Multistage Knapsack

Evripidis Bampis
Sorbonne Université, CNRS, LIP6, France
evripidis.bampis@lip6.fr

Bruno Escoffier
Sorbonne Université, CNRS, LIP6, France
bruno.escoffier@lip6.fr

Alexandre Teiller
Sorbonne Université, CNRS, LIP6, France
alexandre.teiller@lip6.fr

Abstract

Many systems have to be maintained while the underlying constraints, costs and/or profits change over time. Although the state of a system may evolve during time, a non-negligible transition cost is incured for transitioning from one state to another. In order to model such situations, Gupta et al. (ICALP 2014) and Eisenstat et al. (ICALP 2014) introduced a *multistage* model where the input is a sequence of instances (one for each time step), and the goal is to find a sequence of solutions (one for each time step) that simultaneously (i) have good quality on the time steps and (ii) as stable as possible. We focus on the *multistage* version of the KNAPSACK problem where we are given a time horizon $t = 1, 2, \ldots, T$, and a sequence of knapsack instances $I_1, I_2, \ldots, I_T$, one for each time step, defined on a set of n objects. In every time step t we have to choose a feasible knapsack S_t of I_t, which gives a *knapsack profit*. To measure the stability/similarity of two consecutive solutions S_t and S_{t+1}, we identify the objects for which the decision, to be picked or not, remains the same in S_t and S_{t+1}, giving a *transition profit*. We are asked to produce a sequence of solutions $S_1, S_2, \ldots, S_T$ so that the total knapsack profit plus the overall transition profit is maximized.

We propose a PTAS for the MULTISTAGE KNAPSACK problem. This is the first approximation scheme for a combinatorial optimization problem in the considered multistage setting, and its existence contrasts with the inapproximability results for other combinatorial optimization problems that are even polynomial-time solvable in the static case (e.g.MULTISTAGE SPANNING TREE, or MULTISTAGE BIPARTITE PERFECT MATCHING). Then, we prove that there is no FPTAS for the problem even in the case where $T = 2$, unless $P = NP$. Furthermore, we give a pseudopolynomial time algorithm for the case where the number of steps is bounded by a fixed constant and we show that otherwise the problem remains NP-hard even in the case where all the weights, profits and capacities are 0 or 1.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Knapsack, Approximation Algorithms, Multistage Optimization

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.22

Acknowledgements This research benefited from the support of FMJH program PGMO and from the support of EDF-Thalès-Orange.

1 Introduction

In a classical combinatorial optimization problem, given an instance of the problem we seek a feasible solution optimizing the objective function. However, in many systems the input may change over the time and the solution has to be adapted to the input changes. It is then necessary to determine a tradeoff between the optimality of the solutions in each time step and the stability/similarity of consecutive solutions. This is important since in many applications there is a significant transition cost for changing (parts of) a solution. Recently,

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 22; pp. 22:1–22:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Gupta et al. [15] and Eisenstat et al. [11] introduced a *multistage* model in order to deal with such situations. They consider that the input is a sequence of instances (one for each time step), and the goal is to find a sequence of solutions (one for each time step) reaching such a tradeoff.

Our work follows the direction proposed by Gupta et al. [15] who suggested the study of more combinatorial optimization problems in their multistage framework. In this paper, we focus on the multistage version of the KNAPSACK problem. Consider a company owning a set $N = \{u_1, \ldots, u_n\}$ of production units. Each unit can be used or not; if u_i is used, it spends an amount w_i of a given resource (energy, raw material,...), and generates a profit p_i. Given a bound W on the global amount of available resource, the static KNAPSACK problem aims at determining a feasible solution that specifies the chosen units in order to maximize the total profit under the constraint that the total amount of the resource does not exceed the bound of W. In a multistage setting, considering a time horizon $t = 1, 2, \ldots, T$ of, let us say, T days, the company needs to decide a production plan for each day of the time horizon, given that data (such as prices, level of resources,...) usually change over time. This is a typical situation, for instance, in energy production planning (like electricity production, where units can be nuclear reactors, wind or water turbines,...), or in data centers (where units are machines and the resource corresponds to the available energy). Moreover, in these examples, there is an extra cost to turn ON or OFF a unit like in the case of turning ON/OFF a reactor in electricity production [25], or a machine in a data center [1]. Obviously, whenever a reactor is in the ON or OFF state, it is beneficial to maintain it at the same state for several consecutive time steps, in order to avoid the overhead costs of state changes. Therefore, the design of a production plan over a given time horizon has to take into account both the profits generated each day from the operation of the chosen units, as well as the potential transition profits from maintaining a unit at the same state for two consecutive days.

We formalize the problem as follows. We are given a time horizon $t = 1, 2, \ldots, T$, and a sequence of knapsack instances $I_1, I_2, \ldots, I_T$, one for each time step, defined on a set of n objects. In every time step t we have to choose a feasible knapsack S_t of I_t, which gives a *knapsack profit*. Taking into account transition costs, we measure the stability/similarity of two consecutive solutions S_t and S_{t+1} by identifying the objects for which the decision, to be picked or not, remains the same in S_t and S_{t+1}, giving a *transition profit*. We are asked to produce a sequence of solutions $S_1, S_2, \ldots, S_T$ so that the total knapsack profit plus the overall transition profit is maximized.

Our main contribution is a polynomial time approximation scheme (PTAS) for the multistage version of the KNAPSACK problem. Up to the best of our knowledge, this is the first approximation scheme for a multistage combinatorial optimization problem and its existence contrasts with the inapproximability results for other combinatorial optimization problems that are even polynomial-time solvable in the static case (e.g. MULTISTAGE SPANNING TREE [15], or MULTISTAGE BIPARTITE PERFECT MATCHING [4]).

1.1 Problem definition

Formally, the MULTISTAGE KNAPSACK problem can be defined as follows.

