European Journal of Operational Research 239 (2014) 625–635

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

An adaptive stochastic knapsack problem ☆

Kai Chen *, Sheldon M. Ross

Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089, United States

ARTICLE INFO

Article history:
Received 29 August 2013
Accepted 19 June 2014
Available online 27 June 2014

Keywords:
Decision process
Dynamic programming
Stochastic knapsack

ABSTRACT

We consider a stochastic knapsack problem in which the event of overflow results in the problem ending with zero return. We assume that there are n types of items available where each type has infinite supply. An item has an exponentially distributed random weight with a known mean depending on its type and the item's value is proportional to its weight with a given factor depending on the item's type. We have to make a decision on each stage whether to stop, or continue to put an item of a selected type in the knapsack. An item's weight is learned when placed to the knapsack. The objective of this problem is to find a policy that maximizes the expected total values. Using the framework of dynamic programming, the optimal policy is found when $n = 2$ and a heuristic policy is suggested for $n > 2$.

1. Introduction

In the classic knapsack problem, given a set of items whose values and weights are deterministic, the objective is to find a subset of items to put in the knapsack in order to maximize the total values without incurring overflow. The knapsack problem and its variants have many applications in such areas as transportation scheduling, projects selection, resource allocation/management, and others.

This paper considers the zero return if broken (ZRB) knapsack problem. In ZRB knapsack problem, an item's weight is unknown before being put in but follows a known distribution; its value per unit weight is given and determined by the item's type. It's assumed that once the knapsack is broken from breaching the capacity constraint, all the existing items in the knapsack are wiped out without any salvage value, i.e., no additional items can be inserted into the now empty knapsack and we stop there. The ZRB knapsack problem is an adaptive stochastic knapsack problem, for which a policy is defined as a schedule to put in items sequentially which adapts to the information feedback on the updated system state. The objective is to find a policy which maximizes the expected total return.

The ZRB knapsack problem can be applied in the situations where breaking knapsack triggers the wipeout effect, e.g., over-utilization of the credit line freezes account actions; medicine overdosing negates the desired function, etc. Another application is in space exploration where loading over the capacity limit of a space craft leads to total lost of all on-board cargos. The ZRB knapsack problem is also important as its optimal expected return provides a lower bound on the optimal expected return in any adaptive knapsack problem which yields the same return as in our model when stopping occurs before the knapsack reaches capacity but where the return when the knapsack's capacity is exceeded is any arbitrary nonnegative function of the sequence of types and values of the items in the knapsack.

1.1. Literature review

The 0–1 knapsack problem (see Kellerer, Pferschy, & Pisinger (2004)) is one of NP-hard problems including traveling salesman problems, integer programming, etc. Martello, Pisinger, and Toth (2000) give a comprehensive review with further discussions on techniques commonly used in solving the knapsack problem. A stochastic knapsack problem (SKP) differs from the classic model by allowing randomness in the candidate items' weights or values (or both). According to how we assign items to the knapsack, there exist two categories of SKP: static and adaptive.

In a static SKP, the only decision made is to choose a subset of items that are simultaneously put in the knapsack. Kosuch and Lisser (2010, 2011) discuss a static SKP which assumes that a cost proportional to the overflow (that is, the amount by which the sum of the weights of the items put into the knapsack exceed its capacity) is incurred whenever the knapsack's capacity is exceeded, as well as one with a constraint on the probability of exceeding the capacity. They propose methods for locating upper and lower bounds to complement the branch and bound search. Under the assumption of normal distributions on items weights, Cohn and

☆ This material is based upon work supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant Number W911NF-11-1-0115.

* Corresponding author. Tel.: +1 2133001118.
E-mail addresses: kaic@usc.edu (K. Chen), smross@usc.edu (S.M. Ross).

http://dx.doi.org/10.1016/j.ejor.2014.06.027
0377-2217/

Mit (1998) explore dominance rules among candidate items and illustrate a search algorithm based on this. Merzifonluoğlu, Geunes, and Romeijn (2012) extend the discussion to include a penalty cost for capacity overflow and a salvage value for unused capacity. With results from the study of a continuous relaxation of the problem, Merzifonluoglu et al. develop a customized branch and bound search for the optimal decision and a high-quality heuristic policy. Lee and Oh (1997) discuss the asymptotic property when the knapsack capacity increases, and they use the asymptotic value-to-capacity ratio to approximate the optimal solution for a large knapsack capacity.

