General Knapsack Problems in a Dynamic Setting

Yaron Fairstein
Computer Science Department, Technion, Haifa, Israel

Ariel Kulik
Computer Science Department, Technion, Haifa, Israel

Joseph (Seffi) Naor
Computer Science Department, Technion, Haifa, Israel

Danny Raz
Computer Science Department, Technion, Haifa, Israel

Abstract

The world is dynamic and changes over time, thus any optimization problem used to model real life problems must address this dynamic nature, taking into account the cost of changes to a solution over time. The multistage model was introduced with this goal in mind. In this model we are given a series of instances of an optimization problem, corresponding to different times, and a solution is provided for each instance. The strive for obtaining near-optimal solutions for each instance on one hand, while maintaining similar solutions for consecutive time units on the other hand, is quantified and integrated into the objective function. In this paper we consider the Generalized Multistage d-Knapsack problem, a generalization of the multistage variants of the Multiple Knapsack problem, as well as the d-Dimensional Knapsack problem. We present a PTAS for Generalized Multistage d-Knapsack.

2012 ACM Subject Classification Theory of computation → Packing and covering problems; Theory of computation → Problems, reductions and completeness

Keywords and phrases Multistage, Multiple-Knapsacks, Multidimensional Knapsack

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2021.15

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2105.00882

1 Introduction

In many optimization settings, the problem of interest is defined over a time horizon in which the actual setting evolves, resulting in changes over time to the problem constraints and the objective function. Thus, even if the optimization problem at hand can be solved efficiently for a single time unit, it may not be clear how to extend this solution to a time-evolving setting.

An example of such a setting comes from the world of cloud management. A cloud provider maintains a data center with servers and offers clients virtual machines having different processing capabilities. Each client demands a virtual machine (with certain properties), and if provided it must pay for it. It would be naïve to assume that the demand of clients is static over time. Factors, such as peak vs. off-hours, and the day of the week, might affect client demand. Also, the cloud provider might either turn off servers to reduce hardware deterioration and electricity usage, or open more servers to meet higher demand. Thus, the optimization problem is partitioned into multiple *stages*, where in each stage there are different constraints and possibly a different optimization goal.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021).
Editors: Mary Wootters and Laura Sanità; Article No. 15; pp. 15:1–15:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A simple solution is to ignore the dynamicity of the problem, and solve each stage separately and independently of other stages. Thus, profit at each stage is maximized, ignoring the solutions computed for the other stages. Such a solution may result in disgruntled clients, as it can lead to intermittent service between stages. Instead, we will aim for a *multistage* solution that balances between the optimum of each stage, while preserving some continuity between consecutive stages. This will be achieved by incorporating the continuity of the solution into the overall profit.

The multistage model was first introduced by Gupta et al. [18] and Eisenstat et al. [9] to address dynamic environments. Since its introduction, it has received growing attention (examples include [1, 12, 2, 4, 15, 8]). In the multistage model we are given a sequence of instances of an optimization problem. A solution constitutes of a series of solutions, one for each instance.

Two different ideas were used to enforce a balance between single stage optimality and continuity. In [9, 18] a change cost is charged for the dissimilarity of consecutive solutions, while in [4] additional gains were given for their similarity. In the aforementioned cloud management problem, the change cost can be interpreted as installation costs and eviction costs charged when a client is initially served, and then its service is discontinued. The gains can be modeled as increased costs the client is charged to guarantee the continuity of its service.

The cloud management problem described above can be viewed as a multistage problem where the underlying optimization problem is the Multiple Knapsack problem (MKP). In MKP we are given a set of items, each associated with a weight and a profit. Also, we are given a set of bins, each one having a capacity. A feasible solution for MKP is an assignment of items to bins such that the total weight of the items assigned to each bin does not exceed its capacity. The objective is to find a feasible solution maximizing the profit accrued from the assigned items. In the context of the cloud management setting, the items are the virtual machine demands of the clients and the bins are the available servers.

1.1 Problem Definition

We study the Generalized Multistage d-Knapsack problem. We begin with an informal description of the problem. An instance of the problem consists of T stages, where in each stage we are given an instance of a generalization of the classic knapsack problem. While the instances differ between stages, in all stages the same set of items I can be packed. The continuity of the solution is enforced by quantifying the similarity of consecutive solutions and integrating it into the objective function.

We quantify continuity by four types of values. The first two values specify gains earned for the similarity of solutions. For example, if an item i is packed in stages $t-1$ and t, gain $g^+_{i,t}$ is awarded. Similarly, $g^-_{i,t}$ is awarded if i is not packed in $t-1$ and t. The other two values define the cost of changes between consecutive solutions. For example, if an item i was not packed in stage $t-1$, and it is decided to pack it in stage t, a change cost of $c^+_{i,t}$ is charged. Similarly, $c^-_{i,t}$ is charged if i is packed in t, but not in $t+1$.

The packing problem at each stage generalizes the Multiple Knapsack problem, as well as the d-Dimensional Knapsack problem. In each instance of the problem we are given d sets of bins, and the weight an item occupies in a bin depends on the set to which the bin belongs to. The profit of an item is accrued once it is assigned to some bin in all d sets of bins. This problem is called d-Multiple Knapsack Constraints Problem and is formally defined below.