▶ **Definition 1.** *In the* MULTISTAGE KNAPSACK *problem (MK) we are given:*

- *a time horizon $T \in \mathbb{N}^*$, a set $N = \{1, 2, \ldots, n\}$ of objects;*
- *For any $t \in \{1, \ldots, T\}$, any $i \in N$:*
  - *p_{ti} the profit of taking object i at time t*
  - *w_{ti} the weight of object i at time t*

- *For any* $t \in \{1, \dots, T-1\}$*, any* $i \in N$*:* $B_{ti} \in \mathbb{R}^+$ *the bonus of the object* i *if we keep the same decision for* i *at time* t *and* $t+1$*.*
- *For any* $t \in \{1, \dots, T\}$*: the capacity* C_t *of the knapsack at time* t*.*

We are asked to select a subset $S_t \subseteq N$ *of objects at each time* t *so as to respect the capacity constraint:* $\sum_{i \in S_t} w_{ti} \leq C_t$*. To a solution* $S = (S_1, \dots, S_T)$ *are associated:*

- *A knapsack profit* $\sum_{t=1}^{T} \sum_{i \in S_t} p_{ti}$ *corresponding to the sum of the profits of the* T *knapsacks;*
- *A transition profit* $\sum_{t=1}^{T-1} \sum_{i \in \Delta_t} B_{ti}$ *where* Δ_t *is the set of objects either taken or not taken at both time steps* t *and* $t+1$ *in* S *(formally* $\Delta_t = (S_t \cap S_{t+1}) \cup (\overline{S_t} \cap \overline{S_{t+1}})$*).*

The value of the solution S *is the sum of the knapsack profit and the transition profit, to be maximized.*

1.2 Related works

Multistage combinatorial optimization. A lot of optimization problems have been considered in online or semi-online settings, where the input changes over time and the algorithm has to modify the solution by making as few changes as possible. Tradeoffs between modification costs and quality of solutions have been also studied in the reoptimization setting. We refer the reader to [3, 6, 10, 14, 22, 23, 26] and the references therein.

Multistage optimization has been studied for fractional problems by Buchbinder et al. [8] and Buchbinder, Chen and Naor [7]. The multistage model considered in this article is the one studied in Eisenstat et al. [11] and Gupta et al. [15]. Eisenstat et al. [11] studied the multistage version of facility location problems. They proposed a logarithmic approximation algorithm. An et al. [2] obtained constant factor approximation for some related problems. Gupta et al. [15] studied the Multistage Maintenance Matroid problem for both the offline and the online settings. They presented a logarithmic approximation algorithm for this problem, which includes as a special case a natural multistage version of Spanning Tree. The same paper also introduced the study of the Multistage Minimum Perfect Matching problem. They showed that the problem becomes hard to approximate even for a constant number of stages. Later, Bampis et al. [4] showed that the problem is hard to approximate even for bipartite graphs and for the case of two time steps. In the case where the edge costs are metric within every time step they first proved that the problem remains APX-hard even for two time steps. They also showed that the maximization version of the problem admits a constant factor approximation algorithm but is APX-hard. In another work [5], the Multistage Max-Min Fair Allocation problem has been studied in the offline and the online settings. This corresponds to a multistage variant of the Santa Klaus problem. For the off-line setting, the authors showed that the multistage version of the problem is much harder than the static one. They provided constant factor approximation algorithms for the off-line setting.

Knapsack variants. Our work builds upon the Knapsack literature [18]. It is well known that there is a simple 2-approximation algorithm as well as a fully polynomial time (FPTAS) for the static case [16, 20, 21, 17]. There are two variants that are of special interest for our work:

(i) The first variant is a generalization of the Knapsack problem known as the k-Dimensional Knapsack ($k - DKP$) problem:

▶ **Definition 2.** *In the* k-Dimensional Knapsack *problem* ($k - DKP$)*, we have a set* $N = \{1, 2, \ldots, n\}$ *of objects. Each object* i *has a profit* p_i *and* k *weights* w_{ji}, $j = 1, \ldots, k$*. We are also given* k *capacities* C_j*. The goal is to select a subset* $Y \subseteq N$ *of objects such that:*

- *The capacity constraints are respected: for any* j, $\sum_{i \in Y} w_{ji} \leq C_j$*;*
- *The profit* $\sum_{i \in Y} p_i$ *is maximized.*

It is well known that for the usual Knapsack problem, in the continuous relaxation (variables in $[0, 1]$), at most one variable is fractional. Caprara et al. [9] showed that this can be generalized for $k - DKP$.

Let us consider the following ILP formulation ($ILP - DKP$) of the problem:

$$\begin{cases} \max \sum_{i \in N} p_i y_i \\ s.t. \left| \begin{array}{lll} \sum_{i \in N} w_{ji} y_i & \leq \; C_j & \forall j \in \{1, ..., k\} \\ y_i \in \{0, 1\} & & \forall i \in N \end{array} \right. \end{cases}$$

▶ **Theorem 3.** *[9] In the continuous relaxation* ($LP - DKP$) *of* ($ILP - DKP$) *where variables are in* $[0, 1]$*, in any basic solution at most* k *variables are fractional.*

A basic solution is an extreme point (vertex) of the polytope of solutions. Note that with an easy affine transformation on variables, the same result holds when variable y_i is subject to $a_i \leq y_i \leq b_i$ instead of $0 \leq y_i \leq 1$: *in any basic solution at most* k *variables* y_i *are such that* $a_i < y_i < b_i$.

Caprara et al. [9] use the result of Theorem 3 to show that for any fixed constant k, $k - DKP$ admits a polynomial time approximation scheme (PTAS). Other PTASes have been presented in [24, 12]. Korte and Schrader [19] showed that there is no FPTAS for $k - DKP$ unless $P = NP$.

(ii) The second related variant is a simplified version of $k - DKP$ called Cardinality-2-KP, where the dimension is 2, all the profits are 1 and, given a K, we are asked if there is a solution of value at least K (decision problem). In other words, given two knapsack constraints, can we take K objects and verify the two constraints? The following result is shown in [18].

▶ **Theorem 4.** *[18]* Cardinality-2-KP *is* NP*-complete.*

1.3 Our contribution

As stated before, our main contribution is to propose a PTAS for the multistage Knapsack problem. Furthermore, we prove that there is no FPTAS for the problem even in the case where $T = 2$, unless $P = NP$. We also give a pseudopolynomial time algorithm for the case where the number of steps is bounded by a fixed constant and we show that otherwise the problem remains NP-hard even in the case where all the weights, profits and capacities are 0 or 1. The following table summarizes our main result pointing out the impact of the number T of time steps on the difficulty of the problem ("no FPTAS" means "no FPTAS unless P=NP").

$T = 1$	T fixed	any T
pseudopolynomial	pseudopolynomial	strongly NP-hard
FPTAS	PTAS	PTAS
-	no FPTAS	no FPTAS

We point out that the negative results (strongly NP-hardness and no FPTAS) hold even in the case of *uniform bonus*, i.e., when $B_{ti} = B$ for all $i \in N$ and all $t = 1, \ldots, T - 1$.

