



HAL
open science

Multistage knapsack

Evripidis Bampis, Bruno Escoffier, Alexandre Teiller

► **To cite this version:**

Evripidis Bampis, Bruno Escoffier, Alexandre Teiller. Multistage knapsack. Journal of Computer and System Sciences, 2022, 126, pp.106-118. 10.1016/j.jcss.2022.01.002 . hal-03660984

HAL Id: hal-03660984

<https://hal.science/hal-03660984v1>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Multistage Knapsack^{*}

Evipidis Bampis

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Bruno Escoffier

*Sorbonne Université, CNRS, LIP6, F-75005 Paris, France, and
Institut Universitaire de France*

Alexandre Teiller

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

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 incurred for transitioning from one state to another. In order to model such situations, we look at a recently introduced 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 are both (i) near optimal for each time step and (ii) as stable as possible. 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.

Keywords: Multistage Optimization, Knapsack, Complexity, Approximation Algorithms

^{*}An extended abstract of this work appeared in the proceedings of MFCS 2019 [1].

Email addresses: evripidis.bampis@lip6.fr (Evripidis Bampis),
bruno.escoffier@lip6.fr (Bruno Escoffier), alexandre.teiller@lip6.fr (Alexandre Teiller)

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, Gupta et al. [2] and Eisenstat et al. [3] 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. [2] 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 $N = \{u_1, \dots, u_n\}$ 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 $t = 1, 2, \dots, T$ 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 [4], or a machine in a data center [5]. 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 consecutive days. We refer the reader interested in planning problems in electricity production to [4].

We formalize the problem as follows. We are given a time horizon $t = 1, 2, \dots, T$, and a sequence of knapsack instances I_1, I_2, \dots, 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 set of objects 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*¹. We are asked to produce a sequence of solutions S_1, S_2, \dots, 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. the MULTISTAGE SPANNING TREE problem [2], or the MULTISTAGE BIPARTITE PERFECT MATCHING problem [6]).

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 $N = \{1, 2, \dots, n\}$ of objects;*
- *For any $t \in \{1, \dots, T\}$, 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 $i \in N$:*

¹One may want to consider penalties for modifying decisions. However, dealing with a maximization problem, introducing penalties would lead to a mixed-sign objective, which may, in turn, prevent to reach multiplicative approximation guarantees.

- $B_{ti} \in \mathbb{R}^+$ 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 chosen sets of objects;
- 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 (re-optimize) by making as few changes as possible. We refer the reader to [7, 8, 9, 10, 11, 12] and the references therein.

Multistage optimization has been studied for fractional problems by Buchbinder et al. [13] and Buchbinder, Chen and Naor [14]. The multistage model considered in this article is the one studied in Eisenstat et al. [3] and Gupta et al. [2]. Eisenstat et al. [3] studied the multistage version of facility location problems. They proposed a logarithmic approximation algorithm. An et al. [15] obtained constant factor approximation for some related problems. Gupta et al. [2] 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. [6] 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 show that the maximization version of the problem admits a constant factor approximation algorithm but is APX-hard. In another work [16], 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 provide constant factor approximation algorithms for the off-line setting. Regarding the on-line setting, we refer the reader to [17] where some approximability results are shown for a variety of subset maximization problems in a multistage framework. Finally, in a very recent work [18], the MULTISTAGE VERTEX COVER problem was studied with a complexity parameterized angle.

Knapsack variants.

Our work builds upon the KNAPSACK literature [19]. It is well known that there is a simple 2-approximation algorithm as well as a fully polynomial time (FPTAS) for the static case [20, 21, 22, 23]. 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 $N = \{1, 2, \dots, n\}$ of objects. Each object i has a profit p_i and k weights w_{ji} , $j = 1, \dots, k$. 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. [24] showed that this can be generalized to ($k - DKP$).

