Operations Research Letters 42 (2014) 367–373

Contents lists available at ScienceDirect

Operations Research Letters

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

A continuous knapsack problem with separable convex utilities: Approximation algorithms and applications

Retsef Levi, Georgia Perakis *, Gonzalo Romero

Sloan School of Management, Massachusetts Institute of Technology, E62-565, Cambridge, MA 02142, United States

ARTICLE INFO

Article history:
Received 6 September 2013
Received in revised form
12 June 2014
Accepted 13 June 2014
Available online 20 June 2014

Keywords:
Knapsack
Subsidies
Cournot competition

ABSTRACT

We study a continuous knapsack problem with separable convex utilities. We show that the problem is NP-hard, and provide two simple algorithms that have worst-case performance guarantees. We consider as an application a novel subsidy allocation problem in the presence of market competition, subject to a budget constraint and upper bounds on the amount allocated to each firm, where the objective is to minimize the market price of a good.

1. Introduction

We study a continuous knapsack problem, where the objective is to maximize the sum of separable convex utility functions. We denote this problem by (CKP). Beyond general methods for concave minimization, see for example [1], there is not much literature on this class of problems. An exception is [10], and their algorithm to find local minima. A comprehensive review of the related nonlinear knapsack problem literature is presented in [3]; however, in most cases, the objective function considered in this literature is concave. On the other hand, for any given tolerance $\epsilon > 0$, a fully polynomial time approximation scheme (FPTAS) is an algorithm that generates a solution which is within a factor $(1-\epsilon)$ of being optimal, while the running time of the algorithm is polynomial in the problem size and $1/\epsilon$. Burke et al. [4] provide a tailored FPTAS for a minimization variant of a continuous knapsack problem, in the context of allocating procurement to suppliers. The knapsack problem we study here is a maximization problem, hence the results from [4] do not apply. Finally, [7] develop a general purpose FPTAS for a class of stochastic dynamic programs, which applies to general nonlinear knapsack problems. In contrast, our main goal in this paper is to study the performance of simple algorithms for problem (CKP), as well as to introduce a novel application of continuous knapsack problems into a subsidy allocation problem in the presence of market competition, subject to a budget constraint and upper bounds on the amount allocated to each firm, where the objective is to minimize the market price of a good.

The main contributions of this paper are two-fold. First, we develop two algorithms that are computationally and conceptually simple, such that they can be used in practical applications. We show that these algorithms have good worst-case performance guarantees for problem (CKP). Moreover, we identify special settings where these simple policies are actually optimal. Second, we show that problem (CKP) characterizes a novel subsidy allocation problem, and that the simple algorithms that we develop admit a practical interpretation.

2. Problem formulation

Consider n items indexed by $i \in \{1, \ldots, n\}$. For each i, let x_i be the non-negative quantity of item i, and let $f_i(x_i)$ be the resulting reward. Moreover, $f_i(x_i)$ is assumed to be convex. The quantity of item i cannot exceed a given upper bound u_i, and the total amount of all items is bounded by the capacity of the knapsack, denoted by B. Moreover, both B and u_i are assumed to be integers. We are interested in the following continuous knapsack problem

$$\text{(CKP)} \quad \begin{aligned} \max \quad & F(x) \equiv \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \le B \\ & 0 \le x_i \le u_i \ \forall\, i. \end{aligned}$$

The objective function is convex over the feasible set, which is a bounded polyhedron. Therefore, the existence of an extreme point

* Corresponding author.
E-mail addresses: retsef@mit.edu (R. Levi), georgiap@mit.edu (G. Perakis), gromeroy@mit.edu (G. Romero).

http://dx.doi.org/10.1016/j.orl.2014.06.007
0167-6377/

optimal solution follows from concave minimization theory, see for example [1].

The next one is our first result

Proposition 1. *Problem (CKP) is NP-hard.*

Proof. The proof is a reduction from the subset sum problem, which is well known to be NP-complete, see [8].

Consider an arbitrary instance of the subset sum problem, where given a set of n positive integers $\{u_1, u_2, \ldots, u_n\}$, and a positive integer B, the question is whether there exists a subset $J \subseteq \{u_1, u_2, \ldots, u_n\}$ that sums to B.

Now consider the following instance of problem (CKP): let u_i be the upper bound on x_i for each item i, B be the capacity of the knapsack and $f(x_i) = x_i(x_i - u_i) + x_i$ be the convex reward for each item i. It follows that this instance of problem (CKP) can be written as

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i(u_i - x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \le B \\ & 0 \le x_i \le u_i \ \forall\, i \in \{1, \ldots, n\}. \end{aligned}$$

Note that B is an upper bound on the optimal objective value of this problem. Moreover, this upper bound is attained if and only if there exists a subset $J \subseteq \{u_1, u_2, \ldots, u_n\}$ that sums to B.

Hence, if we can solve problem (CKP) in polynomial time, it follows that we can solve the subset sum problem in polynomial time.

The proof of Proposition 1 is in the same spirit of [12], who shows the NP-hardness of non-convex quadratic programming, among other problems.

We now make a couple of remarks that will make the rest of the exposition clearer.

Remark 1. There is no loss of generality in assuming that, for each $i \in \{1, \ldots, n\}$, the functions $f_i(x_i)$ are positive and non decreasing.