2 ILP formulation

The Multistage Knapsack problem can be written as an ILP as follows. We define Tn binary variables x_{ti} equal to 1 if i is taken at time t ($i \in S_t$) and 0 otherwise. We also define $(T-1)n$ binary variables z_{ti} corresponding to the transition profit of object i between time t and $t+1$. The profit is 1 if i is taken at both time steps, or taken at none, and 0 otherwise. Hence, $z_{ti} = 1 - |x_{(t+1)i} - x_{ti}|$. Considering that we solve a maximization problem, this can be linearized by the two inequalities: $z_{ti} \leq -x_{(t+1)i} + x_{ti} + 1$ and $z_{ti} \leq x_{(t+1)i} - x_{ti} + 1$. We end up with the following ILP (called $ILP - MK$):

$$\begin{cases} \max \sum_{t=1}^{T} \sum_{i \in N} p_{ti} x_{ti} + \sum_{t=1}^{T-1} \sum_{i \in N} z_{ti} B_{ti} & \\ s.t. \left| \begin{array}{lll} \sum_{i \in N} w_{ti} x_{ti} & \leq \; C_t & \forall t \in \{1, ..., T\} \\ z_{ti} & \leq \; -x_{(t+1)i} + x_{ti} + 1 & \forall t \in \{1, ..., T-1\}, \forall i \in N \\ z_{ti} & \leq \; x_{(t+1)i} - x_{ti} + 1 & \forall t \in \{1, ..., T-1\}, \forall i \in N \\ x_{ti} \in \{0,1\} & & \forall t \in \{1, ..., T\}, \forall i \in N \\ z_{ti} \in \{0,1\} & & \forall t \in \{1, ..., T-1\}, \forall i \in N \end{array} \right. \end{cases}$$

In devising the PTAS we will extensively use the linear relaxation $(LP - MK)$ of $(ILP - MK)$ where variables x_{ti} and z_{ti} are in $[0, 1]$.

3 A polynomial time approximation scheme

In this section we show that Multistage Knapsack admits a PTAS. The central part of the proof is to derive a PTAS when the number of steps is a fixed constant (Sections 3.1 and 3.2). The generalization to an arbitrary number of steps is done in Section 3.3.

Building upon [9], our PTAS for a fixed number of time steps heavily relies on a property of the relaxed LP-formulation of Multistage Knapsack: we show that there are at most T^3 fractional variables in an optimal (basic) solution of the (relaxed) Multistage Knapsack problem. Based on this bound, the PTAS is built from a combination of (1) bruteforce search (to find the most profitable objects), (2) a preprocessing step and (3) a rounding of the fractional solution of the (relaxed) LP-formulation. The preprocessing step associated to the bound on the number of fractional variables allow to bound the global loss of the solution built by the algorithm.

We show how to bound the number of fractional variables in Section 3.1. We first illustrate the reasoning on the case of two time-steps, and then present the general result. In Section 3.2 we present the PTAS for a constant number of steps. For ease of notation, we will sometimes write a feasible solution as $S = (S_1, \ldots, S_T)$ (subsets of objects taken at each time step), or as $S = (x, z)$ (values of variables in $(ILP - MK)$ or $(LP - MK)$).

3.1 Bounding the number of fractional objects in $(LP - MK)$

3.1.1 Warm-up: the case of two time-steps

We consider in this section the case of two time-steps ($T = 2$), and focus on the linear relaxation $(LP - MK)$ of $(ILP - MK)$ with the variables x_{ti} and z_i in $[0, 1]$ (we write z_i instead of z_{1i} for readability). We say that an object is *fractional* in a solution S if x_{1i}, x_{2i} or z_i is fractional.

Let us consider a (feasible) solution $\hat{S} = (\hat{x}, \hat{z})$ of $(LP - MK)$, where $\hat{z}_i = 1 - |\hat{x}_{2i} - \hat{x}_{1i}|$ (variables $\hat{z}_i$ are set to their optimal value w.r.t. $\hat{x}$). We show the following.

▶ **Proposition 5.** *If $\hat{S}$ is a basic solution of $(LP-MK)$, at most 4 objects are fractional.*

Proof. First note that since we assume $\hat{z}_i = 1 - |\hat{x}_{1i} - \hat{x}_{2i}|$, if $\hat{x}_{1i}$ and $\hat{x}_{2i}$ are both integers then $\hat{z}_i$ is an integer. So if an object i is fractional either $\hat{x}_{1i}$ or $\hat{x}_{2i}$ is fractional.

Let us denote:

- L the set of objects i such that $\hat{x}_{1i} = \hat{x}_{2i}$.
- $P = N \setminus L$ the set of objects i such that $\hat{x}_{1i} \neq \hat{x}_{2i}$.

We first show Fact 1.

Fact 1. In P there is at most one object i with $\hat{x}_{1i}$ fractional.

Suppose that there are two such objects i and j. Note that since $0 < |\hat{x}_{1i} - \hat{x}_{2i}| < 1$, $\hat{z}_i$ is fractional, and so is $\hat{z}_j$. Then, for a sufficiently small $\epsilon > 0$, consider the solution S_1 obtained from $\hat{S}$ by transfering at time 1 an amount ϵ of weight from i to j (and adjusting consequently z_i and z_j). Namely, in S_1:

- $x^1_{1i} = \hat{x}_{1i} - \frac{\epsilon}{w_{1i}}$, $z^1_i = \hat{z}_i - d_i \frac{\epsilon}{w_{1i}}$, where $d_i = 1$ if $\hat{x}_{2i} > \hat{x}_{1i}$ and $d_i = -1$ if $\hat{x}_{2i} < \hat{x}_{1i}$ (since i is in P $\hat{x}_{2i} \neq \hat{x}_{1i}$).
- $x^1_{1j} = \hat{x}_{1j} + \frac{\epsilon}{w_{1j}}$, $z^1_j = \hat{z}_i + d_j \frac{\epsilon}{w_{1j}}$, where $d_j = 1$ if $\hat{x}_{2j} > \hat{x}_{1j}$ and $d_j = -1$ otherwise.

Note that (for ϵ sufficiently small) S_1 is feasible. Indeed (1) $\hat{x}_{1i}, \hat{x}_{1j}, \hat{z}_i$ and $\hat{z}_j$ are fractional (2) the weight of the knapsack at time 1 is the same in S_1 and in $\hat{S}$ (3) if $\hat{x}_{1i}$ increases by a small δ, if $\hat{x}_{2i} > \hat{x}_{1i}$ then $|\hat{x}_{2i} - \hat{x}_{i1}|$ decreases by δ so $\hat{z}_i$ can increase by δ (so $d_i = 1$), and if $\hat{x}_{2i} < \hat{x}_{i1}$ then $\hat{z}_i$ has to decrease by δ (so $d_i = -1$), and similarly for $\hat{x}_{1j}$.

