clearly, $\widehat{\sigma}$ can be implemented in a single knapsack of capacity $(1 + \epsilon)\mathbb{C}$. The core step in our transformation is the idea of tiling the decision tree of the policy with *segments*. A path in the decision tree is called a *small realization path* if either every edge on the path corresponds to a small realization or the path contains a single node. Our starting point is a partition of nodes in σ^c into small realization paths, called *segments* such that if a segment has more than one node, the profit of the segment (sum of profits of items on it) is at most $2\epsilon^{14}\mathrm{OPT}$, and otherwise, the single node in the segment satisfies one of the following conditions: (i) it either has a profit is at least $\epsilon^{14}\mathrm{OPT}$, or (ii) it does not a child node corresponding to a small size realization. It is easy to see that such a partition always exists. At a high level, $\widehat{\sigma}$ inserts all items in a single segment of σ^c together. To compare realizations of σ^c and $\widehat{\sigma}$, if all items in a segment realize to a small size, then items inserted in $\widehat{\sigma}$ match the items inserted in σ^c, and when any item in a segment realize to large size, then $\widehat{\sigma}$ may have inserted extra items which σ^c would not have inserted. Thus in a given realization of σ^c, we need to account for extra space required by the items over all segments in which there is a large size realization. It is interesting to note that, such transformation can

be done for canonical policies as for any node their decision tree, the remainder of the policy is uniquely defined when the corresponding item realizes to a small size. It is not clear how to apply such transformation to adaptive policies which look at real sizes of items. The transformation is explained in detail in Algorithm 4.1.

Algorithm 4.1 Policy $\widehat{\sigma}$

1. Initially, $S = \phi$. S represents the set of items added to the auxiliary knapsack.
2. Set $\sigma = \sigma^c$, and compute the segments on the decision tree of σ.
repeat
   3.1 Follow the policy σ by continually adding items to main knapsack until at some node ν, the item corresponding to ν, say b_ν, realizes to a large size, say s.
   3.2 If any items remain in the segment containing ν, add these items, say S', to set S. Add items in S' to auxiliary knapsack and the profit of items in S' to the profit collected.
   3.3 Let σ' be the remainder of policy in σ corresponding to b_ν realizing to (large) size s. Note that items in S' can be present in σ'. For each node ν' in σ' which corresponds to an item in S', replace the subtree rooted ν' by the subtree rooted at the child node of ν' corresponding to small size realization of the item at ν'.
until the policy σ is over.
4. If the auxiliary knapsack overflows, then discard the entire profit of the realization.

Observe that $\widehat{\sigma}$ inserts items in one segment of σ together. The lemma below establishes useful properties of $\widehat{\sigma}$ and lower bounds the expected profit of $\widehat{\sigma}$.

LEMMA 4.2. *In any realization of $\widehat{\sigma}$, the sum of profit of items successfully inserted is $O(\mathrm{OPT}/\epsilon)$; the sum of expected sizes of items in S at the end of the policy is $O(\epsilon^8)$; the sum of expected sizes of items is $O(1/\epsilon)$; and the expected profit of $\widehat{\sigma}$ is $(1 - O(\epsilon^2))\mathrm{OPT}$.*

Proof. The maximum profit in any realization of $\widehat{\sigma}$ is bounded by the maximum profit in any realization of σ^c plus total profit of items in S. The former is bounded by $O(\mathrm{OPT}/\epsilon)$. The total profit of items in S is bounded by $O(\epsilon^9\mathrm{OPT})$, since there are $O(1/\epsilon^5)$ realizations of items to a large size before the main knapsack overflows and with each large realization, items added to S have total profit $O(\epsilon^{14}\mathrm{OPT})$.

The sum of expected sizes of items in S is bounded by $O(\epsilon^8)$, since there are $O(1/\epsilon^5)$ realizations of items to a large size before the main knapsack overflows and S

does not contain any item with profit density less than ϵOPT.

The sum of expected sizes of items $\widehat{\sigma}$ attempts to insert in any realization is bounded by the sum of expected sizes of items in any realization of σ^c plus sum of expected sizes of items in S. The former is bounded by $O(1/\epsilon)$ while the later is bounded by $O(\epsilon^8)$ as argued above.

To prove the fourth claim, we start by noting that if we do not terminate the policy $\widehat{\sigma}$ due to either a large size realization for an item added to auxiliary knapsack or an overflow of the auxiliary knapsack, then the expected profit of $\widehat{\sigma}$ is at least the expected profit of σ^c. Hence to compare the expected profits of $\widehat{\sigma}$ and σ^c, it suffices to analyze the probability of the event that either in some segment two or more items realize to large size, or the auxiliary knapsack overflows before the main knapsack. Note that since for any item, the *canonical* size taken in a small realization is a fixed value (its expected size conditioned on being small), the auxiliary knapsack overflows only if some item added to it takes a large size. The sum of expected sizes of items in S in any realization is $O(\epsilon^8)$. Hence by Markov's inequality, the probability that in any realization some item in S realizes to a large size is $O(\epsilon^3)$. It follows that the expected profit of $\widehat{\sigma}$ is $(1 - O(\epsilon^2))$OPT by Lemma 4.2. $\square$

We now list important structural properties of the decision tree of $\widehat{\sigma}$.

1. The policy is decomposed into segments such that each segment is a path in the tree. A segment with one item has profit at least ϵ^{14}OPT, and for a segment with multiple items, sum of profits is at most $2(\epsilon^{14}$OPT$)$. Each node in a segment has exactly one child node in the decision tree, with the exception of the last node in a segment that can have $poly(1/\epsilon)$ children.

2. For any two consecutive segments, say s' and s'' where on a root-leaf path (other than last segment on the path) in the decision tree, either (a) the sum of the profits of s' and s'' is at least ϵ^{14}OPT, or (b) s'' corresponds to one of the possible next segments in the remainder of the policy to be followed when at least one of the item in s' realize to a large size.

3. When at least one item in a segment realizes to large size, the remainder of the policy is decided by the first item in the segment that realizes to large size along with the size to which it realizes. Since in lower bounding the profit of $\widehat{\sigma}$ we discarded the profit from realizations where more than one item realizes to a large size in a segment, the expected

profit of $\widehat{\sigma}$ is at least $(1-O(\epsilon))$OPT even if items are arbitrarily permuted within each segment of $\widehat{\sigma}$. For the same reason, the expected profit of the policy, where remainder of the policy for any segment is chosen based on total canonical size used by items in the segment even when there are multiple large size realizations, is no worse.

Now we establish the bound on number of segments in the decision tree of $\widehat{\sigma}$.

LEMMA 4.3. *Any root-leaf path in the decision tree of $\widehat{\sigma}$ has $O(1/\epsilon^{15})$ segments. The number of children the last node of any segment can have is $poly(1/\epsilon)$. There are $2^{poly(1/\epsilon)}$ segments in the decision tree of $\widehat{\sigma}$.*

Proof. The first property follows from the fact that the sum of profit of items on any path from root to leaf is $O(\text{OPT}/\epsilon)$ and each pair of consecutive segments either contributes a profit of $O(\epsilon^{14}$OPT$)$, or represents the event that some item realized to a large size. The number of pairs of the former type is bounded by $O(1/\epsilon^{15})$ while the number of pairs of the latter type can not exceed $O(1/\epsilon^5)$.

