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

Stefanie Kosuch

Institutionen för datavetenskap (IDA),Linköpings Universitet, SE-581 83 Linköping, Sweden

Abstract

In this paper the Two-Stage Knapsack problem with random weights is studied under the aspect of approximability. We assume finite probability distributions for the weights and show that, unless P=NP, the so obtained problem cannot be approximated in polynomial time within a better ratio than $K^{-1/2}$ (where K is the number of second-stage scenarios). We further study the special cases where in the second stage items can only be added or only be removed, but not both. Positive approximation results are given for three particular cases, namely linearly dependent first- and second-stage rewards, the polynomial scenario model and the case where the number of scenarios is assumed to be a constant. To the best of our knowledge, this is the first study of a two-stage knapsack problem under the aspect of approximability and the first time a non-approximability result has been proven for a stochastic knapsack problem of any kind.

Keywords: two-stage stochastic programming, stochastic knapsack problem, non-approximation, approximation algorithms

1. Introduction

The knapsack problem is a widely studied combinatorial optimization problem. Special interest arises from numerous real life applications for example in logistics, network optimization and scheduling. The basic problem consists in choosing a subset out of a given set of items such that the total weight (or size) of the subset does not exceed a given limit (the capacity of

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

the knapsack) and the total reward of the subset is maximized (for a survey on (deterministic) knapsack problems see the book by Kellerer et al. [1]).
However, most real life problems are non-deterministic in the sense that some of the parameters are not (exactly) known at the moment when the decision of which items to choose has to be made. For example, the available capacity might be unknown due to delays in previous jobs or the item rewards might depend on market fluctuations ([2],[3],[4]). One possibility to model and solve an optimization problem in the presence of uncertainty is to formulate it as a stochastic programming problem.
In this paper we study a stochastic knapsack problem where the item weights are random. Several variants of stochastic knapsack problems with random item weights (or sizes) have been studied so far and the interest seems still to increase. Among the publications recently released you can find papers on the simple-recourse knapsack problem ([5]), the chance-constrained knapsack problem ([6]), knapsack problems with recourse ([7]) as well as dynamic settings of the stochastic knapsack problem ([8]).
The problem studied in this work has a two-stage formulation. Two-stage optimization models (also known as optimization models with recourse) are important tools in decision making with uncertain parameters as in many cases a corrective decision can or even has to be made once the exact parameters are known. Therefore, they have been extensively studied since they were first introduced in the literature by Dantzig in 1955 (see [9]). For our stochastic knapsack model we assume that in the first stage, while the item weights are still unknown, a pre-decision of which items to include in the knapsack can be made. This decision can be corrected once all the item weights have come to be known. More precisely, items can be removed at a certain cost and/or additional items added, which naturally yields a smaller reward than in the first stage. We call the resulting problem the Two-Stage Knapsack problem with random weights ($TSKP$).
One can imagine various problems that could be modeled as $TSKP$s. In fact, this is true for any problem that in its deterministic form can be modeled as a knapsack problem and where one can think of cases where the weights are uncertain and thus a short-term correction of the decision might be needed once the weights are known for certain. For instance, take the example where the beds of a hotel complex or the seats of an airplane have to be filled, without knowing if there might be later cancellations or if an overbooking will lead to an excess of the capacity (see also section 2.1 where the former example will be given more in detail). In logistics one might have to schedule a fleet without knowing the exact sizes of the goods to be transported: If, in the end, there is some spare space, this might be filled

with supplementary items on short notice. In case of an overload, a penalty might have to be paid to the un- or not fully served customers.
In this study of the $TSKP$ we assume the random weight vector to be discretely distributed, i.e., to only admit a finite number of realizations with non-zero probability. In fact, Kleywegt et al. [10] have 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.
It is well known that in the case of finite weight distributions the $TSKP$ can be equivalently reformulated as a deterministic linear combinatorial programming problem (see e.g. [7]). However, it has been shown that two-stage problems with discrete distributions on (some of) the parameters are in general $\#P$-hard (even in the case of continuous decision variables, see [11]). Moreover, the number of constraints and binary decision variables in the deterministic reformulation grows with the number of scenarios and is thus typically very large, or even exponential in the number of items (e.g., if we assume the random weights to be independently distributed). Solving the problem to optimality is thus only possible in very restricted cases. The aim of this paper is therefore to study the $TSKP$ under the aspect of approximability. We show that, unless $\mathcal{P} = \mathcal{NP}$, the $TSKP$ cannot be approximated within $K^{-\frac{1}{2}+\epsilon}$ in polynomial time, where K is the number of scenarios and $\epsilon > 0$. This is remarkable insofar, as the deterministic knapsack problem admits a very simple $\frac{1}{2}$-approximation algorithm as well as a Fully Polynomial Time Approximation Scheme ($FPTAS$). To obtain this non-approximability result we first show a key property of the $TSKP$: The solution to any instance of the $TSKP$ can be obtained by solving an instance of a two-stage knapsack problem where, in the second stage, items can only be added (here called $AddTSKP$). The inverse is also true. This leads us to showing the mentioned key result for the $TSKP$ by reducing the well studied multiply-constrained knapsack problem with uniform capacities to the $AddTSKP$.
We also show that the deterministic reformulation of the $AddTSKP$ is in turn a multiply-constrained knapsack problem. Note however that, even if we assume the number of second-stage scenarios to be a constant, the deterministic reformulation of the $AddTSKP$ has a number of constraints that is linear in the number of items and can thus not be assumed to be a constant. To the best of our knowledge, applying any so far published approximation algorithm for the multiply-constrained knapsack problem (or its basic idea) directly to the $AddTSKP$ does not yield a polynomial time algorithm for exactly this reason. Instead, we propose an approximation algorithm

with approximation ratio $\frac{1}{2} - \epsilon$ ($\epsilon \in (0, \frac{1}{2})$) where one has to approximately solve both a multiply-constrained knapsack problem and K (normal) knapsack problems. Moreover, we prove simple approximation algorithms for two more special cases of the $TSKP$ (namely linearly dependent first- and second-stage rewards and the polynomial scenario model).

In the last decade, several papers have been published that treat the question of approximability of combinatorial two-stage problems. Some of these results are positive in the sense that the stochastic version of the problem is not much harder to solve than its deterministic counterpart ([12],[13]). For other problems, the introduction of stochasticity increases the problem's complexity significantly (see [14], [15],[16],[17]) which can be due to one of the two following circumstances: In some cases the fact that we have at least two different possible second-stage scenarios is the reason for the increased complexity of the two-stage problem. In these cases the problem reduces to the deterministic counterpart in case the number of second-stage scenarios is 1. The $TSKP$ clearly falls in this category of problems. In other cases the combinatorial structure of the problem changes completely when introducing a second stage. These problems can be characterized by the property that even in the case of only one possible second-stage scenario the problem stays harder to solve or approximate than its deterministic counterpart. For more information on approximation of two-stage optimization problems see the surveys by Swamy and Shmoys [18], Stougie and van der Vlerk [19] or Immorlica et al. [20].

The $TSKP$ has, for instance, not been studied under the aspect of approximability. Moreover, to the best of our knowledge, we present in this paper the first non-approximability result for a stochastic knapsack problem of any kind.

