Approximability of the Two-Stage Knapsack problem with discretely distributed weights

Stefanie Kosuch

Institutionen för datavetenskap (IDA), Linköpings Universitet, Sweden

Key words: stochastic knapsack problem, multidimensional knapsack, two-stage, non-approximability, approximation algorithms

1 Introduction and Mathematical Formulation

The knapsack problem is a widely studied combinatorial optimization problem. It consists in choosing among a set of items a subset such that the total weight of the chosen items respects a given weight restriction (the capacity of the knapsack) while the total reward of the chosen items is maximized. The most common applications arise in fields where some capacity has to be respected (storage, transport, packing, network optimization...) or where the decision maker has to handle limited resources (recourse allocation, cutting stock problems...). However, knapsack problems also serve as subproblems in less obvious fields of application such as cryptography or finance.
As in many applications the decision maker has to face uncertainty in the involved parameters, more and more studies are made on various settings of the *Stochastic Knapsack problem*, where some of the parameters are assumed to be random (i.e. not exactly known in the moment the (pre-)decision has to be made). In this paper we restrict our study to the case where the weights are assumed to be random. Moreover, we assume that the decision can be made in two stages: A *pre*-decision is made while the item weights are still unknown, i.e. the decision maker assigns some items to the knapsack without knowing their exact weights. In this *first stage* we obtain a certain reward for the added items. Then, once the weights of all items have come to be known, we can make a *corrective* decision (*second stage*): If additional items are added, the reward obtained for these items is smaller than it would have been in the first stage. And if items are removed, a penalty has to be paid that is naturally strictly greater than the received first-stage reward. The objective is to maximize the first-stage reward plus the expected second-stage gain,

Email address: stefanie.kosuch@liu.se (Stefanie Kosuch).
URL: http://kosuch.eu/stefanie/ (Stefanie Kosuch).

where the latter is composed of the reward obtained for added items minus the penalty paid for removed ones. We call the resulting problem *Two-Stage Knapsack problem*. Its mathematical formulation is as follows:

$$
\begin{aligned}
(TSKP) \quad & \max_{x \in \{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i + \mathbb{E}[\mathcal{Q}(x,\chi)] && \text{(1a)}\\
\text{s.t.} \quad & \mathcal{Q}(x,\chi) = \max_{y^+,y^- \in \{0,1\}^n} \sum_{i=1}^{n} \overline{r}_i y_i^+ - \sum_{i=1}^{n} d_i y_i^- && \text{(1b)}\\
& \quad \text{s.t. } y_i^+ \le 1 - x_i, \quad \forall\, i = 1,\dots,n, && \text{(1c)}\\
& \quad\quad y_i^- \le x_i, \quad \forall\, i = 1,\dots,n, && \text{(1d)}\\
& \quad\quad \sum_{i=1}^{n} (x_i + y_i^+ - y_i^-)\chi_i \le c. && \text{(1e)}
\end{aligned}
$$

where x is the first-stage decision vector and $r > 0$ the first-stage reward vector, both of dimension n. The weight of item i is represented by the random variable χ_i. The second-stage binary decision vector y^+ models the adding of items while the decision vector y^- models their removal. If item i is added in the second stage we receive a second-stage reward $\overline{r}_i < r_i$, and if it is removed we have to pay a penalty $d_i > r_i$. An item can only be added if it had not been added in the first stage (constraint (1c)) and removed if it has been added previously (constraint (1d)). In the end, the items (remaining) in the knapsack need to respect the knapsack capacity $c > 0$ (constraint (1e)).

To the best of our knowledge there have only been two previous studies of Two-Stage Knapsack problems: In [4] the authors study a Two-Stage Knapsack problem with probability constraint in the first stage where the item weights are assumed to be normally distributed. The main difficulty in this case arises from the question of how to evaluate the objective function exactly. The authors therefore propose upper and lower bounds and apply a branch-and-bound algorithm to search the first-stage solution space for the best such lower bound. In [2] the authors study Static as well as Two-Stage Quadratic Knapsack problems with chance-constraint. The authors assume a finite distribution for the weight vector which allows them to reformulate the studied problems in a deterministic equivalent form. The authors propose semi-definite relaxations to obtain good upper bounds in reasonable time.

As in [2] we assume in this paper that the weight vector only admits a finite number of outcomes (*scenarios*) with non-zero probabilities. This allows to reformulate the $TSKP$ deterministically (see e.g. [2]). In fact, in [3] it has been shown that a stochastic combinatorial optimization problem can, under some mild assumptions, be approximated to any desired precision by replacing the underlying distribution by a finite random sample. However, to obtain a good approximation the used set of random samples needs generally to be rather large. Moreover, if the weights are e.g. independently, discretely distributed, the number of scenarios might grow exponentially with the number of items. Solving the obtained deterministic equivalent problem to optimality is thus

generally not tractable. That is why we are interested in the approximability of the $TSKP$ with discretely distributed weights. Note that, like its deterministic counterpart, the Two-Stage Knapsack problem is $\mathcal{NP}$-hard. Moreover, it has been shown in [1] that two-stage stochastic integer programming problems with discretely distributed parameters are even $\sharp\mathcal{P}$-hard.
In the second section we state a non-approximability result for the $TSKP$ and give a sketch of the proof. This is followed by three positive approximation results.