In an adaptive SKP, the decision maker has the option to select an available item or to stop on each stage while taking into account the latest knapsack state information from the system feedback. Some adaptive SKP papers have avoided the possibility of a broken knapsack by assuming deterministic item weights; others have included penalty terms to account for any overcapacity amount; and others have imposed a chance constraint on the overflow probability. In a SKP that assumes random values but deterministic weights, Iravani, Ilhan, and Daskin (2011) discussed the adaptive target achievement problem where the objective is to maximize the probability of achieving a target total value. They give a heuristic policy with limited look-ahead capabilities and show that it performs well. Lu, Chiu, and Cox (1999) consider the project selection problem with a deadline where projects with unknown value but deterministic resource requirements arrive to the system one by one according to a stochastic process. They give a simple form of the optimal project acceptance rule, although it's usually computationally expensive to lay out the decision map. Lin, Lu, and Yao (2008) discuss a similar problem in revenue management, where offers with stochastic price and quantity information arrive at each tim e point, and the decision is whether to accept or decline the offer. They propose a class of switch-over policies and find the optimal one in the class which has asymptotic optimality as the problem size scales up. And they also apply the result in the dynamic/flexible pricing model. Van Slyke and Young (2000) study the finite horizon SKP and its applications in yield management. In SKP with random weights, Schilling (1994) gives results on the asymptotic optimal values. Ross and Tsang (1989) formulate the network admission problem in SKP and present an optimal static control by dynamic programming. Derman, Lieberman, and Ross (1978) consider a renewal decision problem that is equivalent to an adaptive SKP where the value of an item depends only on its type rather than being proportional to its weight, and where the problem continues until the knapsack is broken at which point a final return equal to the sum of the values of al l but the final item put in the knapsack is earned. Dean, Goemans, and Vondrdk (2004) study the benefit of adaptivity in the SKP with random weights where they assume the final overflowing item contributes no value. They bound the adaptivity gap, which measures the ratio of the optimal adaptive policy value to the optimal static policy value, to a factor of four; and they also devise a polynomial-time adaptive policy that approximates the optimal policy with a factor of $3+\varepsilon$ for any positive ε. Kleywegt and Papastavrou (1998, 2001) define a class of very comprehensive SKP, the dynamic stochastic knapsack problems (DSKP). It assumes items with unknown values and weights arrive to the system stochastically. The item's value and weight are revealed upon its arrival and the decision to accept or to reject has to be made. Penalty incurred by rejection, holding cost, salvage value of items, etc. are all incorporated in the DSKP. The structural results on the optimal policy for DSKP are given in the two papers.

1.2. Outline

In Section 2, the ZRB knapsack problem is defined and formulated mathematically under the dynamic programming framework. Preliminary notations are given in this part before proceed to explore the characteristics of the model structure. We show a type preference order and demonstrate the optimal stopping rule in the latter part of the section. In Section 3, we discuss the problem when $n=2$ and propose an optimal policy in this case. The general ideas behind the optimality proof are given alongside supporting propositions which lead to the core theorem in this section. We also summarize into an easy-to-implement action selection strategy from the optimal policy for $n=2$. In Section 4, we try to generalize the optimal policy for $n=2$ to a policy that's applicable for any n using the same logic. We evaluate the generalized policy and analyze its limitations. A second heuristic policy for general n is then given and tested in a numerical example. We conclude the paper with a brief introduction of our ongoing work in Section 5.

In this paper, we define the indicator function $I_A=1$ if the event A occurs, otherwise $I_A=0$. We put the proofs which involve mainly algebraic manipulations in the Appendix A.

2. Problem setting

Consider a knapsack with a deterministic capacity w. There are n different types of items available to be put to the knapsack, and each type has infinite supply of items. A type i item, $1\leqslant i\leqslant n$, has value v_iW_i, where v_i is a deterministic positive value and W_i is the item's weight where $W_i\sim Exp(w_i)$ and w_i is the mean weight. It is assumed that an item's weight is independent with the weights of other items both within the same type and between types. At each stage, we can either choose to stop and leave the system with all the existing values in the knapsack, or, we can choose to select an item of any type to put to the knapsack. An item's weight is immediately revealed after its being put to the knapsack. If the knapsack is broken, because total weights of items in the knapsack exceed the capacity, we are forced out of the system with no return at all. Otherwise, we move to the next stage. We call the model defined above as the zero return if broken (ZRB) knapsack problem. The objective of the ZRB knapsack problem is to find a policy that achieves the maximal expected return.

2.1. Dynamic programming framework

The ZRB knapsack problem can be formulated in a dynamic programming framework. Let (r,v) be the state variable of the model where r is the remaining capacity and v is the total values of items in the knapsack. Let $V(r,v)$ be the optimal expected value function at state (r,v).

Optimality Equations where $\lambda_i=\frac{1}{w_i},\ \forall i\in[1,n]$,

$$V(r,v)=\max\left\{v,\max_{i=1,\dots,n}\left\{\int_0^r\lambda_i e^{-\lambda_i t}V(r-t,v+v_i t)dt\right\}\right\}$$
$$V(r,v)=0,\text{if } r<0. \tag{1}$$

Given n types of different items, we first want to discard those types which will never be used by an optimal policy.

Proposition 1. *If $v_i<v_j,\ w_i>w_j$, then a type i item should never be used.*

Proof. The idea of the proof comes from Smith (1978). We construct a composite component which consists of N type j items, where N is a geometric distributed random variable with parameter w_j/w_i. It is easy to see this composite component has weight that is exponentially distributed with mean $\frac{w_j}{w_j/w_i}=w_i$. Since $v_i<v_j$, it is always better to replace type i item with this composite component because the composite item has higher unit weight value than type i item does, and at the same time the weight of the