Dean et al. [21] studied a stochastic knapsack problem where the weight of an item is not known until the item is placed in the knapsack. The authors do not make any assumption about the underlying probability distribution of the random parameters. The aim of their work is "to design a solution policy for sequentially inserting items until the capacity is eventually exceeded". They compare non-adaptive with adaptive policies (where in the latter case the choice of which item to insert next is made with respect to the already added items). The authors propose constant-factor approximation algorithms for both the adaptive and non-adaptive case. Their results have been recently improved (see [8]). Moreover, a new but much related problem has been studied where the knapsacks capacity is allowed to be extended by an arbitrary small amount. While in [21] and [8] items, once they have been inserted cannot be prematurely canceled any more, this is explicitly allowed

in the work of Gupta et al. [22]. Moreover, the authors allow the item weights to depend on the item sizes, which, as well, generalizes the earlier work. Their main result is that despite these generalization the problem can still be solved up to a constant approximation factor in polynomial time.
The main difference between the stochastic knapsack problem studied in [21], [8] and [22] and the $TSKP$ is that in the former we seek for an ordering of the items or a policy by which to add the items into the knapsack, while in the latter a feasible solution is a (generally proper) subset of items. The way these problems are addressed are thus quite different. Nonetheless, the authors of [22] develop an interesting idea that might also be of use for designing approximation algorithms for the $TSKP$ and will thus be briefly sketched in the conclusion of this paper.
Kleinberg et al. [23] considered a stochastic knapsack problem where the items are independent Bernoulli trials. The studied model is a chance-constrained optimization problem and the authors present an approximation algorithm whose approximation ratio depends logarithmically on the allowed probability of overflow. The paper also contains an approximation algorithm that finds a (better) near optimal solution for the case where the knapsack capacity is allowed to be perturbed by a small constant ϵ. Goel and Indyk [24] extended this work by proposing a Polynomial Time Approximation Scheme ($PTAS$)[1] for the chance-constrained knapsack problem where the item weights follow a Poisson or Exponential distribution as well as a Quasi-Polynomial Time Approximation Scheme for the case of Bernoulli trials. The first ones to present approximation results for the chance-constrained knapsack problem with arbitrary weight distributions are Bhalgat et al. in [8]: They show the existence of a $PTAS$ for the chance-constrained knapsack problem when both the knapsack capacity and the overflow probability are relaxed by a factor $(1 + \epsilon)$.
Both the type of stochastic knapsack problems studied e.g. in [21] and the chance-constrained knapsack problem differ from the $TSKP$ considered in this paper by the fact that in the $TSKP$ every item has two different rewards: The first- and the second-stage reward. This makes it difficult to apply the main idea of most approximation algorithms for deterministic or stochastic knapsack problems to the $TSKP$: Sort the items by their ratio of reward and weight (or a similar value) and then insert them following this

[1] In short, a $PTAS$ for a maximization problem is a scheme that, for any fixed $\epsilon > 0$, allows to compute in polynomial time a solution with solution value at least $(1 - \epsilon)$ times the optimal solution value.

order. For instance, the authors of [21] use an ordering of the items by the ratio of reward and mean size.
The remainder of this paper is organized as follows: In section 2 we give the mathematical formulation of the Two-Stage Knapsack problem considered. In subsection 2.1 one of the examples for an application briefly sketched in this introduction is specified and in subsection 2.2 some properties of the $TSKP$ are discussed. Among others we introduce the abovementioned special variant of the $TSKP$ where items can only be added in the second stage ($AddTSKP$). We show that for any instance of the $AddTSKP$ there exists an equivalent instance of the $TSKP$. In section 3 we present the multiply-constrained knapsack problem with uniform capacities and show that it is equivalent to the $AddTSKP$ in the sense that any instance of the former can be solved via an instance of the latter and vice versa. Section 4 contains the abovementioned non-approximability result, followed by a study of the special variant of the $TSKP$ where items can only be rejected in the second stage. The approximation algorithms under particular assumptions are given in section 5. In the last section before the conclusion (section 6) the special case of independently discretely distributed weights is discussed.

2. Mathematical Formulation

We consider a stochastic knapsack problem of the following form: Let a knapsack with fixed weight capacity $c > 0$ as well as a set of n items be given. Each item has a weight that is not known in the first stage but comes to be known before the second-stage decision has to be made. Therefore, we handle the weights as random variables and assume that the random weight vector χ is discretely distributed with K possible outcomes $\chi^1, \dots, \chi^K$. We denote the respective non-zero probabilities of these K *scenarios* by $p^1, \dots, p^K$. All weights are assumed to be strictly positive.
In the first stage, items can be placed in the knapsack (*first-stage items*). The corresponding first-stage decision vector is denoted $x \in \{0, 1\}^n$. Placing item i in the knapsack in the first stage results in a reward $r_i > 0$. At the beginning of the second stage, the weights of all items are revealed. First-stage items can now be removed and additional items be added (*second-stage items*) in order to make the capacity constraint be respected and/or increase the total reward. Note that the set of items available in the second stage is assumed to be a subset of those items that have already been available in the first stage and that no first-stage item can be re-added in the second stage.

In case of the removal of an item i, a penalty d_i has to be paid that is naturally strictly greater than the first-stage reward r_i. The removal of item i is modeled by the decision variable y_i^- that is set to 1 if the item is removed and to 0 otherwise. Similarly, we assume that the second-stage reward for this item $\overline{r}_i > 0$ is strictly smaller than its first-stage reward. If an item is added in the second stage we set the corresponding binary decision variable y_i^+ to 1. The resulting Two-Stage Knapsack problem with discrete weight distributions can be mathematically formulated as follows:

Two-Stage Knapsack problem with discretely distributed weights

$$
\begin{aligned}
(TSKP) \quad & \max_{x\in\{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \mathcal{Q}(x,\chi^k) \\
& \text{s.t.} \quad \mathcal{Q}(x,\chi) = \max_{y^+,y^-\in\{0,1\}^n} \sum_{i=1}^{n} \overline{r}_i y_i^+ - \sum_{i=1}^{n} d_i y_i^- \\
& \qquad\qquad \text{s.t.} \quad y_i^+ \leq 1 - x_i, \quad \forall\, i = 1,\dots,n, \\
& \qquad\qquad\qquad\; y_i^- \leq x_i, \quad \forall\, i = 1,\dots,n, \\
& \qquad\qquad\qquad\; \sum_{i=1}^{n} (x_i + y_i^+ - y_i^-)\chi_i \leq c.
\end{aligned}
$$

In this paper we present approximability and non-approximability results for the $TSKP$ and some special cases. We use the following definitions: Let Π be an optimization problem and let $\mathcal{I}$ be an instance of Π. Given an algorithm $\mathcal{A}$ for Π, let $\mathcal{A}(\mathcal{I})$ denote the objective function value of the solution returned by $\mathcal{A}$ on instance $\mathcal{I}$. Also, let $OPT(\mathcal{I})$ denote the optimal solution value for instance $\mathcal{I}$.

Definition 2.1. *An* approximation algorithm *for an optimization problem* Π *is a polynomial-time algorithm that solves near-optimally every instance of* Π. *An approximation algorithm* $\mathcal{A}$ *of a maximization problem* Π *has an* approximation ratio *of* $r(n)$ *and is called* $r(n)$-approximation algorithm *if the following condition holds for all instances* $\mathcal{I}$ *of* Π *of input size* n*:*

$$\frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \geq r(n)$$

Remark that for a maximization problem $\frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq 1$.

2.1. Example for an application of the Two-Stage Knapsack problem

As an application consider an (online) travel agency that aims to fill the vacant beds (the deterministic capacity) of a hotel complex. Clients are travel groups whose exact number of travelers (the "weight" of the group) is still unknown at the moment where the decision of which groups to accept has to be made. This randomness can for example be a result of later cancellations. If an upper bound on the sizes of the travel groups is known, the probability space for the weights is discrete and finite. However, the distributions of the group sizes might be dependent. In order to maximize the final occupancy of the beds, the travel agent might allow an overbooking. If, in the end, the number of beds is not sufficient, one or more of the groups need to be relocated in neighboring hotels which leads to a loss of benefit. If beds are left unoccupied, last minute offers at reduced prices might be an option to fill these vacancies.
A simple recourse version of this problem with a set of hotel sites has been previously considered in [25]. However, in this formulation it is assumed that in the case of an insufficient number of beds single travelers can be relocated, whereas in the two-stage formulation of the problem whole groups need to be installed in other hotels.

