Title:	Stochastic Knapsack
Name:	Viswanath Nagarajan[1]
Affil./Addr.	Industrial and Operations Engineering Department, University of Michigan
Keywords:	Stochastic optimization; adaptive; packing constraints
SumOriWork:	2004; Dean, Goemans, Vondrák 2011; Bhalgat, Goel, Khanna

Stochastic Knapsack

Viswanath Nagarajan[1]
Industrial and Operations Engineering Department, University of Michigan

Years aud Authors of Summarized Original Work

2004; Dean, Goemans, Vondrák
2011; Bhalgat, Goel, Khanna

Keywords

Stochastic optimization; adaptive; packing constraints

Problem Definition

This problem deals with packing a maximum reward set of items into a knapsack of given capacity, when the item-sizes are random. The input is a collection of n items, where each item $i \in [n] := \{1, \cdots, n\}$ has reward $r_i \geq 0$ and size $S_i \geq 0$, and a knapsack capacity $B \geq 0$. In the *stochastic knapsack* problem, all rewards are deterministic but the sizes are random. The random variables S_is are independent with known, arbitrary distributions. The actual size of an item is known only when it is placed into the knapsack. The objective is to add items sequentially (one by one) into the knapsack so as to maximize the expected reward of the items that fit into the knapsack. As usual, a subset T of items is said to fit into the knapsack if the total size $\sum_{i \in T} S_i$ is at most the knapsack capacity B.

A feasible solution (or policy) to the stochastic knapsack problem is represented by a decision tree. Nodes in this decision tree denote the current "state" of the solution (i.e. previously added items and the residual knapsack capacity) as well as the new item to place into the knapsack at this state. Branches in the decision tree denote the random size instantiations of items placed into the knapsack. Such solutions are called *adaptive policies*, to emphasize the fact that the items being placed may depend on previously observed outcomes. More formally, an adaptive policy is given by a mapping $\pi : 2^{[n]} \times [0, B] \to [n]$, where $\pi(T, C)$ denotes the next item to place into the knapsack when some subset $T \subseteq [n]$ of items has already been added, and $C = B - \sum_{i \in T} S_i$ is the residual knapsack capacity. The policy ends when the knapsack overflows (i.e. the

total size of items added exceeds the knapsack capacity); we use the convention that no reward is obtained from the last overflowing item.

Notice that an arbitrary adaptive policy may require exponential space to even store. This motivates a special class of solutions, called *non-adaptive policies*. A non-adaptive policy is just specified by a fixed ordering of the n items, and the solution adds items into the knapsack in that order (irrespective of the actual size instantiations) until the knapsack overflows. Again, there is no reward obtained from the last overflowing item. While it may be easier to obtain a good non-adaptive policy, the obvious drawback is that non-adaptive policies may perform much worse than adaptive policies. The benefit of being adaptive is quantified by a measure called the *adaptivity gap*, which is the maximum ratio (over all instances) of the expected reward of an optimal adaptive policy to the expected reward of an optimal non-adaptive policy.

In both the adaptive and non-adaptive settings, the stochastic knapsack problem is at least NP-hard, since it generalizes the deterministic knapsack problem. Moreover, certain questions regarding adaptive policies are PSPACE-hard [4].

Notation We assume that the item size distributions are given explicitly. For any item $i \in [n]$ define its effective reward $w_i = r_i \cdot \Pr[S_i \leq B]$ and its mean truncated size $\mu_i = \mathbb{E}[\min\{S_i, B\}]$. Note that the expected reward obtained by placing the single item i into the knapsack is exactly w_i.

Key Results

Dean, Goemans and Vondrák introduced the stochastic knapsack problem and the notion of adaptivity gaps. They proved the following.

Theorem 1 ([4]). *There is a polynomial time algorithm for the stochastic knapsack problem that computes a non-adaptive policy having expected reward at least $\frac{1}{4}$ that of an optimal adaptive policy.*

As a consequence, the adaptivity gap of the stochastic knapsack problem is also upper bounded by four. [4] also showed an instance of stochastic knapsack that lower bounds the adaptivity gap by $\frac{5}{4}$.

The algorithm in Theorem 1 uses a natural greedy approach. It outputs the better of the following two non-adaptive policies:

- Place the single item $i^* = \arg\max_{i \in [n]} w_i$.
- Place items in non-increasing order of w_i/μ_i.

In terms of adaptive policies, Bhalgat, Goel and Khanna proved the following.

Theorem 2 ([3; 2]). *For any constant $\epsilon > 0$, there is polynomial time algorithm for the stochastic knapsack problem that computes an adaptive policy having expected reward at least $\frac{1}{2+\epsilon}$ that of an optimal adaptive policy.*

The algorithm in Theorem 2 relies on an intricate transformation of general size distributions to certain canonical distributions, and an algorithm for computing a near-optimal adaptive policy under canonical size distributions.