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

$$\left\{ \begin{array}{l} \max \sum_{i \in N} p_i y_i \\ \text{s.t.} \left| \begin{array}{l} \sum_{i \in N} w_{ji} y_i \leq C_j \quad \forall j \in \{1, \dots, k\} \\ y_i \in \{0, 1\} \quad \forall i \in N \end{array} \right. \end{array} \right.$$

Theorem 1. [24] *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.*

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 $a_i < y_i < b_i$.*

[24] use the result of Theorem 1 to show that for any fixed constant k (k -DKP) admits a polynomial time approximation scheme (PTAS). Other PTASes have been presented in [25, 26]. Korte and Schrader [27] 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 [19].

Theorem 2. [19] *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 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* when $B_{ti} = B$ for all $i \in N$ and all $t = 1, \dots, T - 1$.

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*):

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

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 [24], 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) brute-force 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 allows 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 $S = (S_1, \dots, S_T)$ (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 1. *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 transferring at time 1 an amount ϵ of weight from i to j (and adjusting consequently z_i and z_j). Namely, in S_1 :

- $x_{1i}^1 = \hat{x}_{1i} - \frac{\epsilon}{w_{1i}}$, $z_i^1 = \hat{z}_i - d_i \frac{\epsilon}{w_{1i}}$, 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 $\hat{x}_{2i} \neq \hat{x}_{1i}$).
- $x_{1j}^1 = \hat{x}_{1j} + \frac{\epsilon}{w_{1j}}$, $z_j^1 = \hat{z}_j + d_j \frac{\epsilon}{w_{1j}}$, 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 $|\hat{x}_{2i} - \hat{x}_{1i}|$ decreases by δ so \hat{z}_i can increase by δ (so $d_i = 1$), and if $\hat{x}_{2i} < \hat{x}_{1i}$ 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 :

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

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 if $\hat{x}_{1i} < \hat{x}_{2i}$ then x_{1i} must be in $[0, \hat{x}_{2i}]$ (in S^1 , x_{1i}^1 cannot be larger than \hat{x}_{2i} , otherwise the previous value of z_i^1 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_{1i} 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

$$C'_t = C_t - \sum_{j \in P} w_{tj} \hat{x}_{tj}, \text{ and "imposing" } x_{1i} = x_{2i}: \left\{ \begin{array}{l} \max \sum_{i \in L} (p_{1i} + p_{2i}) y_i + \sum_{i \in L} B_{1i} \\ \sum_{i \in L} w_{1i} y_i \leq C'_1 \\ \sum_{i \in L} w_{2i} y_i \leq C'_2 \\ y_i \in [0, 1] \quad \forall i \in L \end{array} \right.$$

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 that this is not the case, i.e., $\hat{S}_L = \frac{S_L^1 + S_L^2}{2}$, 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 [24], \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. \square

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 1, 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 1, 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 3. *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 $1 \leq t_0 \leq t_1 \leq T$.

Definition 3. *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*

- $0 < \hat{x}_{t_0i} = \hat{x}_{(t_0+1)i} = \dots = \hat{x}_{t_1i} < 1$;
- Either $t_0 = 1$ or $\hat{x}_{(t_0-1)i} \neq \hat{x}_{t_0i}$;
- Either $t_1 = T$ or $\hat{x}_{(t_1+1)i} \neq \hat{x}_{t_1i}$.

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 $t_0 \leq t \leq t_1$, 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 $x_{t_0i}, \dots, x_{t_1i}$ 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 3 as:

- Initialize $a_i \leftarrow 0$. If $\hat{x}_{(t_0-1)i} < \hat{x}_{t_0i}$ then do $a_i \leftarrow \hat{x}_{(t_0-1)i}$. If $\hat{x}_{(t_1+1)i} < \hat{x}_{t_1i}$ then do $a_i \leftarrow \max(a_i, \hat{x}_{(t_1+1)i})$.
- Similarly, initialize $b_i \leftarrow 1$. If $\hat{x}_{(t_0-1)i} > \hat{x}_{t_0i}$ then do $b_i \leftarrow \hat{x}_{(t_0-1)i}$. If $\hat{x}_{(t_1+1)i} > \hat{x}_{t_1i}$ 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

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

Definition 4. *The solution associated to $\hat{S} = (\hat{x}, \hat{z})$ is \hat{y} defined as $\hat{y}_i = \hat{x}_{t_0i}$ for $i \in F(t_0, t_1)$.*

Lemma 1. *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 $a_i < \hat{y}_i = \hat{x}_{t_0i} < b_i$ so \hat{y} is feasible.