For the second property, note that in the policy tree of $\widehat{\sigma}$, for each segment s, there is a fixed child segment which corresponds to the first segment in the remainder of the policy to be followed when all items in s realize to small size. All remaining child segments of s correspond to first segment in the remainder of the policy when at least one item in s' realizes to a large size, and are simply labeled with a distinct large size value. Thus the number of child segments for any segment can be bounded by $poly(1/\epsilon)$.

The third property is an immediate corollary of first two claims. $\square$

Thus any such policy has $f_1(\epsilon) = 2^{\text{poly}(1/\epsilon)}$ segments. Now we enumerate over all block-adaptive policies and select the policy with the maximum expected profit. The PTAS follows from the lemma below.

LEMMA 4.4. *The set of block-adaptive canonical policies can be enumerated in polynomial time.*

Proof. To enumerate all policies, we will explicitly enumerate in top down manner choices of items for each segment. We will show that there are $n^{f_2(\epsilon)}$ choices for each segment, where $f_2(\epsilon) = 2^{\text{poly}(1/\epsilon)}$, yielding an algorithm with running time $n^{f_2(\epsilon)^{f_1(\epsilon)}}$.

We classify items based on their discretized size distribution. The items belonging to same *size class* have same discretized distribution. We can assume w.l.o.g. that for any pair of items b_i and b_j same size class, $\widehat{\sigma}$ always inserts an item with higher profit first.

Thus items can be assumed to be arranged in non-increasing order of profit in each size class. We have shown that for the policies in consideration, items can be permuted within a segment.

We are now ready to describe the enumeration process for a given segment. For any given segment, there are two possible choices, either it has only one item or it has at least two items. Total number of choices for a segment when the segment has only one item is n. We now focus on the case when there are at least two items in the segment; total profit of items in any such segment is bounded by $2\epsilon^{14}$OPT. It is sufficient to show that number of choices for each segment is $n^{2^{\text{poly}(1/\epsilon)}}$.

This technique is similar to one used in [CK00]. For any segment which has profit less than $2\epsilon^{14}$OPT, we assume that remaining profit is coming from a dummy class of items. The items in the dummy class do not contribute any profit while computing the expected profit of the policy and they have zero size. For any size class j, let P_j^i be the profit contributed by the items in size class j to the current segment. For each size class j, we define $\alpha_j = \lfloor \frac{2qP_j^i}{\epsilon^{16}\text{OPT}} \rfloor$. We only consider classes j with $\alpha_j \geq 1$. Total loss in profit by this step because of floor operation is at most ϵ^{16}OPT per segment. By Lemma 4.3, this will reduce the profit by $O(\epsilon\text{OPT})$.

Now the problem is to distribute $\frac{2q}{\epsilon^2}$ among q classes. Number of ways of doing this is at most $\binom{2q/\epsilon^2+q}{q}$ which is at most

$$\frac{(2q/\epsilon^2 + q)^q}{(q/e)^q} \leq (3e/\epsilon^2)^q \leq (1/\epsilon^2)^{\frac{O(\log n)}{\epsilon^{\text{poly}(1/\epsilon)}}} = n^{2^{\text{poly}(1/\epsilon)}}$$

Now, for every size class j, we allocate items to the given segment in the decreasing order of profits until the sum of profits is more than $\alpha_j \epsilon^{17}\text{OPT}/2q$. There is one more issue that we have to consider here. Since the weight of the items from the class may not exactly add up to profit allocated to the class, we consider both allocations, first being maximum profit possible such that it is not more than allocated profit and lowest profit possible which is not less than allocated profit. This adds a multiplicative factor of $2^q = n^{2^{poly(1/\epsilon)}}$ to the number of combinations that need to be tried per segment. Thus the lemma follows. □

5 An $(\frac{8}{3} + \epsilon)$-Approximation for Adaptive Stochastic Knapsack without Extra Space

In this section, we give a polynomial-time algorithm to compute an $(\frac{8}{3} + \epsilon)$-approximate adaptive policy when there is no relaxation in the knapsack capacity. Our result is based on the following lemma.

LEMMA 5.1. *There exists an adaptive policy with expected profit* (3OPT)/8 *which uses a knapsack of capac-*

ity $(1 - \epsilon)$ *with the following exception: if the very first item inserted into the knapsack realizes to a size between* $(1 - \epsilon)$ *and 1, then this insertion is still considered a success, and the policy terminates.*

Given the lemma above, we can use Theorem 1.1 to compute a canonical policy which is a $(1 + \epsilon)$-approximation to such a policy; the only modification is that the dynamic program adds $(p_i \cdot \text{Pr}_{X_i \sim \pi_i^*}(1 - \epsilon < X_i \leq 1))$ to the profit of the very first item b_i it inserts into the knapsack. The total real knapsack capacity required for this canonical policy to compensate for discretization will be $1 - \epsilon + \epsilon = 1$ and hence, Theorem 1.2 follows. In the remainder of the section, we prove Lemma 5.1.

Recall that π^* is the vector of original size distributions on items. Let σ^* be the optimal adaptive policy for the knapsack of capacity 1, thus $\mathbb{P}(\sigma^*, \pi^*, 1) = \text{OPT}$. For any item b_i, we define its *scaled profit*[1], denoted by p_i^{sc}, as $(p_i \text{Pr}_{X_i \sim \pi_i^*}(X_i \leq 1))$. Thus, maximum contribution by b_i towards the expected profit of any policy for a knapsack of capacity 1 is p_i^{sc}. We now explain the *two knapsack experiment* in Lemma 5.2.

LEMMA 5.2. *Given any* $\mathbb{C} \leq 1 - \epsilon$ *such that* $2\mathbb{C} \geq 1$, *and any adaptive policy* σ *such that the maximum scaled profit of any item played by* σ *is* p_{max}^{sc}, *there exists a policy* σ' *such that*

$$\mathbb{P}(\sigma', \pi^*, \mathbb{C}) \geq \frac{\mathbb{P}(\sigma, \pi^*, 1) - p_{max}^{sc}}{2}$$

Proof. We simulate realizations of σ for a knapsack of capacity 1 in two knapsacks, each of capacity $\mathbb{C}$. We follow the decision tree associated with σ for the knapsack of capacity 1 assigning each item dynamically to one of the two knapsacks. Here onwards, any reference to σ will be to the associated decision tree for the knapsack of capacity 1.

We start adding items according to σ to the first knapsack, until the sum of sizes of inserted items is at least $(1 - \mathbb{C})$. Then we start adding the items to second knapsack until the end of the policy σ. Note that, $(1 - \mathbb{C}) + \mathbb{C} = 1$ and $2C \geq 1$. Thus, the two knapsacks together have enough capacity to accommodate all items in any realization of σ for a knapsack of capacity 1. We now want to compare the expected profit of the items inserted into the first and second knapsack to $\mathbb{P}(\sigma, \pi^*, 1)$.

In any realization of σ, all items which contribute towards the profit of σ also contribute towards the profit of one of the two knapsacks, with a possible exception

---

[1]To clarify, this scaling operation is different than the scaling operation on huge profit items, mentioned in Section 2.

of the last item added to the first knapsack. The last item added to the first knapsack may cause an overflow in the first knapsack which is not an overflow in σ. Thus the expected profit lost in the two knapsack experiment is p_{max}^{sc}. Hence, for at least one of the two knapsacks, the expected profit is at least $\frac{\mathbb{P}(\sigma, \pi^*, 1) - p_{max}^{sc}}{2}$. $\qquad\square$

If the given set of items contains any item with scaled profit more than $3\mathrm{OPT}/8$, then we can simply play that item in the knapsack of capacity 1 and achieve the required profit. W.l.o.g. assume that the scaled profit of any item is at most $3\mathrm{OPT}/8$. We classify items into three types based on profits and size distributions.