2.2. Properties of the TSKP

Property 1:. The $TSKP$ is an $\mathcal{NP}$-hard problem. This follows from the fact that in case of only one scenario in the second-stage the problem reduces to a deterministic knapsack problem (see also Property 3).

Property 2:. The $TSKP$ is a *relatively complete recourse* problem, i.e., for every feasible first-stage decision there exists a feasible second-stage decision.

Property 3:. Given a first-stage decision and a realization of χ, solving the second-stage problem means solving a deterministic knapsack problem: Let $S \subseteq \{1, \dots, n\}$ be the index set of the first-stage items, and $\overline{S} := \{1, \dots, n\} \setminus S$. Let us define a new second-stage decision vector $z \in \{0,1\}^n$: for $i \in S$ $z_i = 1$ indicates that item i is kept in the second stage and for $j \in \overline{S}$ we set $z_j = 1$ if and only if item j is a second-stage item. Then, in scenario k the second-stage problem consists in solving the following problem:

$$\begin{aligned} \max_{z \in \{0,1\}^n} \quad & \sum_{j \in \overline{S}} \overline{r}_j z_j - \sum_{i \in S} d_i (1 - z_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} z_i \chi_i^k \leq c. \end{aligned}$$

Defining $C := -\sum_{i\in S} d_i$, the objective can be rewritten as

$$\max \quad \sum_{i\in S} d_i z_i + \sum_{j\in\overline{S}} \overline{r}_j z_j + C$$

We therefore obtain a knapsack problem with reward d_i for the first-stage items (if kept) and reward $\overline{r}_j$ for an added second-stage item.

Property 4:. The $TSKP$ has a deterministic equivalent reformulation as a combinatorial optimization problem with linear objective and constraints: By introducing K copies of both the second-stage decision vector y^+ and the second-stage decision vector y^- (denoted $(y^+)^k$ and $(y^-)^k$, $k \in \{1,\ldots,K\}$, respectively) and treating the second-stage constraints for each second-stage scenario separately, one obtains the following reformulation:

Deterministic reformulation of the $TSKP$

$$\begin{aligned}
(TSK^D) \quad \max \quad & \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} \overline{r}_i (y^+)_i^k - \sum_{i=1}^{n} d_i (y^-)_i^k \right) \\
\text{s.t.} \quad & (y^+)_i^k \le 1 - x_i, \quad \forall\, i = 1,\ldots,n, \forall\, k = 1,\ldots,K, \\
& (y^-)_i^k \le x_i, \quad \forall\, i = 1,\ldots,n, \forall\, k = 1,\ldots,K, \\
& \sum_{i=1}^{n} (x_i + (y^+)_i^k - (y^-)_i^k)\chi_i^k \le c, \quad \forall\, k = 1,\ldots,K, \\
& x \in \{0,1\}^n, \\
& (y^+)^k, (y^-)^k \in \{0,1\}^n \quad \forall\, k = 1,\ldots,K.
\end{aligned}$$

However, the TSK^D has $n(2K+1)$ binary variables and $(2n+1)K$ constraints. Especially if the weights are independently distributed and if we assume K_i possible outcomes for weight χ_i, we obtain a total number of $K = \prod_{i=1}^{n} K_i \ge (\min_{i\in\{1,\ldots,n\}} K_i)^n$ scenarios, i.e., the number of scenarios is (generally) exponential in the number of items. So solving the TSK^D using common methods for linear programming might already be computationally cumbersome for relatively small $K_i \ge 2$.

Property 5:. Let *AddTSKP* denote the variant of the $TSKP$ where, in the second stage, items can only be added. The following proposition shows that this problem is a special case of the general $TSKP$:

Proposition 2.2. *For any instance of the AddTSKP there exists an instance of the $TSKP$ with identical optimal solution value and such that an* *optimal solution of the $TSKP$ instance is optimal solution of the Add$TSKP$ instance, and vice versa.*

Remark: Before proving the above proposition, let us remark that the Add$TSKP$ is not a relatively complete recourse problem, as there exist first-stage decisions that make the second-stage problem (in one or more scenarios) infeasible. For these solutions we define the corresponding objective function value to be $-\infty$. Such a solution is thus clearly not optimal as there always exists a solution of an instance of the Add$TSKP$ with objective function value greater or equal than 0.
The problem of second-stage infeasibility of the Add$TSKP$ could be arranged by adding K capacity constraints to the first stage.

Proof of Proposition 2.2. Let an instance of the Add$TSKP$ be given. We construct a corresponding instance of the $TSKP$ as follows: The n-dimensional parameter vectors $r, \overline{r}, \chi^k$, the probabilities p^k and the capacity c are carried over to the new problem. For all $i = 1, \dots, n$ the penalty d_i is set to $\max_{k \in \{1,\dots,K\}} \frac{r_i}{p^k} + 1$.
Let x^* be an optimal solution of the obtained instance of the $TSKP$ and let $(y^-)^k = (y^-)^k(x)$ be a corresponding optimal second-stage solution for y^- in scenario k. Assume that there exists $\widetilde{k} \in \{1, \dots, K\}$ and $i \in \{1, \dots, n\}$ such that $(y^-)_i^{\widetilde{k}} = 1$ (i.e., item i has been added in the first stage and rejected in scenario $\widetilde{k}$ in the second stage). Then, item i contributes at most

$$r_i - p^{\widetilde{k}} d_i = r_i - p^{\widetilde{k}} (\max_k \frac{r_i}{p^k} + 1) \leq r_i - r_i - p^{\widetilde{k}} < 0$$

to the objective function value. We are thus able to strictly increase the objective function value by setting $x_i = 0$, a contradiction. It follows that for any optimal (first-stage) solution of the constructed $TSKP$ instance all corresponding optimal second-stage solutions are such that $(y^-)_i^k = 0$ for all $k \in \{1, \dots, K\}$ and $i \in \{1, \dots, n\}$. An optimal solution of the constructed $TSKP$ instance is therefore feasible for the corresponding Add$TSKP$ instance, with identical objective function value. As any optimal solution of the Add$TSKP$ is, in turn, feasible for the $TSKP$ instance (with identical objective function value), the proposition is proved. □

Property 6:. In the TSK^D the knapsack capacity is assumed to be deterministic, i.e., identical in all scenarios. However, the formulation where the capacities are, as well, scenario dependent (with a finite number of outcomes) is in fact equivalent to the TSK^D as we could simply multiply the capacity constraint by an appropriate factor in each of the finitely many scenarios. Note that the fact that all outcomes of the capacity are known and that their number is finite is used here. In the case where the knapsack capacity is bounded from above but has an infinite number of possible outcomes, an equivalent reformulation with uniform (deterministic) capacity can still be obtained by introducing an additional item and setting the uniform capacity to a value strictly greater than the capacity's upper bound. The random weight of the additional item is defined as the difference between the newly defined, deterministic capacity and the initial, random capacity. Its outcomes are thus always strictly positive. The reward of the additional item is defined in a way that any optimal first-stage solution must contain this item.