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

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

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

- $z_{ti}^1 = 1 = \hat{z}_{ti}$ 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} concerns the “boundary” variables z_{ti}^1 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} is the half sum of S^1 and S^2 .

Let us first consider variables x :

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

Now let us look at variables z : first, for $t \notin \{t_0 - 1, t_1\}$, $z_{ti}^1 = z_{ti}^2 = \hat{z}_{ti}$ so $\hat{z}_{ti} = \frac{z_{ti}^1 + z_{ti}^2}{2}$. The last and main part concerns the last 2 variables $z_{(t_0-1)i}$ (if $t_0 > 1$) and z_{t_1i} (if $t_1 < T$).

We have $z_{(t_0-1)i}^1 = 1 - |x_{t_0i}^1 - x_{(t_0-1)i}^1| = 1 - |x_{t_0i}^1 - \hat{x}_{(t_0-1)i}|$ and $\hat{z}_{(t_0-1)i} = 1 - |\hat{x}_{t_0i} - \hat{x}_{(t_0-1)i}|$. 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 , $x_{t_0i}^1, x_{t_0i}^2$ and \hat{x}_{t_0i} 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:

$$z_{(t_0-1)i}^1 - \hat{z}_{(t_0-1)i} = |\hat{x}_{t_0i} - \hat{x}_{(t_0-1)i}| - |x_{t_0i}^1 - \hat{x}_{(t_0-1)i}| = \hat{x}_{t_0i} - x_{t_0i}^1 = \hat{y}_i - y_i^1.$$

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

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

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

$$z_{(t_0-1)i}^1 - \hat{z}_{(t_0-1)i} = |\hat{x}_{t_0i} - \hat{x}_{(t_0-1)i}| - |x_{t_0i}^1 - \hat{x}_{(t_0-1)i}| = x_{t_0i}^1 - \hat{x}_{t_0i} = y_i^1 - \hat{y}_i.$$

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

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

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

With the very same arguments we can show that $\frac{z_{t_1i}^1 + z_{t_1i}^2}{2} = \hat{z}_{t_1i}$.

Then, \hat{S} is the half sum of S^1 and S^2 , contradiction with the fact that \hat{S} is basic. \square

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

Lemma 2. $|F(t_0, t_1)| \leq t_1 + 1 - t_0$.

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 1 (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 $|F(t_0, t_1)| \leq t_1 + 1 - t_0$. \square

Now we can easily prove Theorem 3.

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 2, $|F(t_0, t_1)| \leq t_1 + 1 - t_0$. 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. \square

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 1.

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 ($ILP - MK$) 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 $x_{ti}^r = 1$.

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

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

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** $PTAS_{ConstantMK}$, which takes as input an instance of MULTISTAGE KNAPSACK and an $\epsilon > 0$.

Algorithm $PTAS_{ConstantMK}$

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 $|X| = \ell, \forall X_1 \subseteq X, \dots, \forall X_T \subseteq X$:
If for all $t = 1, \dots, T$ $w_t(X_t) = \sum_{j \in X_t} w_{tj} \leq C_t$, then:
 - Compute the rewards of object $i \in X$ in the solution (X_1, \dots, 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 $t \in \{1, \dots, T\}$: 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, \dots, Y_T) be the sets of objects taken at time $1, \dots, T$ by A^{LP} . Consider the solution $(X_1 \cup Y_1, \dots, X_T \cup Y_T)$.
3. Output the best solution computed.

Theorem 4. *The algorithm $PTAS_{ConstantMK}$ is a $(1 - \epsilon)$ -approximation algorithm running in time $O\left(n^{O(T^5/\epsilon)}\right)$.*

Proof. First, there are $O(n^\ell)$ choices for X ; for each X there are 2^ℓ choices for each X_t , so in all there are $O(n^\ell 2^{\ell T})$ choices for (X_1, \dots, X_T) . For each choice (X_1, \dots, X_T) , we compute the reward of elements, and then apply A^{LP} . Since $\ell \leq \lceil T(T+1)^3/\epsilon \rceil$, the running time follows.