We say an item b_i is of *type T_1* if $p_i^{sc} < \mathrm{OPT}/4$; it is of *type T_2* if $p_i^{sc} \geq \mathrm{OPT}/4$ and $\mathrm{Pr}_{X_i \sim \pi_i^*}(X_i < \epsilon) \geq 2/3$; and finally, it is of type T_3 if $p_i^{sc} \geq \mathrm{OPT}/4$ and $\mathrm{Pr}_{X_i \sim \pi_i^*}(X_i < \epsilon) < 2/3$. Now we consider two cases based on whether or not the given set of items contains an item of type T_2. We will prove Lemma 5.1 in each case by showing the existence of the desired policy.

Case 1: *There is at least one item of type T_2, say b_i, in the given set of items.* Clearly, there exists a policy σ_1 which never plays b_i and $\mathbb{P}(\sigma_1, \pi^*, 1) \geq \mathrm{OPT} - p_i^{sc}$. Now using Lemma 5.2 with $\mathbb{C} = 1 - 2\epsilon$, there exists a policy σ_2 such that

$$\mathbb{P}(\sigma_2, \pi^*, 1 - 2\epsilon) \geq \frac{\mathbb{P}(\sigma_1, \pi^*, 1) - 3\mathrm{OPT}/8}{2}$$
$$\geq \frac{\mathrm{OPT} - p_i^{sc} - 3\mathrm{OPT}/8}{2}$$

We now design a policy with expected profit at least $(3\mathrm{OPT})/8$. The policy starts by playing item b_i. If item b_i realizes to a size less than ϵ, then we add items according to σ_2 for a knapsack of capacity $1 - 2\epsilon$. Now we compute the expected profit of such a policy. We get profit p_i^{sc} from item b_i (recall that the first item can realize to size up to 1), and when b_i realizes to a size of at most ϵ, we get an additional profit of $\mathbb{P}(\sigma_2, \pi^*, 1 - 2\epsilon)$. Thus using the fact that $p_i^{sc} \geq \mathrm{OPT}/4$, we get the expected profit of the policy is

$$p_i^{sc} + \frac{2}{3}\left(\mathbb{P}(\sigma_2, \pi^*, 1 - 2\epsilon)\right)$$
$$\geq p_i^{sc} + \frac{\mathrm{OPT} - p_i^{sc} - 3\mathrm{OPT}/8}{3} \geq \frac{3\mathrm{OPT}}{8}$$

The space used by the policy when it attempts to insert two or more items is no more than $\epsilon + 1 - 2\epsilon = 1 - \epsilon$.

Case 2: *There is no item of type T_2 in the given set of items.* We perform a modified version of the two knapsack experiment to simulate realizations of σ^* for a knapsack of capacity 1 in two knapsacks, each of capacity $1 - \epsilon$.

Step (1): Start inserting items to the first knapsack according to σ^* until the total size of items in the first knapsack is at least ϵ, or the next item to be inserted is of type T_3. At this point we switch to the second knapsack and follow Step (2a) if it is applicable. If not, then we follow Step (2b).

Step (2a): If we switched to the second knapsack because the total size of items in the first knapsack is at least ϵ, then we insert items into the second knapsack until the end of the policy.

Step (2b): If we switched to the second knapsack because the next item to be inserted according to σ^* is a type T_3 item, then we continue inserting items into the second knapsack until the total size of items in the second knapsack is at least ϵ. Then we switch back to the first knapsack and insert items into the first knapsack until the end of the policy. While adding the first item in Step (2b), we assume the knapsack capacity to be 1, not $1 - \epsilon$. After adding the first item, we treat knapsack capacity to be $1 - \epsilon$.

Thus in each of the two knapsacks, the space used by the policy is $1 - \epsilon$, with the possible exception for the first item inserted into the second knapsack in Step (2b). Note that if the first item realizes to a size more than ϵ, then then we switch back to the first knapsack and it is the only item which is inserted into the second knapsack. Now we establish the lower bound on the expected profit for the better of two knapsacks in Lemma 5.3.

LEMMA 5.3. *At least one of the two knapsacks achieves an expected profit of $(3\mathrm{OPT})/8$ or more.*

Proof. We define two disjoint events in the realization of the policy σ^* for a knapsack of capacity 1.

1. We define an event E_1 when there is an overflow in the first knapsack in step 1. This is always associated with an item of type T_1.

2. We define an event E_2 when the realization of σ^* enters step 2b. Note that if we enter step 2b, then there has been no overflow in the first knapsack in step 1.

   We define an event E_{21} which is a sub event of E_2, where the first item of type T_3 inserted into the second knapsack realizes to size at least ϵ. In this event there is no profit lost in the two knapsack experiment, since if the first item realizes to size more than 1, then there is overflow both in second knapsack and σ^*. Similarly, when the first item realizes to a size between ϵ and 1, there is no overflow in the second knapsack and items in the remainder of the policy in σ^* now get added to the

first knapsack with their total size no more than free space available in first knapsack.

Now we want to compute the total profit lost in the two knapsack experiment, by computing the profit of the items which cause an overflow in one of the two knapsacks but with no associated overflow in σ^*. The overflow in the two knapsack experiment which is not associated with an overflow in σ^* can only occur in knapsack 1 in step 1 or in knapsack 2 in step 2b, but not both.

Now we compute the profit lost in the two knapsack experiment compared to σ^* in a knapsack of capacity 1. Observe that the events E_1 and E_2 are disjoint. The expected profit lost in event E_1 is at most OPT/4. In event $E_2 - E_{21}$, the expected profit lost is at most 3OPT/8. Hence, total profit lost is

$$\frac{\Pr(E_1)\text{OPT}}{4} + \frac{(\Pr(E_2) - \Pr(E_{21}))3\text{OPT}}{8}$$

As $\Pr(E_{21}) \geq \Pr(E_2)/3$, the total profit lost is at most $\frac{\text{OPT}}{4}$

Thus in at least one of the two knapsacks, the expected profit is at least $\frac{\text{OPT}-\text{OPT}/4}{2} = \frac{3\text{OPT}}{8}$. $\square$

Thus in each of two cases, we have established the existence of a policy as required in Lemma 5.1.

6 An Approximation Scheme for Non-Adaptive Policies

In this section, we briefly outline the PTAS for non-adaptive policies. Let OPT$'$ be the profit of the optimal non adaptive policy for knapsack capacity 1 and size distributions π^*. Let σ^c be the optimal non-adaptive policy for size distributions π^c and knapsack capacity $\mathbb{C} = 1 + \epsilon$ such that it satisfies the properties in Lemma 2.4. It is easy to see that the ordered set of items chosen by σ^c can be partitioned into contiguous sets of items called *blocks* that satisfy the following properties:

1. A block with exactly one item has profit at least ϵOPT$'$, while a block with at least two items has total profit at most $2(\epsilon$OPT$')$.

2. Sum of the profits of any two consecutive blocks is at least $2(\epsilon$OPT$')$.

3. Number of blocks is $O(1/\epsilon^2)$.

The last property follows as sum of profits of items in σ^c is at most $O(\text{OPT}'/\epsilon)$ and sum of profits of two consecutive blocks in σ^c is at least $(\epsilon$OPT$')$.