A Multiple Knapsack Constraint (MKC) is a tuple $K = (w, B, W)$ defined over a set of items I. The function $w : I \to \mathbb{R}_+$ defines the weight of the items, B is a set of bins, each equipped with a capacity defined by the function $W : B \to \mathbb{R}_+$. An *assignment* is a function $A : B \to 2^I$, defining which items are assigned to each of the bins. An assignment is feasible if $w(A(b)) = \sum_{i\in A(b)} w_i \leq W(b)$ for each bin $b \in B$. Similarly, given a tuple of MKCs $\mathcal{K} = (K_j)_{j=1}^d$ over I, a tuple of d assignments $\mathcal{A} = (A^j)_{j=1}^d$ is feasible for $\mathcal{K}$ if for each $j = 1, \ldots, d$ assignment A^j is a feasible assignment for K_j. We say A is an assignment of set $S \subseteq I$ if $S = \cup_{b\in B} A_b$.

In d-Multiple Knapsack Constraints Problem (d-MKCP), a problem first introduced in [11], we are given a tuple $(I, \mathcal{K}, p)$, where I is a set of items, $\mathcal{K}$ is a tuple of d MKCs and $p : I \to \mathbb{R}_{\geq 0}$ defines the profit of each item. A feasible solution for d-MKCP is a set $S \subseteq I$ and a tuple of feasible assignments $\mathcal{A}$ (w.r.t $\mathcal{K}$) of S. The goal is to find a feasible solution that maximizes $p(S) = \sum_{i\in S} p(i)$. We note that if there exists an item with negative profit it can be discarded in advance. This fact is used later on, in Section 2.1.

The Generalized Multistage d-Knapsack problem (d-GMK), is the multistage model of d-MKCP. The problem is defined over a time horizon of T stages as follows. An instance of the problem is a tuple $\left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$ is a d_t-MKCP instance with $d_t \leq d$ for $t \in [T]$, $g^+, g^- \in \mathbb{R}_+^{I\times[2,T]}$ are the gain vectors and $c^+, c^- \in \mathbb{R}_+^{I\times[1,T]}$ are the change cost vectors.[1] We use $g_{i,t}^+$ and $g_{i,t}^-$ to denote the gain of item i at stage t. Similarly, we use $c_{i,t}^+$ and $c_{i,t}^-$ to denote the change cost of item i at stage t.

A feasible solution for d-GMK is a tuple $(S_t, \mathcal{A}_t)_{t=1}^T$, where $(S_t, \mathcal{A}_t)$ is a feasible solution for $\mathcal{P}_t$ (note that $\mathcal{A}_t$ is a tuple of assignments of S_t). Throughout the paper we assume $S_0 = S_{T+1} = \emptyset$ and denote the objective function of instance $\mathcal{Q}$ by $f_{\mathcal{Q}} : I^T \to \mathbb{R}$, where

$$\begin{aligned} f_{\mathcal{Q}}\left((S_t)_{t=1}^T\right) = & \sum_{t=1}^{T}\sum_{i\in S_t} p_t(i) + \sum_{t=2}^{T}\left(\sum_{i\in S_{t-1}\cap S_t} g_{i,t}^+ + \sum_{i\notin S_{t-1}\cup S_t} g_{i,t}^-\right) \\ & - \sum_{t=1}^{T}\left(\sum_{i\in S_t\setminus S_{t-1}} c_{i,t}^+ + \sum_{i\in S_t\setminus S_{t+1}} c_{i,t}^-\right). \end{aligned}$$

The goal is to find a feasible solution that maximizes the objective function $f_{\mathcal{Q}}$.

A study of d-GMK reveals it does not admit a constant factor approximation algorithm (see Section 3). We found that in hard instances the change costs are much larger than the profits. Thus we consider an important parameter of the problem, the *profit-cost ratio*. It is defined as the maximum ratio, over all items, between the change cost (c^+, c^-) and the profit of an item over all stages. It is denoted by $\phi_{\mathcal{Q}}$ for any instance $\mathcal{Q}$, and is formally defined as

$$\phi_{\mathcal{Q}} = \min\left(\left\{\infty\right\}\bigcup\left\{r \geq 0 \;\middle|\; \forall i \in I, t_1, t_2 \in [T]: \ \max\left\{c_{i,t_1}^+, c_{i,t_1}^-\right\} \leq r \cdot p_{t_2}(i)\right\}\right)$$

We show that d-GMK instances where the profit-cost ratio is bounded by a constant admit a PTAS.

We also consider Subdmodular d-GMK, a submodular variant of d-GMK where the profit functions are replaced with monotone submodular set functions. A set function $p : 2^I \to \mathbb{R}$ is submodular if for every $A \subseteq B \subseteq I$ and $i \in I \setminus B$ it holds that $p(A \cup \{i\}) - p(A) \geq p(B \cup \{i\}) - p(B)$. Submodular functions appear naturally in many settings such as coverage [13], matroid rank [5] and cut functions [14]. We use similar techniques to develop the algorithms for d-GMK and Submodular d-GMK. Thus, we focus on d-GMK and defer the formal definition as well as the algorithm for Submoduar d-GMK to the full version of this paper [10].

[1] We use the notations $[n, m] = \{i \in \mathbb{N} \mid n \leq i \leq m\}$ and $[n] = [1, n]$ for $n, m \in \mathbb{N}$.

Both d-GMK and Submodular d-GMK generalize the Multistage Knapsack problem recently considered by Bampis et al. [4]. There are several aspects by which it is generalized. First, handling multiple knapsack constraints as well as d-dimensional knapsack vs a single knapsack in [4]. Second, the profit earned from assigning items can be described as a submodular function, not only by a modular function. Third, [4] considered only symmetric gains, i.e., the same gain is earned whether an item is assigned or not assigned in consecutive stages. Lastly, change costs were not considered in [4].