Now let us show the claimed approximation ratio. Consider an optimal solution S^* , and suppose wlog that $g_i(S^*)$ are in non increasing order. Consider the iteration of the algorithm where $X = \{1, 2, \dots, \ell\}$. At this iteration, consider the choice (X_1, \dots, X_T) where X_t is exactly the subset of objects in X taken by S^* at time t (for $t = 1, \dots, T$). The solution S computed by the algorithm at this iteration (with X and (X_1, \dots, X_T)) is $S = (X_1 \cup Y_1, \dots, X_T \cup Y_T)$ where (Y_1, \dots, Y_T) is the solution output by A^{LP} on the subinstance of objects $Y = N \setminus X$, where x_{ti} is set to 0 if $p_{ti} > g_k$.

Note that since we consider the iteration where X corresponds to the ℓ objects of largest reward in the optimal solution, we do know that for an object $i > \ell$, if $p_{ti} > g_k$ then the optimal solution does not take object i at time t (it would have a reward greater than g_k), so we can safely fix this variable to 0. The idea behind putting these variables x_{ti} to 0 is to prevent the relaxed solution to take fractional objects with very large profits. These objects could indeed induce a very high loss when rounding the fractional solution to an integral one as done by A^{LP} . By doing this, the number of fractional objects (i.e., objects in F) does not increase. Indeed, if we put a variable x_{ti} at 0, it is not fractional so nothing changes in the proof of Theorem 3.

The value of S^* is $g(S^*) = \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^*)$. Thus, by equation 1, we have:

$$\begin{aligned} g(S) &\geq \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^r) - \sum_{i \in F} g_i(S^r) \\ &\geq \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^*) - \sum_{i \in F} g_i(S^r), \end{aligned}$$

where F is the set of fractional objects in the optimal fractional solution of the relaxation of the LP on Y , where x_{ti} is set to 0 if $p_{ti} > g_k$.

Each object of F has a profit at most g_k at any time steps and a transition profit at most $\sum_{t=1}^{T-1} B_{ti}$, so

$$\forall i \in F \quad g_i(S^r) \leq Tg_k + \sum_{t=1}^{T-1} B_{ti}.$$

Now, note that in an optimal solution each object has a reward at least $\sum_{t=1}^{T-1} B_{ti}$ (otherwise it simply never takes this object), so for all $i \in F$ $g_k \geq g_i \geq \sum_{t=1}^{T-1} B_{ti}$. So we have :

$$\forall i \in F \quad g_i(S^r) \leq (T+1)g_k$$

Since $g_i(S^*)$ are in non increasing order, we have $g_k = g_\ell(S^*) \leq \frac{\sum_{i \in X} g_i(S^*)}{\ell}$. So for all objects i in F , $g_i(S^r) \leq \frac{(T+1)\sum_{i \in X} g_i(S^*)}{\ell}$. By Theorem 3, there are at most T^3 of them, thus :

$$\begin{aligned} g(S) &\geq \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^*) - \sum_{i \in F} g_i(S^r) \\ &\geq \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^*) - \frac{(T+1)T^3}{\ell} \sum_{i \in X} g_i(S^*) \\ &\geq (1-\epsilon) \sum_{i \in X} g_i(S^*) + \sum_{i \in Y} g_i(S^*) \geq (1-\epsilon)g(S^*). \end{aligned}$$

□

By Theorem 4, for any fixed number of time steps T , MULTISTAGE KNAPSACK admits a PTAS.

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 on which it is based the construction of 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 $\{8, 9, 10\}$.

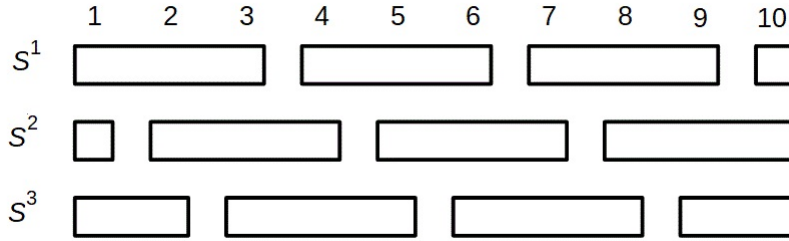


Figure 1: The three solutions for $T_0 = 3$ and $T = 10$.