3. Equivalence of the *AddTSKP* and the *MCKP*

The multiply-constrained knapsack problem $MCKP$ (sometimes also called multi-dimensional (vector) knapsack problem) with uniform capacities is generally defined as:

Multiply-Constrained Knapsack problem

$$
\begin{aligned}
(MCKP) \quad & \max_{\widetilde{x} \in \{0,1\}^{\widetilde{n}}} && \sum_{i=1}^{\widetilde{n}} \widetilde{r}_i \widetilde{x}_i \\
& \text{s.t.} && \sum_{i=1}^{\widetilde{n}} \widetilde{x}_i \widetilde{w}_i^j \leq \widetilde{c} \quad \forall\, j = 1, \ldots, m,
\end{aligned}
$$

where $\widetilde{w}_i^j \geq 0$ and $\widetilde{c}, \widetilde{r}_i > 0$ for all $i \in \{1, \ldots, \widetilde{n}\}$, $j \in \{1, \ldots, m\}$.

3.1. Formulation of the $AddTSK^D$ as an $MCKP$

The $AddTSK^D$ (i.e., the deterministic equivalent formulation of the $AddTSKP$) can be stated as follows:

$$
\begin{aligned}
(AddTSK^D) \quad & \max_{x \in \{0,1\}^n} && \sum_{i=1}^{n} r_i x_i + \sum_{k=1}^{K} p^k \sum_{i=1}^{n} \overline{r}_i (y_i^+)^k && \text{(1a)} \\
& \text{s.t.} && (y_i^+)^k + x_i \leq 1 \quad \forall\, i = 1, ..., n,\ k = 1, ..., K, && \text{(1b)} \\
& && \sum_{i=1}^{n} (x_i + (y_i^+)^k) \chi_i^k \leq c. &&
\end{aligned}
$$

Problem (1) is clearly a multiply-constrained knapsack problem (with non-uniform capacities) with strictly positive rewards and capacities and non-negative weights. Multiplying constraints (1b) by $c > 0$ would turn it in an $MCKP$.
Note that the number of constraints in this reformulation of the $AddTSKP$ depends on n and can thus not be assumed constant. It is therefore not possible to (directly) apply the $PTAS$ for the $MCKP$ (see e.g. [26]) to this reformulation as its running time depends exponentially on n. In section 5.3 we however propose an approximation algorithm with ratio $\frac{1}{2} - \epsilon$ and running time polynomial in n.

3.2. Solving an instance of the $MCKP$ via an instance of the $AddTSK^D$

If we do not allow the rewards to be zero or negative, it is in general not possible to equivalently reformulate the $MCKP$ as an $AddTSKP$. Nevertheless, it is possible to obtain an optimal solution for an instance of the $MCKP$ by solving an instance of the $AddTSKP$. This is the subject of the next proposition:

Proposition 3.1. *Let an instance of the $MCKP$ be given. Then, there exists an instance of the $AddTSKP$ such that any optimal (first-stage) solution of the latter is an optimal solution to the former.*

Proof. W.l.o.g. we assume the reward vector r of the given instance of the $MCKP$ to be integer and construct a corresponding instance of the $AddTSKP$ as follows:
The knapsack capacity is $\widetilde{c}$. The first-stage reward of item i is $\widetilde{r}_i$. There are m second-stage scenarios. The weight of item i in scenario j is set to $\widetilde{w}_i^j$. The probabilities of the scenarios are uniformly set to $\frac{1}{m}$ and the second-stage rewards are fixed at $\frac{1}{\widetilde{n}+1}$ where $\widetilde{n}$ is the number of available items for the given instance of the $MCKP$.
Let x^* be an optimal solution of the constructed instance of the $AddTSKP$. Let v^* be the corresponding solution value. It is easy to see that $\lfloor v^* \rfloor$ is the reward provided by the items added in the first stage, as the first-stage solution value is always integer and the contribution of adding items in the second stage is at most

$$\sum_{j=1}^{m} \frac{1}{m} \sum_{i=1}^{\widetilde{n}} \frac{1}{\widetilde{n}+1} = \frac{\widetilde{n}}{\widetilde{n}+1}$$

which is strictly smaller than 1. Moreover, the vector x^* is feasible for the initial instance of the $MCKP$ with objective function value $\lfloor v^* \rfloor$ as for the

AddTSKP any *optimal* first-stage solution is always second-stage feasible. $\lfloor v^* \rfloor$ is thus a lower bound on the optimal solution value of the given $MCKP$ instance.
Let us assume that there exists a solution $\widetilde{x}^*$ of the given $MCKP$ instance with objective function value $\widetilde{v}^* > \lfloor v^* \rfloor$. As the optimal solution of the $MCKP$ instance is integer, we especially have $\widetilde{v}^* \geq \lfloor v^* \rfloor + 1 > v^*$. As $\widetilde{x}^*$ is of course feasible for the constructed instance of the *AddTSKP*, we have a contradiction.
It follows that any optimal first-stage solution of the constructed *AddTSKP* instance is optimal for the given $MCKP$ instance. Moreover, the optimal solution value of the latter is given by the integer part of the optimal solution value of the former. □

4. Non-approximability results for the *TSKP* and some special cases

In [27] the authors prove that, for all $\epsilon > 0$, the multiply-constrained knapsack problem with non-uniform capacities does not admit a $m^{-\frac{1}{4}+\epsilon}$-approximation algorithm (where m is the number of constraints) unless $\mathcal{P} = \mathcal{NP}$. The authors prove this by a reduction from the maximum clique problem that can be formulated as a multiply-constrained knapsack problem with capacity 1 in each constraint. Their proof is thus directly applicable to the $MCKP$ as well. In their paper the authors use that, for any $\epsilon > 0$, the maximum clique problem cannot be approximated in polynomial time within a factor $n^{-\frac{1}{2}+\epsilon}$ (where n is the number of vertices in the considered graph). A newer result however states that it is even $\mathcal{NP}$-hard to approximate the maximum clique problem within a factor $n^{-1+\epsilon}$ (see [28]). It follows, that the multiply-constrained knapsack problem with non-uniform capacities does not admit a $m^{-\frac{1}{2}+\epsilon}$-approximation algorithm, for any $\epsilon > 0$ (unless $\mathcal{P} = \mathcal{NP}$).
Based on this, we can now show the following:

Theorem 4.1. *For any $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the $TSKP$, unless $\mathcal{P} = \mathcal{NP}$.*

Proof. Due to the $\mathcal{NP}$-hardness of the $TSKP$ the statement is obvious for $\epsilon \geq \frac{1}{2}$. So let $\epsilon \in (0, \frac{1}{2})$ be given. Assume that $\mathcal{P} \neq \mathcal{NP}$ and that there exists an algorithm $\mathcal{A}$ such that $\mathcal{A}$ finds, in polynomial time, a solution to any instance of the $TSKP$ with worst case ratio $K^{-\frac{1}{2}+\epsilon}$ (where K is the number of scenarios in the given $TSKP$ instance).