Furthermore, any policy that agrees with σ^c on items played in each block but not necessarily on the

order in which the items are played collects $(1-2\epsilon)$OPT$'$ profit since it can at most fail to collect the profit from the last block played in any realization of σ^c. Thus to $(1 + O(\epsilon))$-approximate σ^c, it suffices to merely guess items played in each block of σ^c, ignoring the order in which they are played. We can perform this guessing in polynomial time, at expense of losing another $O(\epsilon$OPT$')$ profit, as explained next.

We can assume w.l.o.g. that σ^c plays items with same size distribution in non-increasing order of profit. We can now use a similar enumeration scheme as used in the adaptive case to obtain a $(1 + O(\epsilon))$-approximation to σ^c.

7 Stochastic Multidimensional Knapsack

We briefly outline the proof of Theorem 1.3. By applying our discretization technique to each dimension, then we get $2^{\text{poly}(1/\epsilon)} \log n$ size distributions in each dimension. Thus if there are d dimensions, then the number of size distributions increases to $\left(2^{\text{poly}(1/\epsilon)} \log n\right)^d$. We will now show how to reduce it to $2^{\text{poly}(d/\epsilon)} \log n$.

For an item b_i, let $\mu_i(small, j)$ and $\pi_i(large, j)$ indicate the expected size in the small size region and the probability in the large size region in the jth dimension. For any item b_i, we perform additional discretization steps as follows.

Step a: Similar to step 2 of the discretization for one dimension, probability values for all large sizes across all dimensions can be stored in a relativized manner of the maximum among them. The profit lost in this step is $O(d\epsilon$OPT$)$. So the number of size distributions over all dimensions will be $2^{\text{poly}(d/\epsilon)} \log n$.

Step b: Let $k_2 = \text{argmax}_j\{\mu_i(small, j)\pi_i(small, j)\}$. For all dimensions j such that

$$\mu_i(small, j)\pi_i(small, j) \leq \epsilon^3 \mu_i(small, k_2)\pi_i(small, k_2),$$

we neglect the small size distribution.

To prove that this discretization change does not reduce the expected profit by more than $O(d\epsilon$OPT$)$, we introduce an auxiliary knapsack of capacity ϵ in each dimension. We charge to the auxiliary knapsack, realization to small size of any item in the dimension with small size distribution neglected. It can be shown that, in any realization of the policy, the overflow probability of the auxiliary knapsack before the end of the policy is $O(\epsilon^2)$, which imply a loss of $O(\epsilon$OPT$)$ per dimension.

This enables us to store small size distributions over all dimensions in a relativized manner, and there are at most $poly(1/\epsilon)$ possible values of the small size distribution for any dimension other than the dimension in which the small size distribution is stored explicitly.

Step c: Similar to steps 3 and 4 in discretization, it can be shown that, either the large size distribution or the small size distribution need to stored explicitly, and given a value for one of them, there are only $2^{\text{poly}(d/\epsilon)}$ possible values for the other.

Thus the number of size distributions reduce to $2^{\text{poly}(d/\epsilon)} \log n$. It can be shown that, with constant number of dimensions, there exists a block adaptive policy which gives a $(1 + O(\epsilon))$ approximation to the adaptive optimum and we get a PTAS by enumerating over all such policies.

Now we briefly sketch the result with no extra space in knapsack. For this purpose, consider a generalization of the two knapsack experiment where we have two knapsacks in each dimension. We start by adding items according to the optimum policy, with items being added to the first knapsack in each dimension. If the first knapsack is more than ϵ full in any dimension, we switch to the second knapsack in that dimension. We continue adding items until the end of the policy. We lose profit of at most d items in any realization, namely the last item in first knapsack in each dimension. The experiment partitions the policy into $(d+1)$ parts, along with d items whose profit is lost in two knapsack experiment. Thus we get a factor $(2d + 1 + \epsilon)$ approximation with no extra space.

8 Approximation Schemes for the Fixed Set Models for Stochastic Knapsack

In this section, we give a PTAS for the fixed set models of stochastic knapsack, establishing Theorems 1.4 and 1.5.

8.1 The Bounded Overflow Probability Model

Recall that in this model, the objective is to identify subset of items S with maximum total profit subject to the constraint

$$\Pr\left(\sum_{1 \leq i \leq |S|} X_i > 1\right) \leq \gamma$$

for some *constant* γ, that is referred to as the *overflow probability*. Here $X_1, X_2, ..., X_{|S|}$ are the random variables corresponding to the sizes of items in S. We now state our result for this model.

THEOREM 8.1. *For any fixed $\epsilon > 0$, for any constant overflow probability γ, there is a polynomial time algorithm to compute a set of items S such that the sum of profits of items in S is $(1 - \epsilon)$ factor of the optimum and its overflow probability for a knapsack capacity $(1 + \epsilon)$ is at most $(1 + \epsilon)\gamma$.*

Proof. We assume that $\gamma \leq 1/(1 + \epsilon)$; otherwise the problem admits a trivial solution. Since we are only interested in the deviation in the probability for a set of items for realizing to size more than 1 and the probability of an item realizing to size more than 1 is crucial in this formulation, we do not scale profits of huge profit items. In Lemma 8.1, we limit the sum of expected sizes of items in a set subject to the constraint that its overflow probability is less than $1 - \epsilon$, so that we can use the discretization process in Section 3.

LEMMA 8.1. *For any set S with overflow probability $\gamma \leq 1/(1 + \epsilon)$, the sum of expected sizes of items in S is $O(1/\epsilon)$.*

Proof. By Lemma 2.1, the sum of expected sizes of items successfully inserted into the knapsack is at most $O(1)$. Hence if the sum of expected sizes of items in S is more than $O(1/\epsilon)$, then by Markov's inequality, the probability that all items in S fit in the knapsack is less than ϵ, which is a contradiction, since we have assumed that overflow probability of S is no more than $1/(1+\epsilon)$. □

We choose $\epsilon' = \epsilon\gamma/2$, and use ϵ' as the parameter for the discretization process in Section 3. The discretization process ensures that for any set S with sum of expected sizes of items at most $O(1/\epsilon')$

1. $\Pr\left(X^c(S) > 1 + \epsilon'\right) \leq \Pr\left(X^*(S) > 1\right) + \epsilon'$.

2. $\Pr\left(X^*(S) > 1 + 2\epsilon'\right) \leq \Pr\left(X^c(S) > 1 + \epsilon'\right) + \epsilon'$

where $X^*(S)$ and $X^c(S)$ are the random variables indicating the sum of sizes of items in S with original and discretized distributions.

We enumerate over all possible sets to guess the optimal set when the overflow probability is $(1+\epsilon')\gamma$ and the knapsack capacity is $(1 + \epsilon')$. With $2^{poly(1/\epsilon')} \log n$ types of items, this can be done in time $n^{2^{poly(1/\epsilon')}}$. The overflow probability of the set computed with distributions π^* on items and the knapsack capacity $(1 + 2\epsilon')$ is $(\gamma + 2\epsilon') = (1 + \epsilon)\gamma$. This completes the proof. □

8.2 The All-or-None Model

In the model, the objective is to identify the set S which maximizes

$$\Pr\left(\sum_{1 \leq i \leq |S|} X_i \leq 1\right) \cdot (\text{Profit of items in } S)$$

where $X_1, X_2, ..., X_{|S|}$ are the random variables corresponding to the sizes of items in S. For this model, Dean et al [DGV08] gives a factor 9.5 approximation. We now state our result for this model.