More formally, given an $\epsilon > 0$, the algorithm $PTAS_{MK}$ works as follows.

- Let $\epsilon' = \epsilon/2$ and $T_0 = \lceil \frac{1}{\epsilon'} \rceil$. Let $\mathcal{I}_t = [t, \dots, t + T_0 - 1] \cap [1, \dots, T]$ be

the set of (at most) T_0 consecutive time steps starting at t . We consider \mathcal{I}_t for any t for which it is non empty (so $t \in [-T_0 + 2, \dots, T]$).

- For $t \in \{1, \dots, T_0\}$:
 - Apply A_{ϵ', T_0} on all intervals $\mathcal{I}_{t'}$ with $t' \equiv t \pmod{T_0}$. Note that each time step belongs to exactly one of such intervals.
 - Define the solution S^t built from the partial solutions given by the applications of A_{ϵ', T_0} .
- Choose the best solution S among the T_0 solutions S^1, \dots, S^{T_0} .

Theorem 5. *The algorithm $PTAS_{MK}$ is a polynomial time approximation scheme.*

Proof. The algorithm calls the A_{ϵ', T_0} algorithm $\lceil T/T_0 \rceil$ times for each of the T_0 generated solutions. Yet, the running time of A_{ϵ', T_0} is $n^{f(\frac{1}{\epsilon'}, T_0)} = n^{f(\frac{2}{\epsilon}, \lceil \frac{2}{\epsilon} \rceil)}$, i.e a polynomial time for any fixed ϵ . So, the running time of the algorithm for any T is

$$T_0 \times \left\lceil \frac{T}{T_0} \right\rceil n^{f(\frac{2}{\epsilon}, \frac{2}{\epsilon})} = O(Tn^{f(\frac{2}{\epsilon}, \frac{2}{\epsilon})}), \text{ a polynomial time for any fixed } \epsilon.$$

Each solution S^t of the T_0 generated solutions may loose some bonus between the last time step of one of its intervals \mathcal{I}_{t+kT_0} and the first time step of its next interval $\mathcal{I}_{t+(k+1)T_0}$ (in Figure 1 for instance, S^1 misses the bonuses between steps 3 and 4, 6 and 7, and 9 and 10). Let $loss(S^t)$ be this loss with respect to some optimal solution S^* . Since we apply A_{ϵ', T_0} to build solutions, we get that the value $g(S^t)$ of S^t is such that:

$$g(S^t) \geq (1 - \epsilon')g(S^*) - loss(S^t).$$

Since the output solution S is the best among the solutions S^t , by summing up the previous inequality for $t = 1, \dots, T_0$ we get:

$$g(S) \geq (1 - \epsilon')g(S^*) - \frac{\sum_{t=1}^{T_0} loss(S^t)}{T_0}.$$

Now, by construction the bonus between steps j and $j + 1$ appears in the loss of exactly one S^t (see Figure 1). So the total loss of the T_0 solutions is the global transition bonus of S^* , so at most $g(S^*)$. Hence:

$$\begin{aligned}
g(S) &\geq (1 - \epsilon')g(S^*) - \frac{g(S^*)}{T_0} \\
&\geq (1 - 2\epsilon')g(S^*) = (1 - \epsilon)g(S^*).
\end{aligned}$$

□

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 $P = NP$), even if there are only two time steps (Section 4.1) and the bonus is uniform ($B_{ti} = B$ for all i , all t);
- Second, the problem is pseudo-polynomial if the number of time steps T is a fixed constant (Section 4.2) but is strongly NP -hard in the general case even in the case of uniform bonus (Section 4.3).