Let an instance of the $MCKP$ be given. Like in the proof of Proposition 3.1 we assume the rewards to be integer. Then, we multiply the objective function by $m^{\frac{1}{2}-\epsilon}$ such that the $m^{-\frac{1}{2}+\epsilon}$-fraction of any optimal solution of the given $MCKP$ instance becomes integer.
As shown in subsection 3 we can find an optimal solution and the solution value of this $MCKP$ instance by solving a corresponding instance of the $AddTSKP$. Note that the number m of constraints of the $MCKP$ equals the number of second-stage scenarios in the obtained instance of the $AddTSKP$. The $AddTSKP$ can in turn be equivalently reformulated as an instance of the $TSKP$. Applying algorithm $\mathcal{A}$ to the latter gives us therefore an approximate solution of the constructed instance of the $AddTSKP$ with worst case ratio $m^{-\frac{1}{2}+\epsilon}$.
Let $v^{\mathcal{A}}$ be the obtained solution value and v^* the optimal solution value of the $AddTSKP$, i.e., we have $\frac{v^{\mathcal{A}}}{v^*} \geq m^{-\frac{1}{2}+\epsilon}$. Recall that by the construction in the proof of Proposition 3.1 the integer part of the optimal solution value of the $AddTSKP$ gives us the optimal solution value of the initial $MCKP$. It follows that $m^{-\frac{1}{2}+\epsilon}\lfloor v^* \rfloor$ is integer. We have:

$$\begin{aligned}
\lfloor v^{\mathcal{A}} \rfloor + (v^{\mathcal{A}} \mod \mathbb{Z}) &= v^{\mathcal{A}} \\
&\geq m^{-\frac{1}{2}+\epsilon} \left(\lfloor v^* \rfloor + (v^* \mod \mathbb{Z}) \right) \\
&= \lfloor m^{-\frac{1}{2}+\epsilon} v^* \rfloor + m^{-\frac{1}{2}+\epsilon} (v^* \mod \mathbb{Z}) \\
&\Longrightarrow \lfloor v^{\mathcal{A}} \rfloor \geq m^{-\frac{1}{2}+\epsilon} \lfloor v^* \rfloor
\end{aligned}$$

Solving the $TSKP$ by algorithm $\mathcal{A}$ thus gives us a solution of the constructed instance of the $AddTSKP$ whose objective function value is at least a $m^{-\frac{1}{2}+\epsilon}$ fraction of the optimal solution value of the initial $MCKP$ instance. As all problem reformulations can be made in a number of steps polynomial in the input size of the given $MCKP$ instance, we get a contradiction to the fact that the $MCKP$ cannot be approximated within a factor $m^{-\frac{1}{2}+\epsilon}$ unless $\mathcal{P} \neq \mathcal{NP}$. This terminates the proof. □

From the proof of the previous theorem we immediately get the following corollary:

Corollary 4.2. *For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the $AddTSKP$, unless $\mathcal{P} = \mathcal{NP}$.*

As the TSK^D defined in this chapter is a special case of the Two-Stage Knapsack problem with scenario dependent capacities, we also have:

Corollary 4.3. *For any $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the Two-Stage Knapsack problem with scenario dependent capacities.*

4.1. The special case where items can only be rejected in the second stage

Let $RejTSK^D$ be the variant of the TSK^D where, in the second stage, items can only be rejected. Similar to the case of the $AddTSK^D$ (see subsection 3) it can be shown that any instance of the $RejTSK^D$ can be solved via a corresponding instance of the $MCKP$ and vice versa.

Proposition 4.4. *Any instance of the $RejTSK^D$ can be equivalently reformulated as an instance of the $MCKP$.*

Proof. Let an instance of the $RejTSK^D$ be given. First of all we redefine the first and second-stage decision variables:

- First-stage decision vector $\widetilde{x}$: $\widetilde{x}_i = 1$ if and only if item i *is not* added in the first stage
- Second-stage decision vector $(\widetilde{y})^k$: $(\widetilde{y})^k_i = 1$ if and only if item i has been added in the first stage and is kept in the second stage in scenario k (i.e., if and only if item i is in the knapsack after the second-stage choice has been made)

There are three possible cases for an item i in scenario k:

- Item i is added in the first and kept in the second stage, i.e., $\widetilde{x}_i = 0$ and $(\widetilde{y})^k_i = 1$. In this case item i contributes r_i to the total reward.
- Item i is added in the first and rejected in the second stage, i.e., $\widetilde{x}_i = (\widetilde{y})^k_i = 0$. In this case item i contributes $-(d_i - r_i)$ to the total reward (in scenario k).
- Item i is not added in the first stage, i.e., $x_i = 1$ and $(\widetilde{y})^k_i = 0$. In this case item i does not contribute at all to the total reward (in any scenario).

Remark that $x_i = (\widetilde{y})^k_i = 1$ is not possible.

Based on these observations, the $RejTSK^D$ instance can be reformulated as follows:

$$\max \quad \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} r_i \widetilde{y}_i^k - \sum_{i=1}^{n} (d_i - r_i)(1 - \widetilde{x}_i - \widetilde{y}_i^k) \right) \tag{2a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \widetilde{y}_i^k \chi_i^k \leq c \quad \forall\, k = 1, \ldots, K,$$

$$\widetilde{x}_i + \widetilde{y}_i^k \leq 1 \quad \forall\, i = 1, \ldots, n, \forall\, k = 1, \ldots, K, \tag{2b}$$

$$\widetilde{x} \in \{0,1\}^n,$$

$$\widetilde{y}^k \in \{0,1\}^n \quad \forall\, k = 1, \ldots, K.$$

After removing the constant term in the objective function and multiplying the $n \cdot K$ constraints (2b) by c, we obtain the following multiply-constrained knapsack problem with nonnegative weights, strictly positive rewards and uniform, strictly positive capacities:

$$\max \quad \sum_{k=1}^{K} p^k \left(\sum_{i=1}^{n} r_i \widetilde{y}_i^k + \sum_{i=1}^{n} (d_i - r_i)(\widetilde{x}_i + \widetilde{y}_i^k) \right)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \widetilde{y}_i^k \chi_i^k \leq c, \quad \forall\, k = 1, \ldots, K,$$

$$c\widetilde{x}_i + c\widetilde{y}_i^k \leq c, \quad \forall\, i = 1, \ldots, n, \forall\, k = 1, \ldots, K,$$

$$\widetilde{x} \in \{0,1\}^n,$$

$$\widetilde{y}^k \in \{0,1\}^n \qquad \forall\, k = 1, \ldots, K.$$

□

Contrary to the $AddTSK^D$ the $RejTSK^D$ is even equivalent to the $MCKP$ as any instance of the $MCKP$ can, as well, be equivalently reformulated an instance of the $RejTSK^D$: It is sufficient to set the second-stage penalties high enough, i.e., such that rejecting an item in the second stage is never optimal:

Proposition 4.5. *Any instance of the $MCKP$ can be equivalently reformulated an instance of the $RejTSK^D$.*

Proof. Let an instance of the $MCKP$ be given. We construct an equivalent instance of the $RejTSK^D$ having the following parameters:

- $p^k = \frac{1}{m}$ for all $k = 1, \ldots, m$

- $r_i := \widetilde{r}_i$ for all $i = 1, \ldots, \widetilde{n}$
- $d_i^k := m \cdot r_i + 1$ for all $i = 1, \ldots, \widetilde{n}$ and $k = 1, \ldots, m$
- $w_i^k := \widetilde{w}_i^k$ for all $i = 1, \ldots, \widetilde{n}$ and $k = 1, \ldots, m$
- $c = \widetilde{c}$

Clearly, any solution of the initial $MCKP$ instance is feasible first-stage solution for the constructed $RejTSK^D$ instance with identical (overall) objective function value (as no item needs to be rejected in any scenario). In turn, any first-stage solution of the $RejTSK^D$ instance satisfies the constraints of the $MCKP$ if no item needs to be rejected in any of the K second-stage scenarios. The objective function values are once more the same. It thus remains to prove the following claim:

Claim 4.6. *Let $(x_1^*, \ldots, x_{\widetilde{n}}^*)$ be an optimal first-stage solution of the constructed $RejTSK^D$ instance and let $(y^1)^*, \ldots, (y^{\widetilde{n}})^*$ be a corresponding optimal second-stage solution. Then $(y^k)_i^* = 0$ for all $i = 1, \ldots, \widetilde{n}$ and $k = 1, \ldots, m$.*