Similarly, let us define S_2 obtained from $\hat{S}$ with the reverse transfer (from j to i). In S_2:

- $x^2_{1i} = \hat{x}_{1i} + \frac{\epsilon}{w_{1i}}$, $z^2_i = \hat{z}_i + d_i \frac{\epsilon}{w_{1i}}$
- $x^2_{1j} = \hat{x}_{1j} - \frac{\epsilon}{w_{1j}}$, $z^2_j = \hat{z}_i - d_j \frac{\epsilon}{w_{1j}}$

As previously, S_2 is feasible. Then $\hat{S}$ is clearly a convex combination of S_1 and S_2 (with coefficient 1/2), so not a basic solution, and Fact 1 is proven.

In other words (and this interpretation will be important in the general case), for this case we can focus on variables at time one, and interpret *locally* the problem as a (classical, unidimensional) fractional knapsack problem. By locally, we mean that we consider as fixed the variables at time 2: for variables at time 1, if $\hat{x}_{1i} < \hat{x}_{2i}$ then x_{1i} must be in $[0, \hat{x}_{2i}]$ (in S^1, x^1_{1i} cannot be larger than $\hat{x}_{2i}$, otherwise the previous value of z^1_i would be erroneous); similarly if $\hat{x}_{1i} > \hat{x}_{2i}$ then x_{1i} must be in $[\hat{x}_{2i}, 1]$. The profit associated to object i is $p_{1i} + d_i B_{1i}$ (if x_{i1} increases/decreases by ϵ, then the knapsack profit increases/decreases by $p_{1i}\epsilon$, and the transition profit increases/decreases by $\epsilon d_i B_{1i}$, as explained above). Then we have at most one fractional variable, as in any fractional knapsack problem.

In P there is at most one object i with $\hat{x}_{1i}$ fractional. Similarly there is at most one object k with $\hat{x}_{2k}$ fractional. In P, for all but at most two objects, both $\hat{x}_{1i}$ and $\hat{x}_{2i}$, and thus $\hat{z}_i$, are integers.

Note that this argument would not hold for variables in L. Indeed if $\hat{x}_{1i} = \hat{x}_{2i}$, then $\hat{z}_i = 1$, and the transition profit decreases in *both* cases: when $\hat{x}_{1i}$ increases by $\delta > 0$ and when it decreases by δ. So, we cannot express $\hat{S}$ as a convex combination of S_1 and S_2 as previously.

However, let us consider the following linear program $2-DKP$ obtained by fixing variables in P to their values in $\hat{S}$, computing the remaining capacities $C'_t = C_t - \sum_{j\in P} w_{tj}\hat{x}_{tj}$, and "imposing" $x_{1i} = x_{2i}$:

$$\begin{cases} \max \sum_{i\in L}(p_{1i}+p_{2i})y_i + \sum_{i\in L} B_{1i} & & \\ \qquad \sum_{i\in L} w_{1i}y_i & \leq & C'_1 \\ \qquad \sum_{i\in L} w_{2i}y_i & \leq & C'_2 \\ \qquad y_i \in [0,1] & & \forall i \in L \end{cases}$$

Clearly, the restriction of $\hat{S}$ to variables in L is a solution of $2-DKP$. Formally, let $\hat{S}_L = (\hat{y}_j, j \in L)$ defined as $\hat{y}_j = \hat{x}_{1j}$. $\hat{S}_L$ is feasible for $2-DKP$. Let us show that it is basic: suppose a contrario that $\hat{S}_L = \frac{S_L^1+S_L^2}{2}$, with $S_L^1 = (y_i^1, i \in L) \neq S_L^2$ two feasible solutions of $2-DKP$. Then consider the solution $S^1 = (x^1, y^1)$ of $(LP-MK)$ defined as:

- If $i \in L$ then $x_{1i}^1 = x_{2i}^1 = y_i^1$, and $z_{1i}^1 = 1 = \hat{z}_{1i}$.
- Otherwise (for i in P) S^1 is the same as $\hat{S}$.

S^1 is clearly a feasible solution of Multistage Knapsack. If we do the same for S_L^2, we get a (different) feasible solution S^2, and $\hat{S} = \frac{S^1+S^2}{2}$, so $\hat{S}$ is not basic, a contradiction.

By the result of [9], $\hat{S}_L$ has at most 2 fractional variables. Then, in L, for all but at most 2 variables both $\hat{x}_{1i}$, $\hat{x}_{2i}$ and $\hat{z}_i$ are integers. ◀

3.1.2 General case

The case of 2 time steps suggests to bound the number of fractional objects by considering 3 cases:

- Objects with $\hat{x}_{1i}$ fractional and $\hat{x}_{1i} \neq \hat{x}_{2i}$. As explained in the proof of Proposition 5, this can be seen locally (as long as x_{1i} does not reach $\hat{x}_{2i}$) as a knapsack problem from which we can conclude that there is at most 1 such fractional object.
- Similarly, objects with $\hat{x}_{2i}$ fractional and $\hat{x}_{1i} \neq \hat{x}_{2i}$.
- Objects with $\hat{x}_{1i} = \hat{x}_{2i}$ fractional. As explained in the proof of Proposition 5, this can be seen as a $2-DKP$ from which we can conclude that there are at most 2 such fractional objects.

For larger T, we may have different situations. Suppose for instance that we have 5 time steps, and a solution (x, z) with an object i such that: $x_{1i} < x_{2i} = x_{3i} = x_{4i} < x_{5i}$. So we have x_{ti} fractional and constant for $t = 2, 3, 4$, and different from x_{1i} and x_{5i}. The idea is to say that we cannot have many objects like this (in a basic solution), by interpreting these objects on time steps $3, 4, 5$ as a basic optimal solution of a $3-DKP$ (locally, i.e. with a variable y_i such that $x_{1i} \leq y_i \leq x_{5i}$).

Then, roughly speaking, the idea is to show that for any pair of time steps $t_0 \leq t_1$, we can bound the number of objects which are fractional and constant on this time interval $[t_0, t_1]$ (but not at time $t_0 - 1$ and $t_1 + 1$). Then a sum on all the possible choices of (t_0, t_1) gives the global upper bound.

Let us state this rough idea formally. In all this section, we consider a (feasible) solution $\hat{S} = (\hat{x}, \hat{z})$ of $(LP-MK)$, where $\hat{z}_{ti} = 1 - |\hat{x}_{(t+1)i} - \hat{x}_{ti}|$ (variables $\hat{z}_{ti}$ are set to their optimal value w.r.t. $\hat{x}$).