4.1. No FPTAS

Theorem 6. *There is no FPTAS for MULTISTAGE KNAPSACK unless $P = NP$, even if there are only two time steps and the bonus is uniform.*

Proof. We prove the result by a reduction from CARDINALITY($2 - KP$), known to be NP -complete (Theorem 2)

For an instance I of CARDINALITY($2 - KP$), we consider the following instance I' of MULTISTAGE KNAPSACK:

- There are $T = 2$ time steps, and the same set of objects $N = \{1, 2, \dots, n\}$ as in I .
- The weights w_{1i} and w_{2i} are the same as in I , for all $i \in N$.
- $p_{1i} = p_{2i} = 1$ for all $i \in N$.
- $B_{1i} = 2$ for all $i \in N$.

We show that the answer to I is Yes (we can find K objects fulfilling the 2 knapsack constraints) if and only if there is a solution of value at least $2K + 2n$ for the instance I' of MK .

If we have a set A of K objects for I , then we simply take these objects at both time steps in I' . This is feasible, the knapsack revenue is $2K$ and the transition revenue is $2n$.

Conversely, consider an optimal solution of value at least $2K + 2n$ in I' . In an optimal solution, we necessarily have $x_{1j} = x_{2j}$, i.e an object is taken at the time 1 if and only if it is taken at the time 2. Indeed, assume that there is one $i \in N$ such that $x_{1i} \neq x_{2i}$, then consider the solution where $x_{1i} = x_{2i} = 0$. This is still feasible; the knapsack revenue reduces by 1 but the transition revenue increases by 2, contradiction. Thus the same set of objects A is taken at both time steps. Since the value of the solution is at least $2K + 2n$, the size of A is at least K . Hence the answer to I is Yes.

Note that in I' the optimal value is at most $4n$. We produce in this reduction polynomially bounded instances of MULTISTAGE KNAPSACK (with only two time steps), so the problem does not admit an FPTAS. Indeed, suppose that there is an FPTAS producing a $(1 - \epsilon)$ -approximation in time $p(1/\epsilon, n)$, for some polynomial p . Let $\epsilon = \frac{1}{4n+1}$. If we apply the FPTAS with this value of ϵ on I' we get a solution S of value at least $(1 - \epsilon)OPT(I') \geq OPT(I') - \frac{OPT(I')}{4n+1} > OPT(I') - 1$. Yet all the possible values are integers so S is optimal. The running time is polynomial in n , impossible unless $P = NP$. \square

So, if $P \neq NP$, the problem does not admit an FPTAS even if only the weight constraint is allowed to change over time. We note that, in contrast, if both the profits and the weight constraints are the same over time, i.e., $p_{ti} = p_i$, $w_{ti} = w_i$ and $C_t = C$, then the multistage problem reduces to the classical knapsack and then admits an FPTAS. Indeed, we only have to compute a $(1 + \epsilon)$ solution for one stage (all stages are the same) using an FPTAS algorithm for the classical knapsack problem, and replicate it all along the process - thus getting both full bonus and $(1 + \epsilon)$ -approximate profit. The existence of an FPTAS in the remaining case where the weight constraints are the same but the profits and/or bonuses may vary over time is left as an open question.

4.2. Pseudo-polynomiality for a constant number of time steps

We show here that the pseudo-polynomiality of the KNAPSACK problem generalizes to MULTISTAGE KNAPSACK when the number of time steps is constant. More precisely, with a standard dynamic programming procedure, we have the following.

Theorem 7. MULTISTAGE KNAPSACK is solvable in time $O(T(2C_{max}+2)^T n)$ where $C_{max} = \max\{C_i, i = 1, \dots, T\}$.

Proof. For any T -tuple (c_1, \dots, c_T) where $0 \leq c_i \leq C_i$, and any $s \in \{0, \dots, n\}$, we define $\alpha(c_1, \dots, c_T, s)$ to be the best value of a solution $S = (S_1, \dots, S_T)$ such that:

- The weight of knapsack at time t is at most c_t : for any t , $\sum_{i \in S_t} w_{ti} \leq c_t$;
- The solution uses only objects among the first s : for any t , $S_t \subseteq \{1, \dots, s\}$.

The optimal value of MULTISTAGE KNAPSACK is then $\alpha(C_1, \dots, C_T, n)$. We compute α by increasing values of s . For $s = 0$, we cannot take any object so $\alpha(c_1, \dots, c_T, 0) = 0$.