Extensions

Several variants of the stochastic knapsack problem have been studied, and good algorithms have been obtained for them.

Correlated Stochastic Knapsack This is a generalization of the stochastic knapsack problem, where each item's reward is also random and possibly correlated with its size. The distributions across items are still independent: so the correlations are only between the size and reward of a single item. Gupta, Krishnaswamy, Molinaro and Ravi [6] gave an algorithm for this problem that computes a non-adaptive policy having expected reward within factor 8 of the optimal adaptive policy. Recently, Ma [8] gave an algorithm that for any constant $\epsilon > 0$ computes an adaptive policy having expected reward within factor $2+\epsilon$ of the optimal adaptive policy; this algorithm requires item-sizes and the capacity B to be specified in unary.

Budgeted Multi-Armed Bandit The input to this problem consists of a bound B, and n "arms" (each arm is a Markov chain with rewards at its states, and a specified starting state). A feasible policy consists of B steps. In each step, the policy can select one arm $i \in [n]$: upon selecting arm i, it gets the reward at the current state of arm i and the arm transitions to its next state according to its Markov chain. The objective is to maximize the expected total reward over B steps of the policy. Again, we are interested in adaptive policies, whose actions may depend on past outcomes. Guha and Munagala [5] introduced this problem and gave a $(2+\epsilon)$-approximation algorithm, under the assumption that the rewards of each arm satisfy a "Martingale" condition (which is natural in many settings). Gupta, Krishnaswamy, Molinaro and Ravi [6] gave the first constant-factor approximation algorithm for this problem without the Martingale rewards assumption. The constant factor in the latter result was improved to 6.75 by Ma [8].

Stochastic Orienteering This problem is defined on a finite metric space (V, d) with vertex set V and distance function $d : V \times V \to \mathbb{R}_+$ that satisfies (i) symmetry: $d(u, v) = d(v, u)$ for all $u, v \in V$, and (ii) triangle inequality: $d(u, w) \leq d(u, v) + d(v, w)$ for all $u, v, w \in V$. The distances between vertices denote travel times. Each vertex $i \in V$ corresponds to a job having deterministic reward $r_i \geq 0$ and random processing time $S_i \geq 0$. The random variables S_is are independent with known, arbitrary distributions. Given a start-vertex $\rho \in V$ and bound B, the goal is to compute a policy, which describes a (possibly adaptive) path originating from ρ that visits vertices and runs the respective jobs. The actual processing time of a job is known only when it completes. The policy ends when the the total time (for travel plus processing) exceeds B. The objective is to maximize the expected total reward; there is no reward obtained from a partially completed job (which may occur at the end of the policy). As before, an optimal policy may be adaptive and choose the next job to run based on previously observed outcomes. Gupta, Krishnaswamy, Nagarajan and Ravi [7] gave an $O(\log \log B)$-approximation algorithm for the stochastic orienteering problem; this result requires the bound B, distances, and processing times to be integer valued. As a corollary, [7] also upper bounded the adaptivity gap by $O(\log \log B)$. Recently, Bansal and Nagarajan [1] gave an $\Omega\left(\sqrt{\log \log B}\right)$ lower bound on the adaptivity gap.

Applications

The stochastic knapsack problem and its variants model various applications in advertising, logistics, medical diagnosis and robotics.

Open Problems

It is not known if the stochastic knapsack problem is any harder to approximate than the usual (deterministic) knapsack problem. In particular, is there a PTAS for stochastic knapsack? Determining a tight bound on its adaptivity gap is also an interesting open question.

Recommended Reading

1. Bansal N, Nagarajan V (2014) On the adaptivity gap of stochastic orienteering. In: IPCO, pp 114–125
2. Bhalgat A (2011) A $(2+\epsilon)$-approximation algorithm for the stochastic knapsack problem. Unpublished Manuscript
3. Bhalgat A, Goel A, Khanna S (2011) Improved approximation results for stochastic knapsack problems. In: SODA, pp 1647–1665
4. Dean BC, Goemans MX, Vondrák J (2008) Approximating the stochastic knapsack problem: The benefit of adaptivity. Math Oper Res 33(4):945–964
5. Guha S, Munagala K (2013) Approximation algorithms for bayesian multi-armed bandit problems. CoRR abs/1306.3525
6. Gupta A, Krishnaswamy R, Molinaro M, Ravi R (2011) Approximation algorithms for correlated knapsacks and non-martingale bandits. In: FOCS, pp 827–836
7. Gupta A, Krishnaswamy R, Nagarajan V, Ravi R (2012) Approximation algorithms for stochastic orienteering. In: SODA, pp 1522–1538
8. Ma W (2014) Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms: Extended abstract. In: SODA, pp 1154–1163