1.2 Our Results

Our main result is stated in the following theorem.

▶ **Theorem 1.** *For any fixed $d \in \mathbb{N}$ and $\phi \geq 1$ there exists a randomized PTAS for d-GMK with a profit-cost ratio bounded by ϕ.*

The result uses the general framework of [4], in which the authors first presented an algorithm for instances with bounded time horizon, and then showed how it can be scaled for general instances. To handle bounded time horizons we show an approximation factor preserving reduction (as defined in [22][2]) from d-GMK to a generalization of q-MKCP. The reduction illuminates the relationship between d-GMK and q-MKCP. As q-MKCP admits a PTAS [11], this results in a PTAS for d-GMK instances with a bounded time horizon.

We note the reduction can be applied to the problem considered in [4] as well. In this case the target optimization problem is d-dimensional knapsack with a matroid constraint. As the latter problem is known to admit a PTAS [17], this suggest a simpler solution for bounded time horizon in comparison to the one given in [4].

To generalize the result to unbounded time horizon we use an approach similar to [4], though a more sophisticated analysis was required to handle the change costs. The generalization is achieved by cutting the time horizon into sub-instances with a fixed time horizon. Each sub-instance is solved separately, and then the solutions are combined to create a solution for the full instance. Handling change costs is trickier as cutting an instance may lead to an excessive charge of change costs at the cut points. We must compensate for these additional costs, or we will not be able to bound the value of the solution.

The results for the modular variant generalizes the PTAS for Multistage Knapsack [4]. For $d \geq 2$, we cannot expect better results as even d-KP, also generalized by d-GMK, does not admit an *efficient* PTAS (EPTAS). Theorem 2 shows an EPTAS cannot be obtained for 1-GMK as well

▶ **Theorem 2.** *Unless $W[1] = FPT$, there is no EPTAS for 1-GMK, even if the length of the time horizon is $T = 2$, the set of bins in each MKC contains one bin and there are no change costs.*

The theorem is proved using a simple reduction from 2-dimensional knapsack. Bampis et al. [4] considered a similar withered down instance and proved that even if the gains are symmetric (i.e., $g^+_{i,t} = g^-_{i,t}$) a *Fully* PTAS (FPTAS) does not exist for the problem.

Using a reduction from multidimensional knapsack we show that 1-GMK, in its general form, cannot be approximated to any constant factor.

▶ **Theorem 3.** *For any $d \geq 1$, there is no polynomial time approximation algorithm for d-GMK with a constant approximation ratio, unless $NP = ZPP$.*

[2] A formal definition is provided in Appendix A for completeness

This result justifies our study of the special cases of d-GMK in which the profit-cost ratio is bounded by a constant.

The techniques used to develop the algorithm for d-GMK can be adjusted slightly to produce an approximation algorithm for Submodular d-GMK. The complete details are given in the full version of this paper [10].

▶ **Theorem 4.** *For any fixed $d \in \mathbb{N}$ and $\epsilon > 0$ there exists a randomized $\left(1 - \frac{1}{e} - \epsilon\right)$-approximation algorithm for Submodular d-GMK.*

In the submodular variant one cannot hope for vast improvement over our results as the algorithm is almost tight. This is due to the hardness results for submodular maximization subject to a cardinality constraint presented by Nemhauser and Wolsey [20].

1.3 Related Work

In the multistage model we are given a series of instances of an optimization problem, and we search for a solution which optimizes each instance while maintains some similarity between solutions. Many optimization problem were considered under this framework. These include matching [2, 7], clustering [8], subset sum [3], vertex cover [15] and minimum $s-t$ path [16].

The multistage model was first presented by both Eisenstat et al. [9] and Gupta et al. [18]. In [9] the Multistage Facility Location problem was considered, where the underlying metric in which clients and facility reside changes over time. A logarithmic approximation algorithm was presented for two variants of the problem; the hourly opening costs where the opening cost of a facility is charged at each stage in which it is open, and the fixed opening costs where a facility is open at all stages after its opening costs is paid. A logarithmic hardness result was also presented for the fixed opening costs variant. An et al. [1] improved the result for the hourly opening costs variant and presented a constant factor approximation. Fairstein et al. [12] proved that the logarithmic hardness result does not hold if only the client locations change over time and the facilities are static.

As mentioned, the multistage model was also introduced by Gupta et al. [18], where the Multistage Matroid Maintenance (MMM) problem was considered. In MMM we are given a set of elements equipped with costs that change over time. In addition, we are given a matroid. The goal is to select a base of minimum costs at each stage whilst minimizing the cost of the difference of bases selected for consecutive stages. Gupta et al. [18] presented a logarithmic approximation algorithm for the problem, as well as fitting lower bound, proving this result is tight.

Organization

Section 2 provides the approximation schemes from Theorem 1. Hardness results are presented in Section 3.

2 Approximation Scheme for d-GMK

In this section we derive the approximation scheme for d-GMK with bounded profit-cost ratio. In Section 2.1 we show how a PTAS for instances with bounded time horizon can be obtain via a reduction to a variant of d-MKCP. Subsequently, in Section 2.2 we show how the algorithm for bounded time horizon can be used to approximate general instance.