Take now $s \geq 1$. To compute $\alpha(c_1, \dots, c_T, s)$, we simply consider all the 2^T possibilities for taking or not object s in the T time steps. Let $A \subseteq \{1, \dots, T\}$ be a subset of time steps. If we take object s at time steps in A (and only there), we first check if A is a *valid* choice, i.e., $w_{ts} \leq c_t$ for any $t \in A$; then we can compute in $O(T)$ the corresponding reward $r_s(A)$ ($\sum_{t \in A} p_{ts}$ plus the transition bonus). We have:

$$\alpha(c_1, \dots, c_T, s) = \max\{r_s(A) + \alpha(c_1 - w_{1s}, \dots, c_T - w_{Ts}, s - 1) : A \subseteq \{1, \dots, T\} \text{ valid}\}.$$

The running time to compute one value of α is $O(T2^T)$. There are $O(n \prod_{i=1}^T (C_i + 1)) = O(n(C_{max} + 1)^T)$ values to compute, so the running time follows. A standard backward procedure allows to recovering the solution. \square

4.3. Strongly NP-hardness

Definition 5. BINARY MULTISTAGE KNAPSACK is the sub-problem of the 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 8. BINARY MULTISTAGE KNAPSACK is NP-hard, even in the case of uniform bonus.

Proof. 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 well known to be NP-hard, see [28].

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 .

We claim 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) $n(m-1)B + mK$.

Suppose first that there is an independent set V' of size at least K . We take the K objects corresponding to V' at all time steps. This is feasible since we cannot take 2 objects corresponding to one edge. The built solution sequence has knapsack profit mK and transition profit $n(m-1)B$ (no modification).

Conversely, take a solution of BINARY MULTISTAGE KNAPSACK of value at least $n(m-1)B + mK$. Since $B = 2nm$, there must be no modification of the knapsack over the time. Indeed, otherwise, the transition profit would be at most $n(m-1)B - B$, while the knapsack profit is at most mn , so the value would be less than $n(m-1)B$. So we consider the set of vertices corresponding to this knapsack. Thanks to the value of the knapsack, it has size at least K . Thanks to the capacity constraints of the bags, this is an independent set. \square

Since B is polynomially bounded in the proof, this shows that MULTISTAGE KNAPSACK is strongly NP -hard.

Acknowledgements

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

References

- [1] E. Bampis, B. Escoffier, A. Teiller, Multistage knapsack, in: P. Rossmanith, P. Heggernes, J. Katoen (Eds.), 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany, Vol. 138 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 22:1–22:14. doi:10.4230/LIPIcs.MFCS.2019.22.
URL <https://doi.org/10.4230/LIPIcs.MFCS.2019.22>
- [2] A. Gupta, K. Talwar, U. Wieder, Changing bases: Multistage optimization for matroids and matchings, in: J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias (Eds.), Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, Vol. 8572 of Lecture Notes in Computer Science, Springer, 2014, pp. 563–575.
- [3] D. Eisenstat, C. Mathieu, N. Schabanel, Facility location in evolving metrics, in: J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias (Eds.), Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II, Vol. 8573 of Lecture Notes in Computer Science, Springer, 2014, pp. 459–470.
- [4] C. Rottner, Combinatorial aspects of the unit commitment problem, Ph.D. thesis, Sorbonne Université (2018).
- [5] S. Albers, On energy conservation in data centers, in: Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017, ACM, 2017, pp. 35–44. doi:10.1145/3087556.
URL <https://doi.org/10.1145/3087556>