In such a solution $\hat{S} = (\hat{x}, \hat{z})$, let us define as previously an object as *fractional* if at least one variable $\hat{x}_{ti}$ or $\hat{z}_{ti}$ is fractional. Our goal is to show the following result.

▶ **Theorem 6.** *If $\hat{S} = (\hat{x}, \hat{z})$ is a basic solution of $(LP-MK)$, it has at most T^3 fractional objects.*

Before proving the theorem, let us introduce some definitions and show some lemmas. Let t_0, t_1 be two time steps with $1 \le t_0 \le t_1 \le T$.

▶ **Definition 7.** *The set $F(t_0,t_1)$ associated to $\hat{S} = (\hat{x}, \hat{z})$ is the set of objects i (called fractional w.r.t. (t_0,t_1)) such that*

- $0 < \hat{x}_{t_0 i} = \hat{x}_{(t_0+1)i} = \cdots = \hat{x}_{t_1 i} < 1$;
- *Either $t_0 = 1$ or $\hat{x}_{(t_0-1)i} \ne \hat{x}_{t_0 i}$;*
- *Either $t_1 = T$ or $\hat{x}_{(t_1+1)i} \ne \hat{x}_{t_1 i}$;*

In other words, we have $\hat{x}_{ti}$ fractional and constant on $[t_0,t_1]$, and $[t_0,t_1]$ is maximal w.r.t. this property.

For $t_0 \le t \le t_1$, we note C'_t the remaining capacity of knapsack at time t considering that variables outside $F(t_0,t_1)$ are fixed (to their value in $\hat{x}$):

$$C'_t = C_t - \sum_{i \notin F(t_0,t_1)} w_{ti}\hat{x}_{ti}.$$

As previously, we will see $x_{t_0 i}, \dots, x_{t_1 i}$ as a single variable y_i. We have to express the fact that this variable y_i cannot "cross" the values $\hat{x}_{(t_0-1)i}$ (if $t_0 > 1$) and $\hat{x}_{(t_1+1)i}$ (if $t_1 < T$), so that everything remains locally (in this range) linear. So we define the lower and upper bounds a_i, b_i induced by Definition 7 as:

- Initialize $a_i \leftarrow 0$. If $\hat{x}_{(t_0-1)i} < \hat{x}_{t_0 i}$ then do $a_i \leftarrow \hat{x}_{(t_0-1)i}$. If $\hat{x}_{(t_1+1)i} < \hat{x}_{t_1 i}$ then do $a_i \leftarrow \max(a_i, \hat{x}_{(t_1+1)i})$.
- Similarly, initialize $b_i \leftarrow 1$. If $\hat{x}_{(t_0-1)i} > \hat{x}_{t_0 i}$ then do $b_i \leftarrow \hat{x}_{(t_0-1)i}$. If $\hat{x}_{(t_1+1)i} > \hat{x}_{t_1 i}$ then do $b_i \leftarrow \min(b_i, \hat{x}_{(t_1+1)i})$.

Note that with this definition $a_i < \hat{x}_{t_0,i} < b_i$. This allows us to define the polyhedron $P(t_0,t_1)$ as the set of $y = (y_i : i \in F(t_0,t_1))$ such that

$$\begin{cases} \sum_{i \in F(t_0,t_1)} w_{ti} y_i \le C'_t & \forall t \in \{t_0, ..., t_1\} \\ a_i \le y_i \le b_i & \forall i \in F(t_0,t_1) \end{cases}$$

▶ **Definition 8.** *The solution $\hat{y}$ associated to $\hat{S} = (\hat{x}, \hat{z})$ is defined as $\hat{y}_i = \hat{x}_{t_0 i}$ for $i \in F(t_0,t_1)$.*

▶ **Lemma 9.** *If $\hat{S} = (\hat{x}, \hat{z})$ is a basic solution, then the solution $\hat{y}$ associated to $(\hat{x}, \hat{z})$ is feasible of $P(t_0,t_1)$ and basic.*

Proof. Since $(\hat{x}, \hat{z})$ is feasible, then $\hat{y}$ respects the capacity constraints (remaining capacity), and $a_i < \hat{y}_i = \hat{x}_{t_0 i} < b_i$ so $\hat{y}$ is feasible.

Suppose now that $\hat{y} = \frac{y^1+y^2}{2}$ for two feasible solutions $y^1 \ne y^2$ of $P(t_0,t_1)$. We associate to y^1 a feasible solution $S^1 = (x^1, z^1)$ as follows.

For any object i, we fix $x^1_{ti} = \hat{x}_i$ for $t \notin [t_0,t_1]$, and $x^1_{ti} = y^1_i$ for $t \in [t_0,t_1]$. We fix variables z^1_{it} to their maximal values, i.e. $z^1_{ti} = 1 - |x^1_{(t+1)i} - x^1_{ti}|$. This way, we get a feasible solution (x^1, z^1). Note that:

- $z^1_{ti} = \hat{z}_{ti}$ for $t \notin [t_0 - 1, t_1]$, since coresponding variables x are the same in S^1 and $\hat{S}$;
- $z^1_{ti} = 1 = \hat{z}_{ti}$ for $t \in [t_0, t_1 - 1]$, since variables x are constant on the interval $[t_0,t_1]$.

Then, for variables z, the only modifications between z^1 and $\hat{z}$ concern the "boundary" variables z^1_{ti} for $t = t_0 - 1$ and $t = t_1$.

We build this way two solutions $S^1 = (x^1, z^1)$ and $S^2 = (x^2, z^2)$ of $(LP - MK)$ corresponding to y^1 and y^2. By construction, S^1 and S^2 are feasible. They are also different provided that y^1 and y^2 are different. It remains to prove that $\hat{S} = (S^1 + S^2)/2$.

Let us first consider variables x:

- if $t \notin [t_0, t_1]$, $x^1_{ti} = x^2_{ti} = \hat{x}_{ti}$ so $\hat{x}_{ti} = \frac{x^1_{ti}+x^2_{ti}}{2}$.
- if $t \in [t_0, t_1]$, $x^1_{ti} = y^1_i$ and $x^2_{ti} = y^2_t$, so $\frac{x^1_{ti}+x^2_{ti}}{2} = \frac{y^1_i+y^2_i}{2} = \hat{y}_i = \hat{x}_{ti}$.

Now let us look at variables z: first, for $t \notin \{t_0 - 1, t_1\}$, $z^1_{ti} = z^2_{ti} = \hat{z}_{ti}$ so $\hat{z}_{ti} = \frac{z^1_{ti}+z^2_{ti}}{2}$. The last and main part concerns the last 2 variables $z_{(t_0-1)i}$ (if $t_0 > 1$) and $z_{t_1 i}$ (if $t_1 < T$).