Proof of the claim. Let v^* be the optimal solution value of the constructed $RejTSK^D$ instance. Suppose there exists $i \in \{1, \ldots, \widetilde{n}\}$ and at least one scenario h with $(y^h)_i^* = 1$. In this case item i contributes at most $r_i - \frac{1}{m}(m \cdot r_i + 1) = -\frac{1}{m}$ to the total expected reward. Therefore, by not adding item i in the first stage we obtain a solution with objective function value at least $v^* + \frac{1}{m}$, a contradiction. □

□

As a direct corollary of proposition 4.5 as well as the non-approximability of the $MCKP$ we obtain:

Corollary 4.7. *For all $\epsilon > 0$, there exists no $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for the $RejTSK^D$, unless $\mathcal{P} = \mathcal{NP}$.*

Remark: Before finishing this section on the non-approximability of the $TSKP$, we would like to remark the following:
(i) As long as we require the second-stage rewards to be strictly positive, the $RejTSK^D$ is not a special case of the TSK^D: In general, we cannot reformulate an instance of the $RejTSK^D$ as an instance of the TSK^D with the same solution value, as this is the case for the $AddTSK^D$. However, by combining the results of this subsection with those of the previous subsections (or by proving it with a proof similar to that of Proposition 3.1)

one can show that the optimal solution of an instance of $RejTSK^D$ can be obtained by solving a corresponding instance of TSK^D.
(ii) Nevertheless, the proof of Theorem 4.1 can be rewritten in the following way: Any instance of the $MCKP$ can be equivalently reformulated as an instance of the $RejTSK^D$. In turn, the solution of an instance of the $RejTSK^D$ can be obtained by solving a corresponding instance of the TSK^D.

5. Special-case approximation algorithms

5.1. The $TSKP$ with reward dependent second-stage rewards or penalties

Proposition 5.1. *Let $\alpha \in (0,1)$ and denote $TSKP(\alpha)$ the variant of the $TSKP$ where $\overline{r}_i = \alpha \cdot r_i$ for all $i \in \{1,\dots,n\}$. Then there exists an approximation algorithm for the $TSKP(\alpha)$ with approximation ratio α.*

Proof. Let an instance $\mathcal{I} = (c, r, \overline{r}, d, \chi^1, \dots, \chi^K, p^1, \dots, p^K)$ of the $TSKP(\alpha)$ be given. We show that adding no item in the first-stage yields a solution with overall solution value at least $\alpha \cdot OPT(\mathcal{I})$.
Let $\mathcal{I}_k$ $(k \in \{1,\dots,K\})$ denote the instance of the deterministic knapsack problem having $\overline{r}$ as reward vector, c as capacity and χ^k as weight vector. Let y^k be an optimal solution of $\mathcal{I}_k$. Then adding no items in the first stage gives a solution for $\mathcal{I}$ with overall solution value

$$\sum_{i=1}^{n} 0 \cdot r_i + \sum_{k=1}^{K} p_k \sum_{i=1}^{n} \overline{r}_i y_i^k = \sum_{k=1}^{K} p_k \sum_{i=1}^{n} \alpha \cdot r_i y_i^k = \alpha \sum_{k=1}^{K} p_k \sum_{i=1}^{n} r_i y_i^k$$

$\sum_{k=1}^{K} p_k \sum_{i=1}^{n} r_i y_i^k$ is the optimal solution value of a two-stage knapsack problem with capacity c, weight vectors $(\chi^1,\dots,\chi^K)$, probabilities $(p^1,\dots,p^K)$, reward vector r and second-stage reward vector $\widetilde{r} = r$ (the penalty vector can be arbitrary as long as $d_i > r_i$ for all $i = 1,\dots,n$). $\sum_{k=1}^{K} p_k \sum_{i=1}^{n} r_i y_i^k$ is thus clearly an upper bound on the optimal solution value of any $TSKP$ instance with capacity c, weight vectors $(\chi^1,\dots,\chi^K)$, probabilities $(p^1,\dots,p^K)$ and first-stage reward vector r. This terminates the proof. □

From the proof of Proposition 5.1 we immediately get:

Corollary 5.2. *For an instance of the $TSKP$ define $\alpha := \min_{i=1\dots,n} \frac{\overline{r}_i}{r_i}$. Then adding no items in the first stage always yields a solution whose solution value is at least an α-fraction of the optimal solution value.*

Remark that in case where the first- and second-stage rewards are identical, adding no items in the first stage is even an optimal solution.
The case where we only require the second-stage penalties to depend linearly on the first-stage rewards is more complicated. If we assume the second-stage penalties to be equal to the first-stage rewards (i.e., if removing an item does not cost us anything), then an optimal solution would be to add all items in the first-stage. The optimal solution value would be the same as in the case of a $TSKP$ instance where first- and second-stage rewards are identical, i.e., $\sum_{k=1}^{K} p_k \sum_{i=1}^{n} r_i y_i^k$ (where y^k is the optimal solution vector of the deterministic knapsack problem with same capacity, reward vector r and weight vector as in scenario k of the initial $TSKP$ instance). Due to this observation one might thus think that in case of small enough second-stage penalties this same algorithm would yield a solution with solution value close to the optimum. This is of course true. However, we have the following:

Proposition 5.3. *For any $\beta > 1$ there exists an instance of the $TSKP$ with $d_i = \beta \cdot r_i$ for all $i = 1, \ldots, n$ such that adding all items in the first stage yields a negative solution value. This is still true if one assumes that in each scenario at least one item fits the capacity.*

Proof. Let $\beta > 1$ be given. Choose n and a reward vector r such that

$$\frac{r_{max}}{r_{min}} < (n-1)(\beta - 1)$$

where $r_{max} = \max_{i=1,\ldots,n} r_i$ and $r_{min} = \min_{i=1,\ldots,n} r_i$. Assume w.l.o.g. that $\arg\max_{i=1,\ldots,n} r_i = n$. We have

$$\begin{aligned} \beta - 1 &> \frac{r_{max}}{(n-1)r_{min}} \geq \frac{r_{max}}{\sum_{i=1}^{n-1} r_i} \\ \Rightarrow \beta &> \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} r_i - r_{max}} \Rightarrow \sum_{i=1}^{n} r_i - \beta(\sum_{i=1}^{n} r_i - r_{max}) < 0 \end{aligned}$$

It follows that adding all items in the first-stage would yield a negative solution value in case all but one item have to be removed in each scenario.
□

We thus cannot obtain a result of the form "if $d_i = \beta \cdot r_i$ for all $i = 1, \ldots, n$, then adding all items in the first stage yields a γ-approximation algorithm" (with, of course, $\gamma > 0$).

5.2. The TSKP under the assumption of a polynomial scenario model

Here is a simple approximation algorithm with approximation ratio $\frac{1}{n}$ (denoted by $\mathcal{A}^1_{apx}$): For all $i = 1, \ldots, n$ let R_i denote the maximum expected reward we can obtain for item i within any solution. If $\chi_i^k \leq c$ for all $k = 1, \ldots, K$, this is certainly r_i.
Let $\mathcal{K}_i = \{k \in \{1, \ldots, K\} : \chi_i^k \leq c\}$. 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. We get:

Proposition 5.4. *Under the assumption of a polynomial scenario model, algorithm $\mathcal{A}^1_{apx}$ is a $\frac{1}{n}$-approximation algorithm for the TSKP.*

Note that $\frac{1}{n} > \frac{1}{\sqrt{K}}$ if and only if $n^2 < K$. This means that if $n^2 < K$ (but K polynomial in n) there exists a $K^{-\frac{1}{2}+\epsilon}$-approximation algorithm for some $\epsilon > 0$. This is, however, what one might expect as due to the reduction from the maximum clique problem, the non-approximability result of Theorem 4.1 only applies "directly" to instances with $n^2 \geq K$ (as in a graph the number of edges is smaller than the square of the number of vertices).