Specifically, we can pre-process the problem data replacing $f_i(x_i)$ by the amount $\max\{f_i(x_i), f_i(0)\}$, for each i and x_i, obtaining non decreasing functions without changing the problem. Similarly, by adding a constant $K > \min_i\{f_i(0)\}$ to each of the functions $f_i(x_i)$ we obtain positive functions.

Remark 2. There is no loss of generality in assuming that, for each $i \in \{1, \ldots, n\}$, $u_i \le B$.

Specifically, if any upper bound u_i is larger than the capacity B, then it follows that any feasible solution will allocate at most B to item i. Hence, we can pre-process the data and replace u_i by $\min\{B, u_i\}$, for each i, without changing the problem.

Having established the NP-hardness of problem (CKP), we now focus on simple algorithms with a guaranteed performance, and their practical interpretation.

2.1. A simple 1/2-approximation algorithm

We next describe a 1/2-approximation algorithm for problem (CKP). Specifically, we will show that intuitive ideas perform well in this model. Namely, the best solution between (i) allocating the capacity greedily to the items with the *fastest rate of increase* in their utility function, and (ii) allocating the capacity greedily to the items with the *largest absolute increase* in their utility function, attains an objective value that is at most half the value of the optimal objective value.

This algorithm is a generalization of the well known 1/2-approximation algorithm for the 0/1 knapsack problem. The latter is attained by the best solution between greedily picking the objects by decreasing ratio of profit to size, and picking the most profitable object, see for example [15].

Consider first idea (i). We denote the resulting solution by $\mathbf{x}^{\text{rate}}$. Essentially, $\mathbf{x}^{\text{rate}}$ is the result of a greedy procedure with respect to $\frac{f_i(u_i) - f_i(0)}{u_i}$, which is the rate of increase in the utility function of item i, assuming that x_i is set to its upper bound.

Algorithm 1 Compute $\mathbf{x}^{\text{rate}}$

$\mathbf{x}^{\text{rate}} \Leftarrow \vec{0}$
Let $\lambda_i = \frac{f_i(u_i) - f_i(0)}{u_i}$, for each i
Sort indexes by decreasing λ_i
Find $\hat{i}$ s.t. $\sum_{i=1}^{\hat{i}-1} u_i \le B$ and $\sum_{i=1}^{\hat{i}} u_i > B$
$x_i^{\text{rate}} \Leftarrow u_i$, for each $i \le \hat{i}$
$x_{\hat{i}}^{\text{rate}} \Leftarrow B - \sum_{i=1}^{\hat{i}-1} u_i$

On the other hand, consider idea (ii). We denote the resulting feasible solution by $\mathbf{x}^{\max}$. Essentially, $\mathbf{x}^{\max}$ is the result of a greedy procedure with respect to $f_i(\min(u_i, \tilde{B}))$, which is the absolute increase in the utility function of item i, when allocating the minimum between the remaining capacity $\tilde{B}$, and its upper bound. In case of a tie, $f_i(0)$ is used as a tie-breaker.

Algorithm 2 Compute $\mathbf{x}^{\max}$

$\tilde{B} \Leftarrow B$
$\mathbf{x}^{\max} \Leftarrow \vec{0}$
while $\tilde{B} > 0$ **do**
  Let $S_1 = \left\{ i \mid f_i\left(\min\left(u_i, \tilde{B}\right)\right) \ge f_j\left(\min\left(u_j, \tilde{B}\right)\right), \text{ for each } j : \mathbf{x}_j^{\max} = 0 \right\}$
  Let $S_2 = \left\{ i \in S_1 \mid f_i(0) \le f_j(0), \text{ for each } j \in S_1 \right\}$
  Select $i \in S_2$
  $\tilde{B} \Leftarrow \tilde{B} - \min\left(u_i, \tilde{B}\right)$
  $x_i^{\max} \Leftarrow \min\left(u_i, \tilde{B}\right)$
end while

It is not hard to see that each algorithm, considered separately, can be made to perform arbitrarily bad. Examples drawn from a 0/1 knapsack problem are sufficient.

In order to show a worst-case performance guarantee for problem (CKP), we need an upper bound on its optimal objective value, as provided in the following proposition.

Proposition 2. *Let $\mathbf{x}^*$ be an optimal solution to problem (CKP). Algorithm 1 provides the following upper bound,*

$$F(\mathbf{x}^*) \le F(\mathbf{x}^{\text{rate}}) + f_{\hat{i}}(0) - f_{\hat{i}}(x_{\hat{i}}^{\text{rate}}) + \frac{f_{\hat{i}}(u_{\hat{i}}) - f_{\hat{i}}(0)}{u_{\hat{i}}} x_{\hat{i}}^{\text{rate}}$$

where $F(x) = \sum_{i=1}^{n} f_i(x_i)$.

Proof. Let us relax the knapsack constraint in problem (CKP) with an associated Lagrange multiplier λ, to obtain the following relaxed optimization problem,

$$\begin{aligned} \max \quad & \lambda B + \sum_{i=1}^{n} \left(f_i(x_i) - \lambda x_i\right) \\ \text{s.t.} \quad & 0 \le x_i \le u_i \ \forall\, i. \end{aligned}$$