THEOREM 8.2. *For any fixed $\epsilon > 0$, there is a polynomial time algorithm to compute a set of items S with total profit is at least $(1-\epsilon)$ factor of the optimum when the knapsack capacity is relaxed to $(1+\epsilon)$.*

For any set of items S, let $f(S)$ be the probability that the items in S fit in the knapsack and $P(S)$ be the sum of profits of items in S. Let OPT$'$ be the expected profit of the optimal set and $S_{\text{OPT}'}$ be the corresponding set. Thus OPT$' = \text{P}(S_{\text{OPT}'})\text{f}(S_{\text{OPT}'})$. The overflow probability of $S_{\text{OPT}'}$ is $1 - f(S_{\text{OPT}'})$.

In Section 8.1, given an overflow probability γ, we looked at the problem of finding a subset with maximum total profit and the overflow probability at most $(1+\epsilon)\gamma$. We cannot use it directly to compute the maximum profit set for the overflow probability $1 - f(S_{\text{OPT}'})$ even if we could guess the value of $f(S_{\text{OPT}'})$, because the solution found in section 8.1 increases the overflow probability by $(1 + \epsilon)$ factor, which can arbitrarily reduce the probability of *fitting* a set of items into the knapsack when the overflow probability is close to 1. Also when the value of $f(S_{\text{OPT}'})$ is arbitrarily close to 1, the overflow probability can be arbitrarily small and running time of the algorithm given in Section 8.1 depends exponentially on overflow probability. We first establish an useful property of the optimal solution in all-or-none-model in Lemma 8.2.

LEMMA 8.2. *For the set $S_{\text{OPT}'}$, either $f(S_{\text{OPT}'}) \geq \epsilon$ or it contains an item such that only playing that item achieves expected profit $(1 - O(\epsilon))$OPT$'$*

Proof. We only need to consider the case when $f(S_{\text{OPT}'}) < \epsilon$. Order the items in $S_{\text{OPT}'}$ in decreasing order of profit. Partition the items in $S_{\text{OPT}'}$ into two sets S_1 and S_2 as follows. We add items from $S_{\text{OPT}'}$ in the decreasing order of profit to S_1 until the sum of profits of items in S_1 is more than $P(S_{\text{OPT}'})/3$. Remaining items in $S_{\text{OPT}'}$ are moved to S_2. Expected profit achieved by only using S_1 is $P(S_1)f(S_1)$. Similarly the expected profit achieved by only using S_2 is $P(S_2)f(S_2)$. Clearly, $f(S_{\text{OPT}'}) \leq f(S_1)f(S_2)$.

We consider various cases based on the values of $f(S_1)$ and $f(S_2)$. In the first case, both $f(S_1)$ and $f(S_2)$ are less than $1/2$. In this case, the expected profit of $S_{\text{OPT}'}$ is at most $(P(S_1) + P(S_2))f(S_1)f(S_2)$ which is less than at least one of $P(S_1)f(S_1)$ and $P(S_2)f(S_2)$, hence it contradicts that $S_{\text{OPT}'}$ is the optimal set.

When $f(S_1) \geq 1/2$, the expected profit by only playing items from S_1 is $P(S_1)/2 \geq P(S_{\text{OPT}'})/6 > f(S_{\text{OPT}'})P(S_{\text{OPT}'})$ for small enough ϵ, as $f(S_{\text{OPT}'}) < \epsilon$, hence it contradicts that $S_{\text{OPT}'}$ is the optimal set.

Now the remaining case is when $f(S_2) \geq 1/2$. In this case, we consider two sub cases. If $P(S_2) >$ $2\epsilon P(S_{\text{OPT}'})$, then the expected profit of S_2 is more than $2\epsilon P(S_{\text{OPT}'})/2 > f(S_{\text{OPT}'})P(S_{\text{OPT}'})$, hence it contradicts that $S_{\text{OPT}'}$ is the optimal set. Otherwise, $P(S_2) \leq 2\epsilon P(S_{\text{OPT}'})$. Hence, $P(S_1) \geq (1-2\epsilon)P(S_{\text{OPT}'})$ and the expected profit by only playing S_1 is $(1 - 2\epsilon)P(S_{\text{OPT}'})f(S_1) \geq (1 - 2\epsilon)P(S_{\text{OPT}'})f(S_{\text{OPT}'})$. Now we have to show that S_1 contains only one item. This follows from the definition of S_1, when ϵ is small enough. Items from $S_{\text{OPT}'}$ are added in descending order of profit and we stop adding items to S_1, when for the first time the items in S_1 have profit more than $P(S_{\text{OPT}'})/3$. $\square$

Now we consider the problem of computing such set S. The only interested case is when $f(S_{\text{OPT}'}) \geq \epsilon$. We select $\epsilon' = \epsilon^2$ as approximation parameter for the bounded overflow probability problem and we solve it for each value of overflow probability between ϵ and $1-\epsilon$ in the geometric group which is powers of $1/(1 + \epsilon')$ for approximation factor ϵ' and take the maximum of the product of sum of profits of items and probability of all items fitting the knapsack,

Now we establish a lower bound on the expected profit of the set computed by the algorithm. Notice that, the sum of profits of items in the optimal solution for the bounded overflow probability problem never decreases with an increase in the overflow probability. We consider two cases. In the first case, $f(S_{\text{OPT}'}) \leq 1 - \epsilon$. In this case, $f(S_{\text{OPT}'}) \geq \epsilon$. There is an overflow probability value between $(1 - f(S_{\text{OPT}'}))$ and $(1 - f(S_{\text{OPT}'}))(1 + \epsilon')$ for which we have computed the approximate solution for the bounded overflow probability problem. Let S be that set. We get, $P(S) \geq (1 - \epsilon')P(S_{\text{OPT}'})$ and its overflow probability is at most $(1+\epsilon')(1 - f(S_{\text{OPT}'}))$. Hence $f(S)$ is at least $(1 - \epsilon)f(S_{\text{OPT}'})$, as $\epsilon' = \epsilon^2$ and $f(S_{\text{OPT}'}) \geq \epsilon$. Thus the expected profit of items in S is $f(S)P(S)$, which is $(1 - \epsilon)f(S_{\text{OPT}'})P(S_{\text{OPT}'})$.

In the second case, $f(S_{\text{OPT}'}) > 1 - \epsilon$. In this case, the set computed for the overflow probability ϵ approximates the optimal solution within a factor of $(1 - \epsilon)$.

References

[CK00] C. Chekuri and S. Khanna. A ptas for the multiple knapsack problem. *SODA*, pages 213–222, 2000.

[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 Res. Logist.*, 40:161–173, 1993.

[DGV05] B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *SODA '05: Proceedings of the 16th*

Symposium on Discrete Algorithms, pages 395–404, Vancouver, British Columbia, Canada, 2005. SIAM.

[DGV08] B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity. *MATHEMATICS OF OPERATIONS RESEARCH*, 33(4):945–964, 2008.

[DLR78] C. Derman, C. Lieberman, and S. Ross. Stochastic optimization is (almost) as easy as deterministic optimization. *Management Science*, 24(5):554–561, 1978.

[GI99] A. Goel and P. Indyk. Stochastic load balancing and related problems. In *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 579–586, Washington, DC, USA, 1999. IEEE Computer Society.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

[Hen90] M Henig. Risk criteria in stochastic knapsack. *Oper. Res.*, 38(5):820–825, 1990.

[HKLOS08] N. Halman, D. Klabjan, C. Li, J. Orlin, and D. Simchi-Levi. Fully polynomial time approximation schemes for stochastic dynamic programs. In *SODA '08: Proceedings of the 19th Symposium on Discrete Algorithms*, pages 700–709, San Francisco, California, USA, 2008. SIAM.