2.1 Bounded Time Horizon

In this section we provide a reduction from an instance of q-GMK to a generalization of d-MKCP (for specific values of d and q). The generalization was presented in [11] and is called d-MKCP With A Matroid Constraint (d-MKCP+) and is defined by a tuple $(I, \mathcal{K}, p, \mathcal{I})$, where $(I, \mathcal{K}, p)$ forms an instance of d-MKCP. Also, the set $\mathcal{I} \subseteq 2^I$ defines a matroid[3] constraint. A feasible solution for d-MKCP+ is a set $S \in \mathcal{I}$ and a tuple of feasible assignments $\mathcal{A}$ (w.r.t $\mathcal{K}$) of S. The goal is to find a feasible solution which maximizes $\sum_{i \in S} p(i)$. The following definition presents the construction of the reduction.

▶ **Definition 5.** *Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$ and $\mathcal{K}_t = (K_{t,j})_{j=1}^{d_t}$. Define $R(\mathcal{Q}) = \left(E, \tilde{\mathcal{K}}, \tilde{p}, \mathcal{I}\right)$ where*

- $E = I \times 2^{[T]}$
- $\mathcal{I} = \left\{S \subseteq E \;\middle|\; \forall i \in I : \;\left|S \cap \left(\{i\} \times 2^{[T]}\right)\right| \leq 1\right\}$
- *For $t \in [T], j \leq d_t$ set MKC $\tilde{K}_{t,j} = (\tilde{w}_{t,j}, B_{t,j}, W_{t,j})$ over E, where $K_{t,j} = (w_{t,j}, B_{t,j}, W_{t,j})$ and*

$$\tilde{w}_{t,j}((i, D)) = \begin{cases} w_{t,j}(i) & t \in D \\ 0 & \text{otherwise} \end{cases}$$

- *For $t = 1, ..., T, d_t < j \leq d$ set MKC $\tilde{K}_{t,j} = (w_0, \{b\}, W_0)$ over E, where $w_0 : 2^E \rightarrow \{0\}$, $W_0(b) = 0$ and b is an arbitrary bin (object).*
- $\tilde{\mathcal{K}} = \left(\tilde{K}_{t,j}\right)_{t \in [T], j \in [d]}$.
- *The objective function $\tilde{p}$ is defined as follows.*

$$\tilde{p}(S) = \sum_{(i,D) \in S} \left(\sum_{t \in D} p_t(i) + \sum_{t \in D : t-1 \in D} g^+_{i,t} + \sum_{t \notin D : t-1 \notin D} g^-_{i,t} - \sum_{t \in D : t-1 \notin D} c^+_{i,t} - \sum_{t \in D : t+1 \notin D} c^-_{i,t} \right)$$

Each element $(i, D) \in E$ states the subset of stages in which item i is assigned. I.e., i is only assigned in stages $t \in D$. Thus any solution should include at most one element (i, D) for each $i \in I$. This constraint is fully captured by the partition matroid constraint defined by the set of independent sets $\mathcal{I}$. Finally, if an element (i, D) is selected, we must assign i in each MKC $K_{t,j}$ for $j \in [d_t], t \in D$. This is captured by the weight function $\tilde{w}$, as an element (i, D) weighs $w_{t,j}(i)$ if and only if $t \in D$ (otherwise its weight is zero and it can be assigned for “free”).

▶ **Lemma 6.** *For any d-GMK instance $\mathcal{Q}$ with time horizon T, it holds that $R(\mathcal{Q})$ is a dT-MKCP+ instance.*

Proof. Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$ and $\mathcal{K}_t = (K_{t,j})_{j=1}^{d_t}$. Also, let $R(\mathcal{Q}) = \left(E, \tilde{\mathcal{K}}, \tilde{p}, \mathcal{I}\right)$ be the reduced instance of $\mathcal{Q}$ as defined in Definition 5. It is easy to see that the set $\mathcal{I}$ is the independent sets of a partition matroid, as for each item i at most one element (i, D) can be chosen. Thus, $\mathcal{I}$ is the family of independent sets of a matroid as required.

Next, $\mathcal{K}$ defines a tuple of MKCs, so all that is left to prove is that $\tilde{p}$ is non-negative and modular. For each element $(i, D) \in E$ we can define a fixed value

$$v((i, D)) = \sum_{t \in D} p_t(i) + \sum_{t \in D:\; t-1 \in D} g^+_{i,t} + \sum_{t \notin D:\; t-1 \notin D} g^-_{i,t} - \sum_{t \in D:\; t-1 \notin D} c^+_{i,t} + \sum_{t \in D:\; t+1 \notin D} c^-_{i,t}$$

It immediately follows that $\tilde{p}(S) = \sum_{e \in S} v(e)$ and that $\tilde{p}$ is modular. As stated in Section 1.1, elements with negative values are discarded in advance such that $\tilde{p}$ is also non-negative. ◀

[3] A formal definition for matroid can be found in [21]

▶ **Lemma 7.** *Let $\mathcal{Q}$ be an instance of d-GMK with time horizon T. For any feasible solution $(S_t, \mathcal{A}_t)_{t=1}^T$ of $\mathcal{Q}$ there exists a feasible solution $\big(S, (\tilde{A}_{t,j})_{t\in[T],j\in[d]}\big)$ of $R(\mathcal{Q})$ such that $f_{\mathcal{Q}}\left((S_t)_{t=1}^T\right) = \tilde{p}(S)$.*