5.3. The AddTSKP where the number of scenarios is assumed to be a constant

Let *K-AddTSKP* (*K-MCKP*) denote the variant of the *AddTSKP* (*MCKP*) where the number of scenarios (constraints) is fixed to be K. An algorithm with approximation ratio depending on K is the following (denoted algorithm $\mathcal{A}^2_{apx}$): First solve the following multiply-constrained knapsack problem using the $\frac{1}{m+1}$-approximation algorithm proposed e.g. in [26] for the *K-MCKP*. Denote $\widetilde{x}$ the obtained solution and R' the obtained solution value:

$$\max_{x \in \{0,1\}^n} \quad \sum_{i=1}^{n} r_i x_i \tag{3}$$
$$\text{s.t.} \quad \sum_{i=1}^{n} x_i \chi_i^k \leq c \quad \forall\, k = 1, \ldots, K.$$

Then, for all $k = 1, \ldots, K$, solve the following knapsack problem with the 2-approximation algorithm for deterministic knapsack problems and denote

$(\widetilde{y}^{+})^{k}$ the obtained solution:

$$\begin{aligned} \max_{y^+ \in \{0,1\}^n} \quad & \sum_{i=1}^{n} \overline{r}_i y_i^+ \\ \text{s.t.} \quad & \sum_{i=1}^{n} y_i^+ \chi_i^k \le c. \end{aligned} \tag{4}$$

For all $k = 1, \ldots, K$ define $R^k = \sum_{i=1}^{n} \overline{r}_i (\widetilde{y}^+)_i^k$ and $R'' = \sum p^k R^k$. If $R' \ge R''$, output $\widetilde{x}$. Otherwise output 0.

Proposition 5.5. *Algorithm* $\mathcal{A}^2_{apx}$ *is a* $\frac{1}{2(K+1)}$*-approximation algorithm for the* K*-AddTSKP that runs in linear time.*

Proof. The linear time requirement follows from the running time of the $\frac{1}{2}$-approximation algorithm for knapsack problems and the $\frac{1}{m+1}$-approximation algorithm for K-$MCKP$ that both run in $\mathcal{O}(n)$ time.
First note that the overall solution value of the solution produced by Algorithm $\mathcal{A}^2_{apx}$ is at least $\max(R', R'')$. Then, to prove that our algorithm has an approximation ratio of $\frac{1}{2(K+1)}$, let an instance of the K-$AddTSKP$ be given and fix an optimal (first-stage) solution x^*. Define

$$opt' := \sum_{i=0}^{n} r_i x_i^*$$

i.e., opt' is the reward due to the the first-stage items. As x^* is feasible for problem (3), it follows $R' \ge opt'$.
Let opt'' be the expected second-stage reward for the fixed solution, i.e.,

$$opt'' := \sum_{k=0}^{K} p^k \sum_{i=0}^{n} \overline{r}_i (\overline{y}_i^+)^k$$

Let furthermore opt^k denote the total reward of the items added in scenario k and opt the optimal overall solution value. It follows

$$\begin{aligned} opt &= opt' + opt'' \le 2 \cdot \max(opt', opt'') \\ &= 2 \cdot \max(opt', \sum_{k=1}^{K} p^k opt^k) \le 2 \cdot \max((K+1)R', 2\sum_{k=1}^{K} p^k R^k) \\ &\le 2(K+1)\max(R', R'') \end{aligned}$$

which finishes the proof. □

Theorem 5.6. *For all $\epsilon \in (0, \frac{1}{2})$, there exists an approximation algorithm for the K-$TSKP$ with approximation-ratio $\frac{1}{2} - \epsilon$ and running time polynomial in n.*

Proof. Let ϵ and an instance of the the K-$TSKP$ be given. Define $\epsilon' := 2\epsilon$. The algorithm works as follows: First solve the K-$MCKP$ (3) by a $PTAS$ (see e.g. [1]) with accuracy $1 - \epsilon'$. Denote the obtained solution value by R'. Then, solve the K knapsack problems in (4) using an $FPTAS$ with accuracy $1 - \epsilon'$. Denote R^k, $k = 1, \ldots, K$, the obtained solution values and define $R'' := \sum_{k=1}^{K} p^k R^k$. If $R' > R''$, output the approximate solution of the K-$MCKP$ (3). Otherwise output the zero vector.
Similar as in the proof of Proposition 5.5 we get

$$opt \leq \frac{2}{1-\epsilon'} \max(R', R'') \Longrightarrow \max(R', R'') \geq (\frac{1}{2} - \epsilon)opt$$

The running time follows from the running time of the $PTAS$ for the K-$MCKP$ proposed in [26] (that is $\mathcal{O}(n^{\lceil \frac{K}{\epsilon'} \rceil - K})$) and the running time of the $FPTAS$ for the knapsack problem. □

Note that the running time of both algorithm $\mathcal{A}^2_{apx}$ as well as the algorithm proposed in the proof of Theorem 5.6 are only polynomial in n if K is a constant. The reason is that both the $\frac{1}{m+1}$-approximation algorithm as well as the $PTAS$ for the K-$MCKP$ proposed in [26] have running time that depend exponentially on K.
Furthermore, note that the running time of the algorithm proposed in the proof of Theorem 5.6 depends exponentially on $\frac{1}{\epsilon}$ due to the strong $\mathcal{NP}$-hardness of the K-$MCKP$.

The algorithms in this section were inspired by the $\frac{1}{2}$-approximation algorithm for the Two-Stage Matching problem proposed in [16]. However, there is an important difference between this stochastic matching problem variant and the $TSKP$ studied in this paper: While in the former the uncertainties occur in the objective function, they occur in the constraint in the latter. This prevents us from obtaining a better approximation ratio than $\frac{1}{2(K+1)}$ (or $\frac{1}{2} - \epsilon$) by following the ideas of the algorithm proposed in [29] for the Two-Stage Matching problem: The authors show how by making a slight modification in the algorithm proposed in [16] the approximation ratio can be improved from $\frac{1}{2}$ to $\frac{K}{2K-1}$. However, this idea does not extend to the $TSKP$ as the solution that is constructed during the modified algorithm might be infeasible in the presence of uncertainty in the constraints.

6. The special case of independently, discretely distributed weights

In this section we assume that the item weights are independently distributed and that item i has a number K_i of possible outcomes. As already pointed out, if $K_i \geq 2$ (for all items i) the a total number K of scenarios is consequently exponential in the number of items.
However, in this special case algorithm $\mathcal{A}^1_{apx}$ is a $\frac{1}{n}$-approximation algorithm for the $TSKP$, as long as the K_i are polynomial in the number of items n.