- [6] E. Bampis, B. Escoffier, M. Lampis, V. T. Paschos, Multistage matchings, in: D. Eppstein (Ed.), 16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden, Vol. 101 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 7:1–7:13.
- [7] B. M. Anthony, A. Gupta, Infrastructure leasing problems, in: M. Fischetti, D. P. Williamson (Eds.), Integer Programming and Combinatorial Optimization, 12th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007, Proceedings, Vol. 4513 of Lecture Notes in Computer Science, Springer, 2007, pp. 424–438.
- [8] N. K. Blanchard, N. Schabanel, Dynamic sum-radii clustering, in: S. Poon, M. S. Rahman, H. Yen (Eds.), WALCOM: Algorithms and Computation, 11th International Conference and Workshops, WALCOM 2017, Hsinchu, Taiwan, March 29-31, 2017, Proceedings., Vol. 10167 of Lecture Notes in Computer Science, Springer, 2017, pp. 30–41.
- [9] E. Cohen, G. Cormode, N. G. Duffield, C. Lund, On the tradeoff between stability and fit, *ACM Trans. Algorithms* 13 (1) (2016) 7:1–7:24.
- [10] A. Gu, A. Gupta, A. Kumar, The power of deferral: Maintaining a constant-competitive steiner tree online, *SIAM J. Comput.* 45 (1) (2016) 1–28.
- [11] N. Megow, M. Skutella, J. Verschae, A. Wiese, The power of recourse for online MST and TSP, *SIAM J. Comput.* 45 (3) (2016) 859–880.
- [12] C. Nagarajan, D. P. Williamson, Offline and online facility leasing, *Discrete Optimization* 10 (4) (2013) 361–370.
- [13] N. Buchbinder, S. Chen, J. Naor, O. Shamir, Unified algorithms for online learning and competitive analysis, *Math. Oper. Res.* 41 (2) (2016) 612–625.
- [14] N. Buchbinder, S. Chen, J. Naor, Competitive analysis via regularization, in: C. Chekuri (Ed.), Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, SIAM, 2014, pp. 436–444.

- [15] H. An, A. Norouzi-Fard, O. Svensson, Dynamic facility location via exponential clocks, *ACM Trans. Algorithms* 13 (2) (2017) 21:1–21:20.
- [16] E. Bampis, B. Escoffier, S. Mladenovic, Fair resource allocation over time, in: E. André, S. Koenig, M. Dastani, G. Sukthankar (Eds.), *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018, International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018*, pp. 766–773.
- [17] E. Bampis, B. Escoffier, K. Schewior, A. Teiller, Online multistage subset maximization problems, in: *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany., 2019*, pp. 11:1–11:14. doi:10.4230/LIPIcs.ESA.2019.11. URL <https://doi.org/10.4230/LIPIcs.ESA.2019.11>
- [18] T. Fluschnik, R. Niedermeier, V. Rohm, P. Zschoche, Multistage vertex cover, *arXiv preprint arXiv:1906.00659* (2019).
- [19] H. Kellerer, U. Pferschy, D. Pisinger, *Knapsack Problems*, Springer, Berlin, Germany, 2004.
- [20] O. H. Ibarra, C. E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *J. ACM* 22 (4) (1975) 463–468.
- [21] E. L. Lawler, Fast approximation algorithms for knapsack problems, *Math. Oper. Res.* 4 (4) (1979) 339–356.
- [22] M. J. Magazine, O. Oguz, A fully polynomial approximation scheme for the 0-1 knapsack problem., *European Journal of Operational Research* 8 (1981) 270–273.
- [23] H. Kellerer, U. Pferschy, A new fully polynomial time approximation scheme for the knapsack problem, *J. Comb. Optim.* 3 (1) (1999) 59–71.
- [24] A. Caprara, H. Kellerer, U. Pferschy, D. Pisinger, Approximation algorithms for knapsack problems with cardinality constraints, *European Journal of Operational Research* 123 (2) (2000) 333–345.
- [25] O. Oguz, M. J. Magazine, A polynomial time approximation algorithm for the multidimensional knapsack problem, Working paper, University of Waterloo (1980).

- [26] A. M. Frieze, M. Clarke, et al., Approximation algorithms for the m -dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses, *European Journal of Operational Research* 15 (1984) 100–109.
- [27] B. Korte, R. Schrader, On the existence of fast approximation schemes, in: O. Magasarian, R. Meyer, S. Robinson (Eds.), *Nonlinear Programming* 4, Academic Press, 1981, pp. 415–437.
- [28] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

As all the authors of this article equally participated to all aspects of this article, we would not like to make specific credit author statements.