We have $z^1_{(t_0-1)i} = 1 - |x^1_{t_0 i} - x^1_{(t_0-1)i}| = 1 - |x^1_{t_0 i} - \hat{x}_{(t_0-1)i}|$ and $\hat{z}_{(t_0-1)i} = 1 - |\hat{x}_{t_0 i} - \hat{x}_{(t_0-1)i}|$. The crucial point is to observe that thanks to the constraint $a_i \leq y_i \leq b_i$, and by definition of a_i and b_i, $x^1_{t_0,i}$, $x^2_{t_0,i}$ and $\hat{x}_{t_0,i}$ are either all greater than (or equal to) $\hat{x}_{(t_0-1)i}$, or all lower than (or equal to) $\hat{x}_{(t_0-1)i}$.

Suppose first that they are all greater than (or equal to) $\hat{x}_{(t_0-1)i}$. Then $z^1_{(t_0-1)i} - \hat{z}_{(t_0-1)i} = |\hat{x}_{t_0,i} - \hat{x}_{t_0-1,i}| - |x^1_{t_0,i} - \hat{x}_{t_0-1,i}| = \hat{x}_{t_0 i} - x^1_{t_0 i} = \hat{y}_i - y^1_i$.

Similarly, $z^2_{(t_0-1)i} - \hat{z}_{(t_0-1)i} = \hat{y}_i - y^2_i$. So $\frac{z^1_{(t_0-1)i}+z^2_{(t_0-1)i}}{2} = \frac{2\hat{z}_{(t_0-1)i}+2\hat{y}_i-y^1_i-y^2_i}{2} = \hat{z}_{(t_0-1)i}$.

Now suppose that they are all lower than (or equal to) $\hat{x}_{t_0-1,i}$. Then:

$$z^1_{(t_0-1)i} - \hat{z}_{(t_0-1)i} = |\hat{x}_{t_0 i} - \hat{x}_{(t_0-1)i}| - |x^1_{t_0 i} - \hat{x}_{(t_0-1)i}| = x^1_{t_0 i} - \hat{x}_{t_0 i} = y^1_i - \hat{y}_i$$

Similarly, $z^2_{(t_0-1)i} - \hat{z}_{(t_0-1)i} = y^2_i - \hat{y}_i$. So $\frac{z^1_{(t_0-1)i}+z^2_{(t_0-1)i}}{2} = \frac{2\hat{z}_{(t_0-1)i}-2\hat{y}_i+y^1_i+y^2_i}{2} = \hat{z}_{(t_0-1)i}$.

Then, in both cases, $\hat{z}_{(t_0-1)i} = \frac{z^1_{(t_0-1)i}+z^2_{(t_0-1)i}}{2}$.

With the very same arguments we can show that $\frac{z^1_{t_1 i}+z^2_{t_1 i}}{2} = \hat{z}_{t_1 i}$. Then, $\hat{S}$ is the half sum of S^1 and S^2, contradiction with the fact that $\hat{S}$ is basic. ◀

Now we can bound the number of fractional objects w.r.t. (t_0, t_1).

▶ **Lemma 10.** $|F(t_0, t_1)| \leq t_1 + 1 - t_0$.

Proof. $P(t_0, t_1)$ is a polyhedron corresponding to a linear relaxation of a $k - DLP$, with $k = t_1 + 1 - t_0$. Since $\hat{y}$ is basic, using Theorem 3 (and the note after) there are at most $k = t_1 + 1 - t_0$ variables $\hat{y}_i$ such that $a_i < \hat{y}_i < b_i$. But by definition of $F(t_0, t_1)$, for all $i \in F(t_0, t_1)$ $a_i < \hat{y}_i < b_i$. Then $|F(t_0, t_1)| \leq t_1 + 1 - t_0$. ◀

Now we can easily prove Theorem 6.

Proof. First note that if $\hat{x}_{ti}$ and $\hat{x}_{(t+1)i}$ are integral, then so is $\hat{z}_{ti}$. Then, if an object i is fractional at least one $\hat{x}_{ti}$ is fractional, and so i will appear in (at least) one set $F(t_0, t_1)$.

We consider all pairs (t_0, t_1) with $1 \leq t_0 \leq t_1 \leq T$. Thanks to Lemma 10, $|F(t_0, t_1)| \leq t_1 + 1 - t_0$. So, the total number of fractional objects is at most:

$$N_T = \sum_{t_0=1}^{T} \sum_{t_1=t_0}^{T} (t_1 + 1 - t_0) \leq T^3$$

Indeed, there are less than T^2 choices for (t_0, t_1) and at most T fractional objects for each choice. ◀

Note that with standard calculation we get $N_T = \frac{T^3+3T^2+2T}{6}$, so for $T = 2$ time steps $N_2 = 4$: we have at most 4 fractional objects, the same bound as in Proposition 5.

3.2 A PTAS for a constant number of time steps

Now we can describe the $PTAS$. Informally, the algorithm first guesses the ℓ objects with the maximum reward in an optimal solution (where ℓ is defined as a function of ϵ and T), and then finds a solution on the remaining instance using the relaxation of the LP. The fact that the number of fractional objects is small allows to bound the error made by the algorithm.

For a solution S (either fractional or integral) we define $g_i(S)$ as the reward of object i in solution S: $g_i(S) = \sum_{t=1}^{T} p_{ti}x_{ti} + \sum_{t=1}^{T-1} z_{ti}B_{ti}$. The value of a solution S is $g(S) = \sum_{i \in N} g_i(S)$.

Consider the algorithm A^{LP} which, on an instance of Multistage Knapsack:

- Finds an optimal (basic) solution $S^r = (x^r, z^r)$ of the relaxation $(LP - MK)$ of $(ILP - MK)$;
- Takes at step t an object i if and only if $x^r_{ti} = 1$.

Clearly, A^{LP} outputs a feasible solution, the value of which verifies:

$$g(A^{LP}) \geq g(S^r) - \sum_{i \in F} g_i(S^r) \tag{1}$$

where F is the set of fractional objects in S^r. Indeed, for each integral (i.e., not fractional) object the reward is the same in both solutions.

Now we can describe the algorithm **Algorithm $\boldsymbol{PTAS_{ConstantMK}}$**, which takes as input an instance of Multistage Knapsack and an $\epsilon > 0$, and works as follows.