Proposition 6.1. *Under the assumption that the number K_i of outcomes of χ_i is polynomial in n (for all $i \in \{1, \dots, n\}$), algorithm $\mathcal{A}^1_{apx}$ is a $\frac{1}{n}$-approximation algorithm for the $TSKP$.*

Proof. Let us denote as usual $\chi^1, \dots, \chi^K$ the K possible outcomes of the weight vector χ (where $K = \prod_{i=1}^n K_i$) and $\hat{\chi}_i^1, \dots, \hat{\chi}_i^{K_i}$ the K_i possible outcomes of the random variable χ_i. Instead of computing $\mathcal{K}_i$ as $\mathcal{K}_i = \{k \in \{1, \dots, K\} : \chi_i^k \leq c\}$, it can be computed as $\mathcal{K}_i = \{k \in \{1, \dots, K_i\} : \hat{\chi}_i^k \leq c\}$. Computing $\mathcal{K}_i$ or a sum over the elements of $\mathcal{K}_i$ for all i can thus been done in polynomial time. This makes $\mathcal{A}^1_{apx}$ a $\frac{1}{n}$-approximation algorithm in the case of independently distributed weights with polynomial number of outcomes. □

7. Conclusion

In this paper we studied the Two-Stage Knapsack problem with random weights ($TSKP$) under the aspect of approximability. The studied model allows both removal and adding of items in the second stage, and items can even be exchanged in favor of more reward effective items. We assumed the item weights to be discretely distributed, which allows the reformulation of the problem as a linear combinatorial optimization problem. However, solving the problem exactly is only tractable for a rather small number of second-stage scenarios.
We showed that the problem cannot be approximated within a factor $K^{-\frac{1}{2}+\epsilon}$, unless $\mathcal{P} = \mathcal{NP}$ (where K is the number of second-stage scenarios and $\epsilon > 0$). However, we achieved simple approximation algorithms for the special cases of linearly dependent first- and second-stage rewards, the polynomial scenario model and a fixed number of scenarios.
The key to the results presented in this paper is a reduction of the well studied multiply-constrained knapsack problem to the $TSKP$. While it is easy to see that the deterministic reformulation of the $TSKP$ is a multiply-constrained knapsack problem where some item rewards are negative, the

fact that any instance of the multiply-constrained knapsack problem can be solved via an instance of the $TSKP$ is not that obvious.
Future work might first of all consist in studying further variants and special cases of the $TSKP$ that might allow for better approximation ratios such as the variant where item rewards are a fix multiple of the item weights (and thus also random) or the variant with Bernoulli-distributed "on-off" items (see [24]).
The deterministic reformulation of the $TSKP$ has a very special structure known in the literature as "L-shaped". This structure might be used in order to get a tighter non-approximability result or better approximation ratios for the special cases. However, we conjecture that it is hard to approximate the variant of the $TSKP$ where the number of second-stage scenarios is assumed to be a constant at a better approximation ratio than $\frac{1}{2}$.
As mentioned in the introduction, the authors of [22] present a very interesting idea that despite being applyied to a problem with a much differnt structure might also give an idea for an approximation algorithm for the $TSKP$. Similar to the basic idea of the Fully Polynomial Time Approximation algorithm for the (deterministic) knapsack problem (i.e. to work with "truncated" weights), the authors of [22] propose to reduce the set of possible weight distributions to a well defined set of "canonical" weight distributions. Moreover, instead of dividing the items into large and small size items as it is the basis of some famous approximative knapsack algorithms, the authors classify in their algorithm the items by their *probability* to realize to a small or large size item. The application of one or both paradigms to the $TSKP$ or one of its special variants seems - to us - very much worth looking into.

Acknowledgement

Many thanks to Prof. Peter Jonsson for giving me the hint that a better approximation ratio for the maximum clique problem has been found in 2006 which allowed me to tighten the non-approximability result.
Special thanks to an anonymous referee who pointed out that the case of independently, discretely distributed weights might be an interesting special case to be studied.

References

[1] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack Problems, Springer-Verlag (Berlin, Heidelberg), 2004.

[2] R. L. Carraway, R. L. Schmidt, L. R. Weatherford, An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns, Naval research logistics 40(2) (1993) 161–173.

[3] M. I. Henig, Risk criteria in a stochastic knapsack problem, Operations Research 38(5) (1990) 820–825.

[4] D. P. Morton, R. K. Wood, Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search, Kluwer Academic Publishers, 1997, Ch. On a stochastic knapsack problem and generalizations, pp. 149–168.

[5] Y. Merzifonluoglu, J. Geunes, H. Romeijn, The static stochastic knapsack problem with normally distributed item sizes, Mathematical Programming 134 (2012) 459–489.

[6] V. Goyal, R. Ravi, A ptas for the chance-constrained knapsack problem with random item sizes, Operations Research Letters 38 (3) (2010) 161–164.

[7] A. Gaivoronski, A. Lisser, R. Lopez, H. Xu, Knapsack problem with probability constraints, Journal of Global Optimization 49 (2011) 397–413.

[8] A. Bhalgat, A. Goel, S. Khanna, Improved approximation results for stochastic knapsack problems, in: SODA, 2011, pp. 1647–1665.

[9] G. Dantzig, Linear programming under uncertainty, Management Science 1 (1955) 197–206.

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

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

[12] D. B. Shmoys, C. Swamy, An approximation scheme for stochastic linear programming and its application to stochastic integer programs, Journal of the ACM 53(6) (2006) 978–1012.

[13] D. B. Shmoys, M. Sozio, Approximation algorithms for 2-stage stochastic scheduling problems, in: Proc. of the 12th international conference on Integer Programming and Combinatorial Optimization, Vol. 4513 of Lecture Notes in Computer Science, Springer, 2007, pp. 145–157.

[14] S. Dye, L. Stougie, A. Tomasgard, The stochastic single resource service-provision problem, Naval Research Logistics 50(8) (2003) 869–887.

[15] A. D. Flaxman, A. Frieze, M. Krivelevich, On the random 2-stage minimum spanning tree, in: Proc. of the 16th Annual ACM-SIAM Symposium on Discrete algorithms, 2005, pp. 919–926.

[16] N. Kong, A. J. Schaefer, A factor $\frac{1}{2}$ approximation algorithm for two-stage stochastic matching problems, European Journal of Operational Research 172(3) (2006) 740–746.

[17] I. Katriel, C. Kenyon-Mathieu, E. Upfal, Commitment under uncertainty: Two-stage stochastic matching problems, in: ICALP, Vol. 4596 of Lecture Notes in Computer Science, Springer, 2007, pp. 171–182.

[18] C. Swamy, D. B. Shmoys, Algorithms column: Approximation algorithms for 2-stage stochastic optimization problems, ACM SIGACT News 37(1) (2006) 33–46.

[19] L. Stougie, M. H. van der Vlerk, Approximation in stochastic integer programming, Tech. Rep. 03A14, University of Groningen, Research Institute SOM (2003).

[20] N. Immorlica, D. Karger, M. Minkoff, V. S. Mirrokni, On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems, in: Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, Session 8B, 2004, pp. 691–700.

[21] B. C. Dean, M. X. Goemans, J. Vondrák, Approximating the stochastic knapsack problem: The benefit of adaptivity, in: Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2004, pp. 208–217.

[22] A. Gupta, R. Krishnaswamy, M. Molinaro, R. Ravi, Approximation algorithms for correlated knapsacks and non-martingale bandits, in: IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), 2011, pp. 827–836.

[23] J. Kleinberg, Y. Rabani, E. Tardos, Allocating bandwidth for bursty connections, in: Proc. of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 664–673.

[24] A. Goel, P. Indyk, Stochastic load balancing and related problems, in: 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 579–586.

[25] T. Benoist, E. Bourreau, B. Rottembourg, Towards stochastic constraint programming: A study of online multi-choice knapsack with deadlines, in: Proc. of the CP '01, Springer London, 2001, pp. 61–76.

[26] A. Caprara, H. Kellerer, U. Pferschy, D. Pisinger, Approximation algorithms for knapsack problems with cardinality constraints, European Journal of Operational Research 123 (1998) 2000.

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

[28] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, in: Proc. of the thirty-eighth annual ACM symposium on Theory of computing, STOC '06, 2006, pp. 681–690.

[29] B. Escoffier, L. Gourvs, J. Monnot, O. Spanjaard, Two-stage stochastic matching and spanning tree problems: Polynomial instances and approximation, European Journal of Operational Research 205 (1) (2010) 19–30.