Proof. Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$ and $\mathcal{K}_t = (K_{t,j})_{j=1}^{d_t}$. Also, let $R(\mathcal{Q}) = \left(E, \tilde{\mathcal{K}}, \tilde{p}, \mathcal{I}\right)$ be the reduced instance of $\mathcal{Q}$, where $\tilde{\mathcal{K}} = \left(\tilde{K}_{t,j}\right)_{t\in[T],j\in[d]}$ and $\tilde{K}_{t,j} = (\tilde{w}, B_{t,j}, W_{t,j})$ (see Definition 5). Consider some feasible solution $(S_t, \mathcal{A}_t)_{t=1}^T$ for $\mathcal{Q}$, where $\mathcal{A}_t = (A_{t,j})_{j=1}^{d_t}$. In the following we define a solution $\left(S, \left(\tilde{A}_{t,j}\right)_{t\in[T],\ j\in[d]}\right)$ for $R(\mathcal{Q})$. Let

$$S = \{(i, D) \mid i \in I,\ D = \{t \in [T] \mid i \in S_t\}\}$$

It can be easily verified that $S \in \mathcal{I}$. The value of the subset S is

$$\tilde{p}(S) = \\ \sum_{(i,D)\in S} \left(\sum_{t\in D} p_t(i) + \sum_{t\in D:\ t-1\in D} g^+_{i,t} + \sum_{t\notin D:\ t-1\notin D} g^-_{i,t} - \sum_{t\in D:\ t-1\notin D} c^+_{i,t} - \sum_{t\in D:\ t+1\notin D} c^-_{i,t} \right) = \\ \sum_{t=1}^{T} \sum_{i\in S_t} p_t(i) + \sum_{t=2}^{T} \left(\sum_{i\in S_{t-1}\cap S_t} g^+_{i,t} + \sum_{i\notin S_{t-1}\cup S_t} g^-_{i,t} \right) - \sum_{t=1}^{T} \left(\sum_{i\in S_t\setminus S_{t-1}} c^+_{i,t} + \sum_{i\in S_t\setminus S_{t+1}} c^-_{i,t} \right) = \\ f_{\mathcal{Q}}\left((S_t)_{t=1}^T\right).$$

Next, for each $t \in [T], j \in [d]$ we present an assignment $\tilde{A}_{t,j}$ of S. Consider the following two cases:

1. If $j > d_t$, recall $\tilde{K}_{t,j} = (w_0, \{b\}, W_0)$ where $w_0(i, D) = 0$ for all $(i, D) \in E$ and $W_0(b) = 0$. We define $\tilde{A}_{t,j}$ by $\tilde{A}_{t,j}(b) = S$. It thus holds that $w_0(\tilde{A}_{t,j}(b)) = 0 = W_0$. That is, $\tilde{A}_{t,j}$ is feasible.
2. If $j \le d_t$, let $b^* \in B_{t,j}$ be some unique bin in $B_{t,j}$ and define assignment $\tilde{A}_{t,j} : B_{t,j} \to 2^E$ by

$$\begin{aligned} \tilde{A}_{t,j}(b) &= \left(A_{t,j}(b) \times 2^{[T]}\right) \cap S & \forall b \in B_{t,j} \setminus \{b^*\} \\ \tilde{A}_{t,j}(b^*) &= \left(\left(A_{t,j}(b^*) \times 2^{[T]}\right) \cap S\right) \cup \{(i, D) \in S \mid t \notin D\} \end{aligned} \tag{1}$$

The assignment $\tilde{A}_{t,j}$ is a feasible assignment w.r.t $\tilde{K}_{t,j}$ since for each bin $b \in B_{t,j}$ it holds that

$$\sum_{(i,D)\in \tilde{A}_{t,j}(b)} \tilde{w}_{t,j}((i, D)) = \sum_{i\in A_{t,j}(b)} w_{t,j}(i) \le W_{t,j}(b)$$

Let $(i, D) \in S$. If $i \in S_t$ there is $b \in B_{t,j}$ such that $i \in A_{t,j}(b)$, hence $(i, D) \in \tilde{A}_{t,j}(b)$ by (1). If $i \notin S_t$ then $t \notin D$ and thus $(i, D) \in \tilde{A}_{t,j}(b^*)$. Overall, we have $S \subseteq \bigcup_{b\in B_{t,j}} \tilde{A}_{t,j}(b)$. By (1) it follows that $S \supseteq \bigcup_{b\in B_{t,j}} \tilde{A}_{t,j}(b)$ as well, thus $S = \bigcup_{b\in B_{t,j}} \tilde{A}_{t,j}(b)$. I.e, $\tilde{A}_{t,j}$ is an assignment of S.

Note that the assignments can be constructed in polynomial time. We can conclude that $\left(S, \left(\tilde{A}_{t,j}\right)_{t\in[A],j\in[d]}\right)$ is a feasible solution for $R(\mathcal{Q})$, and its value is $f_{\mathcal{Q}}\left((S_t)_{t=1}^T\right)$. ◀

▶ **Lemma 8.** *Let $\mathcal{Q}$ be an instance of d-GMK (with arbitrary time horizon T). For any feasible solution $\left(S, (\tilde{A}_{t,j})_{t\in[T],j\in[d]}\right)$ for $R(\mathcal{Q})$ a feasible solution $(S_t, \mathcal{A}_t)_{t=1}^T$ for $\mathcal{Q}$ such that $f_{\mathcal{Q}}\left((S_t)_{t=1}^T\right) = \tilde{f}(S)$ can be constructed in polynomial time.*

The proof of Lemma 8 is similar to the proof of Lemma 7, thus it is deferred to Appendix A.

For any d-GMK instance with a fixed time horizon T, the reduction $R(\mathcal{Q})$ can be constructed in polynomial (as $|E| = |I| \cdot 2^{|T|}$). The next corollary follows from this observation and lemmas 7 and 8.

▶ **Corollary 9.** *For any fixed $T \in \mathbb{N}$, there exists an approximation factor preserving reduction from d-GMK with a time horizon bounded by T to dT-MKCP+.*

In [11] a PTAS for d-MKCP+ is presented. Thus, the next lemma follows from the above corollary.