1. Let $\ell := \min\left\{\left\lceil \frac{(T+1)T^3}{\epsilon} \right\rceil, n\right\}$.
2. For all $X \subseteq N$ such that $|X| = \ell$, $\forall X_1 \subseteq X, ..., \forall X_T \subseteq X$:
   If for all $t = 1, \ldots, T$ $w_t(X_t) = \sum_{j \in X_t} w_{tj} \leq C_t$, then:
   - Compute the rewards of object $i \in X$ in the solution $(X_1, \ldots, X_T)$, and find the smallest one, say k, with reward g_k.
   - On the subinstance of objects $Y = N \setminus X$:
     - For all $i \in Y$, for all $t \in \{1, \ldots, T\}$: if $p_{ti} > g_k$ then set $x_{ti} = 0$.
     - apply A^{LP} on the subinstance of objects Y, with the remaining capacity $C'_t = C_t - w_t(X_t)$, where some variables x_{ti} are set to 0 as explained in the previous step.
   - Let $(Y_1, ..., Y_T)$ be the sets of objects taken at time $1, \ldots, T$ by A^{LP}. Consider the solution $(X_1 \cup Y_1, ..., X_T \cup Y_T)$.
3. Output the best solution computed.

▶ **Theorem 11.** *The algorithm $PTAS_{ConstantMK}$ is a $(1 - \epsilon)$-approximation algorithm running in time $O\left(n^{O(T^5/\epsilon)}\right)$.*

Proof. (sketch) Let us briefly argue why the claimed ratio holds. We consider the iteration of the algorithm where X equals the set of ℓ objects which have maximum reward in an optimal solution S^*. By exhaustive search, the algorithm considers the case where $X_1, \ldots, X_T$ are exactly as in S^*, so the reward of the algorithm is the same as S^* for these objects. For the remaining objects, we use the relaxation of the linear program to build an integer solution. The loss corresponds to the rewards generated by fractional objects (see Equation 1) in the *fractional solution.*

While these profits could be very high, the preprocessing step fixing some variables with high profit to 0 (x_{ti} is set to 0 if $p_{ti} > g_k$) allows to bound the loss as (roughly) the reward of these fractional objects *in* S^*. Since there is a bounded number of fractional objects, the loss induced by their rewards can be bounded by a fraction ϵ of the optimal value, by choosing a sufficiently large $\ell = |X|$. ◀

3.3 Generalization to an arbitrary number of time steps

We now devise a PTAS for the general problem, for an arbitrary (not constant) number of steps. We actually show how to get such a PTAS provided that we have a PTAS for (any) constant number of time steps. Let A_{ϵ,T_0} be an algorithm which, given an instance of Multistage Knapsack with at most T_0 time steps, outputs a $(1-\epsilon)$-approximate solution in time $O(n^{f(\epsilon,T_0)})$ for some function f.

The underlying idea is to compute (nearly) optimal solutions on subinstances of bounded sizes, and then to combine them in such a way that at most a small fraction of the optimal value is lost.

Let us first give a rough idea of our algorithm $PTAS_{MK}$.

Given an $\epsilon > 0$, let $\epsilon' = \epsilon/2$ and $T_0 = \lceil \frac{1}{\epsilon'} \rceil$. We construct a set of solutions $S^1, \dots, S^{T_0}$ in the following way:

In order to construct S^1, we partition the time horizon $1, \dots, T$ into $\lceil \frac{T}{T_0} \rceil$ consecutive intervals. Every such interval has length T_0, except possibly the last interval that may have a smaller length. We apply A_{ϵ,T_0} at every interval in this partition. S^1 is then just the concatenation of the partial solutions computed for each interval.

The partition used to build the solution S^i, $1 < i \leq T_0$, is made in a similar way. The only difference is that the first interval of the partition of the time horizon $1, \dots, T$ goes from time 1 to time $i-1$. For the remaining part of the time horizon, i.e. for $i, \dots T$, the partition is made as previously, i.e. starting at time step i, every interval will have a length of T_0, except possibly the last one, whose length may be smaller. Once the partition is operated, we apply A_{ϵ,T_0} to every interval of the partition. S^i, $1 < i \leq T_0$, is then defined as the concatenation of the partial solutions computed on each interval. Among the T_0 solutions $S^1, \dots, S^{T_0}$, the algorithm chooses the best solution.

The construction is illustrated on Figure 1, with 10 time steps and $T_0 = 3$. The first solution S^1 is built by applying 4 times A_{ϵ,T_0}, on the subinstances corresponding to time steps $\{1,2,3\}, \{4,5,6\}, \{7,8,9\}$, and $\{10\}$. The solution S^2 is built by applying 4 times A_{ϵ,T_0}, on the subinstances corresponding to time steps $\{1\}, \{2,3,4\}, \{5,6,7\}$, and $\{8,9,10\}$.

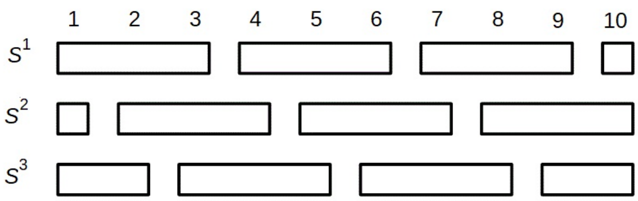

Figure 1 The three solutions for $T_0 = 3$ and $T = 10$.

▶ **Theorem 12.** *The algorithm $PTAS_{MK}$ is a polynomial time approximation algorithm.*

Let us give a rough idea of the proof. As we see from Figure 1, each solution S^t misses some potential transition profit between some time steps (as between 3 and 4, between 6 and 7, and between 9 and 10 for S^1). For each j, such loss between step j and $j+1$ appears in exactly one S^t, so in average we loose a fraction $1/T_0$ of the optimal transition profit (so a fraction at most ϵ'). Another loss is due to the fact that we use an approximation algorithm on the subinstances, inducing also a loss of at most a fraction ϵ' of the optimum value.

4 Pseudo-polynomiality and hardness results

We complement the previous result on approximation scheme by showing the following results for Multistage Knapsack. First, it does not admit an FPTAS (unless $P = NP$), as stated in the following Theorem.

▶ **Theorem 13.** *There is no $FPTAS$ for* Multistage Knapsack *unless $P = NP$, even if there are only two time steps and the bonus is uniform.*

The reduction is from Cardinality-2-KP, and the idea of the proof is to take a sufficiently large (but polynomially bounded) bonus in the multistage instance in order to force the knapsacks of the two time steps to be the same.

The second result states that the problem is pseudo-polynomial for a constant number of time steps. More precisely, with a standard dynamic programming procedure, we have the following.

▶ **Theorem 14.** Multistage Knapsack *is solvable in time $O(T(2C_{max} + 2)^T n)$ where $C_{max} = \max\{C_i, i = 1, \dots, T\}$.*

As a final result, we show that the problem is strongly NP-hard (when the number of steps is not bounded), by showing the NP-hardness of the following subproblem.