[IKMM04] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On costs and benefits of procrastinations: Approximation algorithms for stochastic combinatorial optimization problems. *Proc. 15th SODA*, pages 684–693, 2004.

[KK82] N. Karmarkar and R. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. *Proc. 23rd Annual IEEE Symposium on the Foundations of Computer Science*, 1982.

[KRT97] J. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 664–673, New York, NY, USA, 1997. ACM.

[RS04] R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. *Proc. 10th IPCO*, pages 101–115, 2004.

[Shm04] C. Swamy Shmoys, D. B. Stochastic optimization is (almost) as easy as deterministic optimization. *Proc. 45th FOCS*, pages 228–237, 2004.

[Sne80] M Sneidovik. Preference order stochastic knapsack problems: Methodological issues. *J. Oper. Res Soc.*, 31(11):1025–1032, 1980.

[SP79] E Steinberg and M. S. Parks. Preference order dynamic program for a knapsack problem with stochastic rewards. *J. Oper. Res Soc.*, 30(2):141–147, 1979.

A Discretization of Distributions: The Full Process

We now provide a detailed description of the transformations to discretize the size distributions of items.

For each step in the discretization, we will use notation π and π' to indicate the vector of size distributions on items before and after performing the discretization mentioned in the given step, i.e. for an item b_i, size distributions before and after the discretization changes mentioned in the given step are π_i and π_i' respectively. All references for $\mathbb{C}$ i.e. the capacity of knapsack will be for any $\mathbb{C}, 1 \leq \mathbb{C} \leq \mathbb{C}_{max}$. Also, we restrict the allowed set of policies to ones which satisfy the constraints mentioned in section 2.

A.1 Step 1 We discretize the sizes in the large size region into geometric groups. For each item, $\forall j \geq 1, j \in \mathbb{N}, 1/(1+\epsilon)^{j-1} \geq \epsilon^5$, all probability mass between sizes $[1/(1+\epsilon)^j, 1/(1+\epsilon)^{j-1})$ is assigned to size $1/(1+\epsilon)^j$. Thus for an item b_i, its new distribution in the large size region is

$$\Pr_{X_i' \sim \pi_i'}\left(X_i' = \frac{1}{(1+\epsilon)^j}\right) =$$
$$\Pr_{X_i \sim \pi_i}\left(X_i \in [\frac{1}{(1+\epsilon)^j}, \frac{1}{(1+\epsilon)^{j-1}})\right),$$
$$\forall j \geq 1, \frac{1}{(1+\epsilon)^{j-1}} \geq \epsilon^5$$

The number of different sizes for which size distribution is defined is
$$\frac{\log(1/\epsilon^5)}{\log(1+\epsilon)} = O(1/\epsilon^2)$$

We will use q' to denote this number.

Clearly for any policy σ, $\mathbb{P}(\sigma, \pi', \mathbb{C}) \geq \mathbb{P}(\sigma, \pi, \mathbb{C})$ and $\mathbb{P}(\sigma, \pi, (1+\epsilon)\mathbb{C}) \geq \mathbb{P}(\sigma, \pi', \mathbb{C})$.

A.2 Step 2 Now we discretize the probability distribution in the large size region. For this purpose, we establish an important property in Lemma A.1.

LEMMA A.1. *Given a vector of distributions π on items and a knapsack of capacity $\mathbb{C}$ ($1 \leq \mathbb{C} \leq \mathbb{C}_{max}$), consider any change in the large size distributions of items to π' such that for any item b_i*

- *For any $s(\neq 0), \pi_i(s) \geq \pi_i'(s)$ and*

- $\Delta(\pi_i, \pi_i') \leq \epsilon^8 \pi_i(large)$

- *For each item b_i, the probability for size 0 is increased in π_i' to compensate for the probability mass reduced elsewhere.*

Then for any policy σ,

1. $\mathbb{P}(\sigma, \pi', \mathbb{C}) \geq \mathbb{P}(\sigma, \pi, \mathbb{C})$ and

2. $\mathbb{P}(\sigma, \pi, \mathbb{C}) \geq \mathbb{P}(\sigma, \pi', \mathbb{C}) - O(\epsilon \text{OPT})$

Proof. First claim is trivial since for each item b_i, the distribution π_i dominates $\pi_i^{'}$ for every positive size realization.

Now we prove the second claim. To compute the loss in profit for the policy σ when items come from π compared to when items come from π', we are interested in finding the probability of the event that, any item, say b_i, realizes to different values under distributions π_i and $\pi_i^{'}$ before the policy terminates. If such event happens, then we discard the current item and stop the policy. By Lemma 2.4, the total profit in any realization is $O(\text{OPT}/\epsilon)$, hence to prove the lemma, it suffices to show that probability of this event is $O(\epsilon^2)$.

For any item b_i, $\mu_i \geq \epsilon^5 \pi_i(large)$. Let S be the (random) set of items which the policy has attempted to insert. By Lemma 2.4,

$$\sum_{b_i \in S} \mu_i = O(1/\epsilon)$$

Hence

$$\sum_{b_i \in S} \epsilon^5 \pi_i(large) = O(1/\epsilon)$$

By assumptions of the lemma on π and π', we get

$$\sum_{b_i \in S} \left(\frac{\Delta(\pi_j, \pi_j^{'})}{\epsilon^8} \right) \epsilon^5 = O(1/\epsilon)$$

$$\sum_{b_i \in S} \Delta(\pi_j, \pi_j^{'}) = O(\epsilon^2)$$

This completes the proof. $\square$

We now use Lemma A.1 to discretize the probability values in the large size region.

1. For every item, we round down the probability for each discretized size in the large size region to the nearest power of $\frac{1}{(1+\epsilon^8)}$. Clearly for each item b_i, $\Delta(\pi_i, \pi_i^{'}) \leq \epsilon^8 \pi_i(large)$

2. Recall that there are q' different sizes in the large size region for which the distribution is defined. For each item b_i, for each size s in these q' different sizes, if $\pi_i(s) \leq \frac{\epsilon^8 \pi_i(large)}{q'}$, we make $\pi_i(s)$ zero. Clearly for each item b_i, $\Delta(\pi_i, \pi_i^{'}) \leq \epsilon^8 \pi_i(large)$. This helps us to indicate the large size distribution *relative* of the maximum probability for any size in the large size region.

We perform one more change. For each item b_i, if for any discretized size s in the large size region $\pi_i(s) \leq \frac{\epsilon^2}{nq'}$, then we make $\pi_i(s)$ zero. Over all items, total probability mass which is neglected is $O(\epsilon^2)$ and the expected profit is reduced by $O(\epsilon\text{OPT})$.

By representing the probability values in the large size region relative of the highest value, we get following lemma.

LEMMA A.2. *Number of possible different values of large size distributions is* $2^{poly(1/\epsilon)} \log n$

A.3 Step 3 For any item b_i, if $\pi_i(large) \leq \epsilon^3 \mu_i(small)/2$, then we neglect its large size distribution. The new distribution π' for any such item b_i will have zero probability in the large size region. For any policy σ, create a policy σ' as follows: σ' follows σ with the exception that it *stops* when an item b_i with $\pi_i(large) \leq \epsilon^3 \mu_i(small)/2$ realizes to a large size. Clearly, $\mathbb{P}(\sigma', \pi', \mathbb{C}) \geq \mathbb{P}(\sigma, \pi, \mathbb{C})$. In Lemma A.3, we establish that $\mathbb{P}(\sigma', \pi, \mathbb{C}) \geq \mathbb{P}(\sigma', \pi', \mathbb{C}) - O(\epsilon\text{OPT})$.