▶ **Lemma 10.** *For any fixed $T \in \mathbb{N}$ there exists a randomized PTAS for d-GMK with a time horizon bounded by T.*

2.2 General Time Horizon

In this section we present an algorithm for d-GMK with a general time horizon T. This is done by cutting the time horizon at several stages into sub-instances. Each sub-instance is optimized independently and then the solutions are combined to create a solution for the complete instance. A somewhat similar technique was used in [4]. However, they considered a model without change costs which is much simpler. Our analysis is more delicate as it requires local consideration of the assignment of each item to ensure that any additional costs charged are covered by profit and gains earned.

Given an instance $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ we define a sub-instance for the sub-range $[t_1, t_2]$, denoted throughout this section by $\left((\mathcal{P}_t)_{t=t_1}^{t_2}, g^+, g^-, c^+, c^-\right)$ without shifting or truncating the gain and change costs vectors. For example, for $t \in [t_1, t_2]$ gain $g_{i,t}^+$ is earned for assigning item i in stages t and $t+1$. Also, observe that stages $t_1 - 1$ and $t_2 + 1$ are outside the scope of the instance. Thus, when evaluating a solution for the sub-instance it is assumed that $S_{t_1 - 1} = S_{t_2 + 1} = \emptyset$.

Given an integer $T \in \mathbb{N}$, a set of *cut points* $U = \{u_0, ..., u_k\}$ of T is a set of integers such that for every $j = 0, ..., k-1$ it holds that $u_j < u_{j+1}$ and $1 = u_0 < u_k = T + 1$.

▶ **Definition 11.** *Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$. Also, let $U = \{u_0, \ldots, u_k\}$ be a set of cut points. The tuple of d-GMK instances $\mathcal{Q}_U = \left((\mathcal{P}_t)_{t=u_j}^{u_{j+1}-1}, g^+, g^-, c^+, c^-\right)_{j=0}^{k-1}$ is defined as the cut instances of $\mathcal{Q}$ w.r.t U.*

▶ **Definition 12.** *Let $\mathcal{Q}$ be an instance of d-GMK, $U = \{u_0, ..., u_k\}$ be a set of cut points and $\mathcal{Q}_U$ be the respective cut instances. Also, let $\left((S_t, \mathcal{A}_t)_{t=u_j}^{u_{j+1}-1}\right)_{j=0}^{k-1}$ be a tuple of feasible solutions for the tuple of cut instances $\mathcal{Q}_U$. Then, the solution $(S_t, \mathcal{A}_t)_{t=1}^T$ for $\mathcal{Q}$ is called a cut solution.*

The next corollary elaborates on the relationship between a cut solution and the cut instance solutions from which it is constructed.

▶ **Corollary 13.** *Given any d-GMK instance $\mathcal{Q}$, cut points U, cut instances $\mathcal{Q}_U$ and feasible solutions for the cut instances, the respective cut solution is a feasible solution for $\mathcal{Q}$, and its value is at least the sum of values of the solutions for the cut instances.*

The proof of the corollary is fairly simple, and is defered to Appendix A. We are now ready to present the algorithm for d-GMK with general time horizon length.

Algorithm 1 General Time Horizon.

```
Input : $0 < \epsilon < \frac{1}{4}, \phi \geq 1$, a $d$-GMK instance $\mathcal{Q}$ with time horizon $T$ such that $\phi_{\mathcal{Q}} \leq \phi$,
        and $\alpha$-approximation algorithm $A$ for $d$-GMK with time horizon $T \leq \frac{2\phi}{\epsilon^2}$.
Set $\mu = \frac{\epsilon^2}{\phi}$.
for $j = 1, \ldots, \frac{1}{\mu}$ do
    Set $U_j = \left\{ \frac{a}{\mu} + j - 1 \;\middle|\; a \in \mathbb{N},\ a \geq 1,\ \frac{a}{\mu} + j - 1 \leq T - \frac{1}{\mu} \right\} \cup \{1, T+1\}$.
    Find a solution for each cut instance in $\mathcal{Q}_{U_j}$ using algorithm $A$ and set $\mathcal{S}_j$ as the
      respective cut solution.
Return the solution $\mathcal{S}_j$ which maximizes the objective function $f_{\mathcal{Q}}$.
```

Before analysing the algorithm we present several definitions and lemmas that are essential for the proof. First, we start by reformulating the solution. Instead of describing the assignment of items by the tuple $(S_t)_{t=1}^T$, we define a new set of elements $E = I \times [T] \times [T]$, where each element $(i, t_1, t_2) \in E$ states that item i is assigned in the interval $[t_1, t_2]$. Given a feasible solution $(S_t, \mathcal{A}_t)_{t=1}^T$ for $\mathcal{Q}$ we denote the representation of $(S_t)_{t=1}^T$ as a subset of E by $E\left((S_t)_{t=1}^T\right)$ and it is equal to

$$E\left((S_t)_{t=1}^T\right) = \{(i, t_1, t_2) \in E \mid \forall t \in [t_1, t_2] : i \in S_t \quad \text{and} \quad i \notin S_{t_1 - 1} \cup S_{t_2 + 1}\}\,.$$

If $\tilde{S} = E\left((S_t)_{t=1}^T\right)$, we define the reverse mapping as $\tilde{S}(t) = \{i \in I \mid \exists (i, t_1, t_2) \in S : t \in [t_1, t_2]\} = S_t$. Now, we can define a solution for d-GMK using our new representation as $(\tilde{S}, \mathcal{A}_t)_{t=1}^T$.