▶ **Definition 15.** Binary Multistage Knapsack *is the sub-problem of* Multistage Knapsack *where all the weights, profits and capacities are all equal to 0 or* 1*.*

For the usual Knapsack problem, the binary case corresponds to a trivial problem. For the multistage case, we have the following:

▶ **Theorem 16.** Binary Multistage Knapsack *is NP-hard, even in the case of uniform bonus.*

Proof. (sketch) We prove the result by a reduction from the Independent Set problem where, given a graph G and an integer K, we are asked if there exists a subset of K pairwise non adjacent vertices (called an independent set). This problem is NP-hard, see [13].

Let (G, K) be an instance of the Independent Set problem, with $G = (V, E)$, $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$. We build the following instance I' of Binary Multistage Knapsack:

- There are n objects $\{1, 2 \dots, n\}$, one object per vertex;
- There are $T = m$ time steps: each edge (v_i, v_j) in E corresponds to one time step;
- at the time step corresponding to edge (v_i, v_j): objects i and j have weight 1, while the others have weight 0, all objects have profit 1, and the capacity constraint is 1.
- The transition profit is $b_{ti} = B = 2nm$ for all i, t.

Roughly speaking, B is large enough to ensure that in an optimal solution there are no modifications of the knapsack over the time. Then we can show that there is an independent set of size (at least) K if and only if there is a solution for Binary Multistage Knapsack of value (at least) $n(m - 1)B + mK$. ◀

Since B is polynomially bounded in the proof, Multistage Knapsack is strongly NP-hard.

References

1 Susanne Albers. On Energy Conservation in Data Centers. In Christian Scheideler and Mohammad Taghi Hajiaghayi, editors, *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017*, pages 35–44. ACM, 2017. doi:10.1145/3087556.

2 Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic Facility Location via Exponential Clocks. *ACM Trans. Algorithms*, 13(2):21:1–21:20, 2017.

3 Barbara M. Anthony and Anupam Gupta. Infrastructure Leasing Problems. In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization, 12th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007, Proceedings*, volume 4513 of *Lecture Notes in Computer Science*, pages 424–438. Springer, 2007. doi:10.1007/978-3-540-72792-7.

4 Evripidis Bampis, Bruno Escoffier, Michael Lampis, and Vangelis Th. Paschos. Multistage Matchings. In David Eppstein, editor, *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden*, volume 101 of *LIPIcs*, pages 7:1–7:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.SWAT.2018.7.

5 Evripidis Bampis, Bruno Escoffier, and Sasa Mladenovic. Fair Resource Allocation Over Time. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar, editors, *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 766–773. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. URL: http://dl.acm.org/citation.cfm?id=3237496.

6 Nicolas K. Blanchard and Nicolas Schabanel. Dynamic Sum-Radii Clustering. In Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen, editors, *WALCOM: Algorithms and Computation, 11th International Conference and Workshops, WALCOM 2017, Hsinchu, Taiwan, March 29-31, 2017, Proceedings.*, volume 10167 of *Lecture Notes in Computer Science*, pages 30–41. Springer, 2017. doi:10.1007/978-3-319-53925-6_3.

7 Niv Buchbinder, Shahar Chen, and Joseph Naor. Competitive Analysis via Regularization. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 436–444. SIAM, 2014. doi:10.1137/1.9781611973402.32.

8 Niv Buchbinder, Shahar Chen, Joseph Naor, and Ohad Shamir. Unified Algorithms for Online Learning and Competitive Analysis. *Math. Oper. Res.*, 41(2):612–625, 2016. doi:10.1287/moor.2015.0742.

9 Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123(2):333–345, 2000.

10 Edith Cohen, Graham Cormode, Nick G. Duffield, and Carsten Lund. On the Tradeoff between Stability and Fit. *ACM Trans. Algorithms*, 13(1):7:1–7:24, 2016. doi:10.1145/2963103.

11 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility Location in Evolving Metrics. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 459–470. Springer, 2014. doi:10.1007/978-3-662-43951-7_39.

12 Alan M Frieze, MRB Clarke, et al. Approximation algorithms for the m-dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses. *European Journal of Operational Research*, 15:100–109, 1984.

13 Michael R Garey and David S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

14 Albert Gu, Anupam Gupta, and Amit Kumar. The Power of Deferral: Maintaining a Constant-Competitive Steiner Tree Online. *SIAM J. Comput.*, 45(1):1–28, 2016. doi:10.1137/140955276.

15 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing Bases: Multistage Optimization for Matroids and Matchings. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 563–575. Springer, 2014.

16 Oscar H. Ibarra and Chul E. Kim. Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems. *J. ACM*, 22(4):463–468, 1975. doi:10.1145/321906.321909.

17 Hans Kellerer and Ulrich Pferschy. A New Fully Polynomial Time Approximation Scheme for the Knapsack Problem. *J. Comb. Optim.*, 3(1):59–71, 1999. doi:10.1023/A:1009813105532.

18 Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, Berlin, Germany, 2004.

19 Bernhard Korte and Rainer Schrader. On the existence of fast approximation schemes. In O. Magasarian, R. Meyer, and S. Robinson, editors, *Nonlinear Programming 4*, pages 415–437. Academic Press, 1981.

20 Eugene L. Lawler. Fast Approximation Algorithms for Knapsack Problems. *Math. Oper. Res.*, 4(4):339–356, 1979. doi:10.1287/moor.4.4.339.

21 Michael J. Magazine and Osman Oguz. A fully polynomial approximation scheme for the 0-1 knapsack problem. *European Journal of Operational Research*, 8:270–273, 1981.

22 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The Power of Recourse for Online MST and TSP. *SIAM J. Comput.*, 45(3):859–880, 2016. doi:10.1137/130917703.

23 Chandrashekhar Nagarajan and David P Williamson. Offline and online facility leasing. *Discrete Optimization*, 10(4):361–370, 2013.

24 Osman Oguz and Michael J. Magazine. A polynomial time approximation algorithm for the multidimensional knapsack problem. *Working paper, University of Waterloo*, 1980.

25 Cécile Rottner. *Combinatorial Aspects of the Unit Commitment Problem*. PhD thesis, Sorbonne Université, 2018.

26 Baruch Schieber, Hadas Shachnai, Gal Tamir, and Tami Tamir. A Theory and Algorithms for Combinatorial Reoptimization. *Algorithmica*, 80(2):576–607, 2018. doi:10.1007/s00453-017-0274-8.