LEMMA A.3. $\mathbb{P}(\sigma', \pi, \mathbb{C}) \geq \mathbb{P}(\sigma', \pi', \mathbb{C}) - O(\epsilon\text{OPT})$.

Proof. Let S be the (random) set of items which policy σ' attempts to insert such that the large size distribution of these items is neglected. We discard the entire profit of a realization if any item in S realizes to large size before the policy terminates. It is sufficient to show that the probability of this event is at most $O(\epsilon^2)$, which by Lemma 2.4 would imply a loss of $O(\epsilon\text{OPT})$ in the profit.

We know that,

$$\sum_{b_i \in S} \mu_i(small)\pi_i(small) \leq \sum_{b_i \in S} \mu_i$$

By Lemma 2.4, we get,

$$\sum_{b_i \in S} \mu_i = O(1/\epsilon)$$

Thus we have,

$$\sum_{b_i \in S} \mu_i(small)\pi_i(small) = O(1/\epsilon)$$

Since for any item b_i in S, $\pi_i(large) \leq \epsilon^3 \mu_i(small)/2$, we get,

$$\sum_{b_i \in S} \frac{2\pi_i(large)\pi_i(small)}{\epsilon^3} = O(1/\epsilon)$$

Since $\pi_i(large) \leq \epsilon^3 \mu_i(small)/2$ and $\mu_i(small) \leq \epsilon^5$ we can assume that $\pi_i(large) \leq 1/2$ and since $\pi_i(large) + \pi_i(small) = 1$, hence $\pi_i(small) \geq 1/2$. We get,

$$\sum_{b_i \in S} \pi_i(large) = O(\epsilon^2)$$

Thus the probability that any item in S realizes to a large size is at most $O(\epsilon^2)$. This completes the proof. $\square$

A.4 Step 4 For each item b_i, if $\mu_i(small) \leq \epsilon^9 \pi_i(large)$, then we neglect its small size distribution. The new distribution π_i' for any such item b_i remains unchanged in the large size region and b_i takes size 0 w.p. $\pi_i(small)$.

For any policy σ, create a policy σ' as follows. σ' is identical to σ with one exception. For any item b_i with $\mu_i(small) \leq \epsilon^9 \pi_i(large)$, if b_i realizes to a small size, then σ' follows the remainder of the policy assuming b_i has realized to size 0. Clearly, $\mathbb{P}(\sigma', \pi', \mathbb{C}) \geq \mathbb{P}(\sigma, \pi, \mathbb{C})$. We need to show that, $\mathbb{P}(\sigma', \pi, (1+\epsilon)\mathbb{C}) \geq \mathbb{P}(\sigma', \pi', \mathbb{C}) - O(\epsilon\mathrm{OPT})$.

LEMMA A.4. $\mathbb{P}(\sigma', \pi, (1 + \epsilon)\mathbb{C}) \geq \mathbb{P}(\sigma', \pi', \mathbb{C}) - O(\epsilon\mathrm{OPT})$.

Proof. Let S be the (random) set of items b_i which σ' has attempted such that $\mu_i(small) \leq \epsilon^9 \pi_i(large)$. In addition to the main knapsack of capacity $\mathbb{C}$, we create an *auxiliary knapsack*. If any item b_i with $\mu_i(small) \leq \epsilon^9 \pi_i(large)$ realizes to a small size (i.e. $\mu_i(small)$), we charge it to the auxiliary knapsack, otherwise we put it into the main knapsack. Other items, i.e. items b_i with $\mu_i(small) > \epsilon^9 \pi_i(large)$, are always put into main knapsack. We need to compute the sufficient size for the auxiliary knapsack when the permitted policies are constrained by Lemma 2.4. By Lemma 2.4, for any policy

$$\sum_{b_i \in S} \mu_i = O(1/\epsilon)$$

By conditions on items in S, we get

$$\sum_{b_i \in S} \mu_i(small) \leq \sum_{b_i \in S} \epsilon^9 \pi_i(large)$$

By definition of large size region, for any item b_i, $\mu_i \geq \epsilon^5 \pi_i(large)$. Thus

$$\sum_{b_i \in S} \mu_i(small) \leq \sum_{b_i \in S} \epsilon^4 \mu_i = O(\epsilon^3).$$

If we limit the size of the auxiliary knapsack to ϵ, then by Markov's inequality, w.p. $(1 - O(\epsilon^2))$ there is no overflow in the auxiliary knapsack. Using Lemma 2.4, the profit lost is $O(\epsilon\mathrm{OPT})$ $\qquad \square$

A.5 Step 5 Now we discretize the probability distribution in the small size region. For each item b_i, we replace the probability distribution in the small size region by a single size $\mu_i(small)$ i.e. with probability $\pi_i(small)$, b_i realizes to size $\mu_i(small)$.

Recall that canonical policies treat small size realizations by their expected sizes when small. Thus the decision taken by the policy is based on the canonical sizes of items already inserted, not the true space used in the knapsack.

LEMMA A.5. *1. For any policy σ, there exists a canonical policy σ^c such that $\mathbb{P}(\sigma^c, \pi', (1 + \epsilon)\mathbb{C}) \geq \mathbb{P}(\sigma, \pi, \mathbb{C}) - O(\epsilon\mathrm{OPT})$.*

2. For any canonical policy σ^c, $\mathbb{P}(\sigma^c, \pi, (1 + \epsilon)\mathbb{C}) \geq \mathbb{P}(\sigma^c, \pi', \mathbb{C}) - O(\epsilon\mathrm{OPT})$.

Proof. For any leaf in the decision tree of σ for the size distributions π and the knapsack capacity $\mathbb{C}$, we define its *weight* to be the probability of reaching that leaf when the policy is executed on items with distributions π. Recall that any path from root to leaf is associated with a size realization for each intermediate node. For any given leaf, let $S = \{b_1, b_2, ..., b_{|S|}\}$ be the set of items on the path which realize to small size and let the corresponding sizes be $s_1, s_2, ..., s_{|S|}$. We call a leaf to be a *bad leaf* if $|\sum_{\{b_i \in S\}} (\mu_i(small) - s_i)| \geq \epsilon$, otherwise we call it to be a *good leaf*.

We will prove the lemma in three steps.

Step a: We first show that in the decision tree of σ, the total weight of bad leaves is $O(\epsilon^2)$. Thus the expected profit of the policy restricted to the paths ending in good leaves is $\mathbb{P}(\sigma, \pi, \mathbb{C}) - O(\epsilon\mathrm{OPT})$.

Step b: We then show that there is a randomized policy which for items with size distributions π' achieves the expected profit $\mathbb{P}(\sigma, \pi, \mathbb{C}) - O(\epsilon\mathrm{OPT})$ with ϵ extra space in the knapsack. This also implies the existence of a deterministic policy in such setting, thus establishing the part 1 of the lemma.

Step c: To prove the part 2 of the lemma, we show that if we use the canonical policy σ^c for items with size distributions π, then with ϵ extra space in the knapsack, the expected profit of the policy is $\mathbb{P}(\sigma^c, \pi', \mathbb{C}) - O(\epsilon\mathrm{OPT})$.