▶ **Definition 14.** *The* value, *$v(e)$, of element $e = (i, t_1, t_2)$ is defined as the total value earned from assigning i in the range $[t_1, t_2]$ minus the change costs charge for assigning and discarding it. Formally,*

$$v(e) = \sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g_{i,t}^{+} - c_{i,t_1}^{+} - c_{i,t_2}^{-}$$

The value of solution $\tilde{S} \subseteq E$ for d-GMK instance $\mathcal{Q}$ is $\sum_{e \in \tilde{S}} v(e) + \sum_{t=2}^{T} \sum_{i \notin \tilde{S}(t-1) \cup \tilde{S}(t)} g_{i,t}^{-}$ and it is equal to $f_{\mathcal{Q}}\left((\tilde{S}(t))_{t=1}^T\right)$.

In Algorithm 1 we consider a solution for a tuple of cut instances created by cutting an instance at a set of cut points. Here we consider the opposite action, the effect of cutting a solution at these cut points. We start by considering a single cut point.

▶ **Definition 15.** *Given an element $e = (i, t_1, t_2)$ and a cut point $u \in (t_1, t_2]$ we define the outcome of cutting e at u as the set of intervals $u(e) = \{(i, t_1, u-1), (i, u, t_2)\}$. Also, the* loss *caused by cutting e at u is defined as the difference between the value of e and the sum of value of elements in $u(e)$. It is denoted by $\ell(e, u)$ and is equal to*

$$\ell(e, u) = v(e) - \sum_{e' \in u(e)} v(e') = g_{i,u}^{+} + c_{i,u}^{+} + c_{i,u-1}^{-} \tag{2}$$

We can similarly extend the definition to include more than one cut point as follows.

▶ **Definition 16.** *Given a set of cut points $U = \{u_0, u_1, \ldots, u_k\}$ and an element $e = (i, t_1, t_2)$ define $U(e)$ as the set of elements created by cutting the e at all cut points in U. Formally,*

$$U\left((i, t_1, t_2)\right) = \left\{ (i, \max\{t_1, u_{r-1}\}, \min\{u_r - 1, t_2\}) \;\middle|\; r \in [k] \text{ and } [u_{r-1}, u_r - 1] \cap [t_1, t_2] \neq \emptyset \right\}$$

Consider the following example as a demonstration of the above definition. If $e = (i, t_1, t_2)$ and $(t_1, t_2] \cap U_j = \{u_2, u_3\}$, then $U_j(e) = \{(i, t_1, u_2 - 1), (i, u_2, u_3 - 1), (i, u_3, t_2)\}$.

The definition of loss caused by cutting an element can be extended to a set of cut points U. If element $e = (i, t_1, t_2)$ is cut by a of cut points U the loss is

$$\ell(e, U) = v(e) - \sum_{e' \in U(e)} v(e') = \sum_{u \in U \cap (t_1, t_2]} \left(g^{+}_{i,u} + c^{+}_{i,u} + c^{-}_{i,u-1}\right) = \sum_{u \in U \cap (t_1, t_2]} \ell(e, u) \qquad (3)$$

since only gains g^+ are lost due to cutting as well as change cost for splitting an assignment into two intervals. This means that even if an element is cut multiple times, the loss due to each cut point can be considered separately.

▶ **Lemma 17.** *Let $0 < \epsilon < \frac{1}{4}$, $\phi \geq 1$ and A be an α-approximation algorithm for d-GMK with time horizon $T \leq \frac{2\phi}{\epsilon^2}$. Also, let $\mathcal{Q}$ be an instance of d-GMK such that $\phi_{\mathcal{Q}} \leq \phi$. Algorithm 1 approximates $\mathcal{Q}$ within a factor of $(1 - \epsilon)\alpha$.*

Proof. Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^{T}, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK with time horizon T and profit-cost ratio $\phi_{\mathcal{Q}} \leq \phi$. We assume for simplicity ϕ is integral. Let $0 < \epsilon < \frac{1}{4}$ and $\mu = \frac{\epsilon^2}{\phi}$. Also, let A be an α-approximation algorithm for d-GMK with time horizon $T' \leq \frac{2\phi}{\epsilon^2} = \frac{2}{\mu}$. Note, if $T \leq \frac{2}{\mu}$, the cut points set U_0 is an empty set, and in this case A returns an α-approximation solution for $\mathcal{Q}$ as required.

Let $U_j = \{u^j_1, \ldots, u^j_{k_j}\}$ for $j = 1, \ldots \frac{1}{\mu}$. We show that there exists a set of cut points U_j and a tuple of solutions $\left((S_t, \mathcal{A}_t)_{t=u^j_r}^{u^j_{r+1}-1}\right)_{r=0}^{k_j-1}$ for each cut instance in $\mathcal{Q}_{U_j} = (q^r_j)_{r=0}^{k_j-1}$, such that the sum of values of the solutions, $\sum_{r=0}^{k_j-1} f_{q^r_j}\left((S_t)_{t=u_r}^{u_{r+1}-1}\right)$, is sufficiently large. From Corollary 13 it follows that the value of a cut solution is larger than the sum of its parts (due to lost gains and change costs saved if an item is assigned in adjacent instances). Thus, this also proves that the maximum cut solution found is sufficiently large as well.