2 (Non)-Approximation results

Theorem 2.1 *For any $\epsilon > 0$, there exists no polynomial-time $K^{-\frac{1}{2}+\epsilon}$ - approximation algorithm for the $TSKP$, unless $\mathcal{P} = \mathcal{NP}$.*

Sketch of the proof: The idea of the proof is as follows: Basically we do a reduction from the multi-dimensional knapsack problem ($MDKP$). In [5] the authors prove that, for all $\epsilon > 0$, the $MDKP$ does not admit a polynomial-time $m^{-\frac{1}{4}+\epsilon}$-approximation algorithm (where m is the number of constraints) unless $\mathcal{P} = \mathcal{NP}$. This is proven by a reduction from the maximum clique problem. In their paper the authors use that the maximum clique problem cannot be approximated within a factor $n^{-\frac{1}{2}+\epsilon}$, for any $\epsilon > 0$, where n stands for the number of vertices. A newer result however states that it is even $\mathcal{NP}$-hard to approximate the maximum clique problem within a factor $n^{-1+\epsilon}$.
Instead of giving a direct reduction from the $MDKP$ to the $TSKP$, we first show how the optimal solution value of the $MDKP$ can be obtained by solving a special variant of the $TSKP$ where items can only be added in the second-stage (called *AddTSKP*). Note that this is not done by an equivalent reformulation: In fact, the reduction is such that the integer part of the solution value of the *AddTSKP* instance gives us the optimal solution value of the initial $MDKP$ instance. However, the optimal first-stage solution of the former is optimal solution for the latter. The number of scenarios in the obtained *AddTSKP* instance equals the number of constraints of the $MDKP$. In the second step we show that for any instance of the *AddTSKP* there exists an instance of the $TSKP$ with same number of scenarios, identical optimal solution value and such that an optimal solution of the $TSKP$ instance is optimal solution of the *AddTSKP* instance, and vice versa. The last step consists in proving that these polynomial reductions preserve the non-approximability result for the $MDKP$. □

Proposition 2.2 *For an instance of the $TSKP$ define $\alpha := \min_{i=1...,n} \frac{\overline{r}_i}{r_i}$. Then adding no items in the first stage yields a solution whose solution value is at least an α-fraction of the optimal solution value.*

Idea of the proof: First of all note that $\alpha < 1$. The idea is to first replace, for all i, the second-stage reward $\overline{r}_i$ by $\alpha \cdot r_i$. The optimal solution value of the new instance is thus a lower bound on the optimal solution value of the initial instance. Then adding no item at all in the first stage yields a solution for the obtained instance with approximation ratio at most α. This approximation ratio is thus also valid for the initial instance.

Proposition 2.3 *Under the assumption of a polynomial scenario model, there exists a polynomial-time $\frac{1}{n}$-approximation algorithm for the TSKP.*
Underlying algorithm: For all $i = 1, \ldots, n$ let R_i denote the maximum expected reward we can obtain for item i. Let $\mathcal{K}_i = \{k \in \{1, \ldots, K\} : \chi_i^k \leq c\}$ where K is the number of scenarios and χ_i^k $(k = 1, \ldots, n)$ are the outcomes of the random variable χ_i. Let $p^k > 0$ be the probability of scenario k. It follows that $R_i = \max\{r_i - \sum_{k \notin \mathcal{K}_i} p^k d_i, \sum_{k \in \mathcal{K}_i} p^k \overline{r}_i\}$. Let $j = \arg\max_{i=1,\ldots,n} R_i$. If $R_j = r_j - \sum_{k \notin \mathcal{K}_j} p^k d_j$, set $x_j = 1$ and $x_i = 0$ for all $i \neq j$, otherwise set $x_i = 0$ for all $i = 1, \ldots, n$. This clearly yields a solution with approximation ratio $\frac{1}{n}$. However, in order to determine j in polynomial time, K needs to be polynomial in n.

Proposition 2.4 *Let K-AddTSKP (K-MDKP) denote the variant of the AddTSKP (MDKP) where the number of scenarios (constraints) is fixed to be K. Then, for a given $\epsilon > 0$, there exists a polynomial-time approximation algorithm for the K-AddTSKP with approximation-ratio $\frac{1}{2} - \epsilon$.*
Idea of the underlying algorithm: First, solve the first-stage problem as a *K-MCKP* (i.e. the solution has to respect the K second-stage capacity constraints) using a *PTAS*. Then, solve independently the K second-stage knapsack problems using the well known *FTPAS* and compute the expectation of the obtained solution values (based on the corresponding probabilities of the scenarios). The associated solution is to add no item at all in the first stage. Compare the two solutions and output the better.

References

[1] Dyer, M., Stougie, L.: *Computational complexity of stochastic programming problems.* Mathematical Programming 106(3), 423–432 (2006)

[2] Gaivoronski, A.A., Lisser, A., Lopez, R., Hu, X.: *Knapsack problem with probability constraints.* Journal of Global Optimization 49(3), 397–413 (2010)

[3] Kleywegt, A.J., Shapiro A., Homem-de-Mello, T.: *The sample average approximation method for stochastic discrete optimization.* SIAM Journal on Optimization 12(2), 479–502 (2002)

[4] Kosuch, S., Lisser, A.: *On two-stage stochastic knapsack problems.* Discrete Applied Mathematics (In Press, Corrected Proof) (2010)

[5] Li'ang, Z., Yin, Z.: *Approximation for knapsack problems with multiple constraints.* Journal of Computer Science and Technology 14(4), 289–297 (1999)