Proof of Step a: For a node ν, let S be the random set of items which is played by the policy after reaching node ν (including the item at node ν). We define the random variable R_ν as

$$R_\nu = \sum_{b_i \in S} \left(X_i - \mu_i(small) | X_i \leq \epsilon^5 \right)$$

We denote its variance by $var(\nu)$. Clearly, $E[R_\nu] = 0$ and

$$var(\nu) = E\left[\left(\sum_{b_i \in S} \left(X_i - \mu_i(small) | X_i \leq \epsilon^5 \right) \right)^2 \right]$$

Similarly, we define the variance of an item b_i as

$$var(b_i) = E_{X_i \sim \pi_i} \left[\left(X_i - \mu_i(small) | X_i \leq \epsilon^5 \right)^2 \right]$$

Clearly, $var(b_i) \leq \epsilon^5 \mu_i(small)$. We will use $w(\nu)$ to indicate maximum total expected size of items on any path starting at node ν to a leaf. We will inductively show that, for any node ν in the decision tree, $var(\nu) \leq \epsilon^5 w(\nu)$. Since for any path in the decision tree, the sum of expected size of items is bounded by $O(1/\epsilon)$, the variance of the root node is $O(\epsilon^4)$. We can then apply Chebychev's inequality to show that total weight of bad leaves is $O(\epsilon^2)$.

Now we prove the inductive claim where the induction is applied bottom up in the decision tree of the policy σ. The induction claim is trivially true for the leaf nodes. Let ν be a node in consideration. Let the corresponding item be b_ν, $s_1, s_2, ..., s_k$ be its possible small size realizations, the corresponding probabilities be $\pi_\nu(s_1), \pi_\nu(s_2), ..., \pi_\nu(s_k)$ and the roots of the remainder of the policy for these realizations be $\nu_{s_1}, \nu_{s_2}, ..., \nu_{s_k}$ respectively. Let $l_1, l_2, ..., l_{k'}$ be its possible large size realization, the corresponding probabilities be $\pi_\nu(l_1), \pi_\nu(l_2), ..., \pi_\nu(l_{k'})$, and the roots of the remainder of the policy for these realizations be $\nu_{l_1}, \nu_{l_2}, ..., \nu_{l_{k'}}$ respectively.

The variance of a random variable R_ν is,

$$var(\nu) = \sum_{i,1\leq i\leq k} E\left[\pi_\nu(s_i)\left(R_{\nu_{s_i}} + s_i - \mu_\nu(small)\right)^2\right] + \sum_{i,1\leq i\leq k'} E\left[\pi(l_i)\left(R_{\nu_{l_i}}\right)^2\right]$$

The first term simplifies as follows.

$$\sum_{i,1\leq i\leq k} E\left[\pi_\nu(s_i)\left(R_{\nu_{s_i}} + s_i - \mu_\nu(small)\right)^2\right]$$
$$= \sum_{i,1\leq i\leq k} E\left[\pi_\nu(s_i)\left(R_{\nu_{s_i}}\right)^2\right]$$
$$+ \sum_{i,1\leq i\leq k} E\left[\pi_\nu(s_i)(s_i - \mu_\nu(small))^2\right]$$
$$+ \sum_{i,1\leq i\leq k} 2\pi_\nu(s_i)(s_i - \mu_\nu(small))E\left[R_{\nu_{s_i}}\right]$$
$$\leq \left(\sum_{i,1\leq i\leq k} \pi_\nu(s_i)var(\nu_{s_i})\right) + var(b_\nu) + 0$$

Thus we get,

$$var(\nu) \leq \left(\sum_{i,1\leq i\leq k} \pi_\nu(s_i)var(\nu_{s_i})\right) + \left(\sum_{i,1\leq i\leq k'} \pi_\nu(l_i)var(\nu_{l_i})\right) + var(b_\nu)$$

Using the fact that

$$\sum_{i,1\leq i\leq k} \pi_\nu(s_i) + \sum_{i,1\leq i\leq k'} \pi_\nu(l_i) = 1$$

we get

$$var(\nu) \leq max\{max_{\{i,1\leq i\leq k\}}(var(\nu_{s_i})),$$
$$max_{\{i,1\leq i\leq k'\}}(var(\nu_{l_i}))\} + \epsilon^5 \mu_\nu(small)$$

The proof follows by applying the inductive assumption.

Proof of Step b: Consider a randomized policy σ^c for items in π^c that is generated from σ by following random process. When an item realizes to a large size, it follows the remainder of the policy as specified by σ. When an item realizes to a small size, it decides the remainder of policy as follows. Let ν be the node and b_ν be the item corresponding to it. Let $s_1, s_2, ..., s_k$ are possible small size realizations of b_ν as per size distribution π and let $\pi_\nu(s_1), \pi_\nu(s_2)..., \pi_\nu(s_k)$ be the corresponding probabilities. Hence at node ν, if b_ν realizes to a small size, we choose one of the branches corresponding to sizes $s_1, s_2, ..., s_k$ w.p. $\pi_\nu(s_1), \pi_\nu(s_2)..., \pi_\nu(s_k)$ (normalized) respectively.

In this randomized policy, for any leaf in σ, the probability of reaching the leaf remains same as in σ if we discount the difference between sum of canonical and true sizes of items. For any good leaf, the sum of canonical and true sizes of items differ by at most ϵ. Since the weight of bad leaves is $O(\epsilon^2)$ and any realization has profit $O(\text{OPT}/\epsilon)$, the total profit lost is $O(\epsilon \text{OPT})$.

Proof of Step c: Given an arbitrary canonical policy σ^c, consider any leaf ν in its decision tree. Let $S = \{b_1, b_2, ..., b_{|S|}\}$ be the set of items realizing to small size on the path from root to this leaf. Consider a scenario when we use σ^c for items with distributions π and we reach leaf ν during the execution of the policy. Let $X_1, X_2, ..., X_{|S|}$ be the random variables indicating the sizes to which items in S have realized. We get the profit if $\sum_{\{b_i \in S\}}(X_i - \mu_i(small)) \leq \epsilon$ as we are relaxing the knapsack capacity by ϵ. Notice that, since we have arrived at leaf ν, $\forall b_i \in S, X_i \leq \epsilon^5, E[X_i] = \mu_i(small), var(X_i) \leq \epsilon^5 \mu_i(small)$ (items in S have to realize to small size to ensure that correct branch is taken in decision tree to reach leaf ν).

For any given leaf ν, we define a random variable $X = \sum_{\{b_i \in S\}}(X_i - \mu_i(small))$. The variance of X is $\epsilon^5 \cdot (1/\epsilon) = \epsilon^4$. Note, the set S is fixed for a leaf, hence the variance of X is the addition of variances of items (conditioned on the event that they realize to small size) in S. By Chebychev's inequality, $\Pr(|X| \geq \epsilon) = O(\epsilon^2)$. The error probability for each leaf is $O(\epsilon^2)$. Hence profit lost is $O(\epsilon^2) \cdot O(\text{OPT}/\epsilon) = O(\epsilon \text{OPT})$. $\qquad\square$

Proof of Lemma 3.1: The lemma follows as an immediate corollary of Lemma A.5 by selecting σ to be the optimal adaptive policy for a knapsack capacity $\mathbb{C}$ and size distributions π. □

Now we discretize the expected sizes of items in the small size region into the geometric group which is powers of $\frac{1}{1+\epsilon}$ with values below ϵ/n made zero. Thus by increasing the space in the knapsack by a factor of $O(\epsilon)$, the expected profit is unaffected. Total number of different sizes $= \frac{\log(1/\epsilon) + \log(n)}{\log(1+\epsilon)} \leq \frac{2\log(n)}{\epsilon^2}$.