Let $(S^*_t, \mathcal{A}^*_t)_{t=1}^{T}$ be an optimal solution for $\mathcal{Q}$, and let $\tilde{S}^* = E\left((S^*_t)_{t=1}^{T}\right)$. We partition $\tilde{S}^*$ into two subsets by the length of the interval they describe. Formally, $X = \left\{(i, t_1, t_2) \in \tilde{S}^* \mid t_2 - t_1 < \frac{\phi}{\epsilon}\right\}$ and $Y = \tilde{S}^* \setminus X$. So X contains short intervals, and Y contains long intervals.

Define $\tilde{S}_j$ as the subset of elements longer than ϕ in $\cup_{e \in Y} U_j(e)$ as well as short elements $e \in X$ that are not cut by U_j. I.e.,

$$\tilde{S}_j = \{e \in X \mid U_j(e) = \{e\}\} \cup \bigcup_{e \in Y} \{(i, t_1, t_2) \in U_j(e) \mid t_2 - t_1 \geq \phi\}$$

At each stage $t \in [T]$ it holds that $\tilde{S}_j(t) \subseteq S^*_t$. Thus there exists a tuple of assignments, denoted by $\mathcal{A}^j_t$, such that $(\tilde{S}_j, \mathcal{A}^j_t)_{t=1}^{T}$ is a feasible solution for $\mathcal{Q}$. We partition set $\tilde{S}_j$ as follows. Let $\tilde{S}_{j,r} = \left\{(i, t_1, t_2) \in \tilde{S}_j \mid [t_1, t_2] \subseteq [u^j_r, u^j_{r+1} - 1]\right\}$. It holds that $\tilde{S}_j = \cup_{r=0}^{k_j-1} \tilde{S}_{j,r}$ as each element (i, t_1, t_2) is contained in exactly one interval $[u^j_r, u^j_{r+1} - 1]$. Thus $\left((\tilde{S}_{j,r}, \mathcal{A}^j_t)_{t=u^j_r}^{u^j_{r+1}-1}\right)_{r=0}^{k_j-1}$ is a tuple of feasible solutions for the cut instances in $\mathcal{Q}_{U_j}$ such that $(\tilde{S}_{j,r}, \mathcal{A}^j_t)_{t=u^j_r}^{u^j_{r+1}-1}$ is a solution for the r-th instance. The value of all elements in the defined solutions for the cut instances is

$$\sum_{r=0}^{k_j-1} \sum_{e \in \tilde{S}_{j,r}} v(e) = \sum_{e \in \tilde{S}_j} v(e) = \sum_{e \in X : U_j(e) = \{e\}} v(e) + \sum_{e \in Y} \sum_{e' = (i, t_1, t_2) \in U_j(e) : t_2 - t_1 \geq \phi} v(e')$$

Thus, after including the value of gains g^- earned by solutions $\tilde{S}_{j,r}$, we can bound the total value of optimal solutions for all cut instances $\mathcal{Q}_{U_j}$ by

$$\sum_{e\in X:U_j(e)=\{e\}} v(e)+\sum_{e\in Y}\sum_{\substack{e'=(i,t_1,t_2)\in U_j(e):\\ t_2-t_1\geq\phi}} v(e')+\sum_{\substack{t\in[u^j_r+1,u^j_{r+1}-1]:\\ u^j_r\in U_j}}\ \sum_{i\notin\tilde{S}_{j,r}(t-1)\cup\tilde{S}_{j,r}(t)} g^-_{i,t} \tag{4}$$

We define $\mathcal{B}$ as the total sum of values of optimal solutions for the cut instances $\left(\mathcal{Q}_{U_j}\right)_{j=1}^{\frac{1}{\mu}}$. By utilizing Equation (4) we can bound $\mathcal{B}$ as follows.

$$\begin{aligned}
\mathcal{B} &\geq \sum_{j=1}^{\frac{1}{\mu}}\sum_{r=0}^{k_j-1}\left(\sum_{e\in\tilde{S}_{j,r}} v(e)+\sum_{t\in[u^j_r+1,u^j_{r+1}-1]:u^j_r\in U_j}\ \sum_{i\notin\tilde{S}_{j,r}(t-1)\cup\tilde{S}_{j,r}(t)} g^-_{i,t}\right)\\
&=\sum_{j=1}^{\frac{1}{\mu}}\sum_{e\in\tilde{S}_j} v(e)+\sum_{j=1}^{\frac{1}{\mu}}\sum_{t\in[2,T]\setminus U_j}\ \sum_{i\notin\tilde{S}_j(t-1)\cup\tilde{S}_j(t)} g^-_{i,t}\\
&=\sum_{e\in X}\ \sum_{j\in\left[\frac{1}{\mu}\right]:U_j(e)=\{e\}} v(e)+\sum_{e\in Y}\sum_{j=1}^{\frac{1}{\mu}}\ \sum_{e'=(i,t_1,t_2)\in U_j(e):t_2-t_1\geq\phi} v(e')\\
&\quad+\sum_{j=1}^{\frac{1}{\mu}}\sum_{t\in[2,T]\setminus U_j}\ \sum_{i\notin\tilde{S}_j(t-1)\cup\tilde{S}_j(t)} g^-_{i,t}
\end{aligned} \tag{5}$$

We bound the value of each of the three terms separately by comparing it to the value of the optimal solution.

Consider the first term, value earned from short elements, i.e., elements $e=(i,t_1,t_2)\in X$. It holds that $e\in\tilde{S}_j$ if and only if $U_j(e)=\{e\}$ which means that $(t_1,t_2]\cap U_j=\emptyset$. Since for every $j_1\neq j_2$ it holds that $U_{j_1}\cap U_{j_2}=\{1,T+1\}$ and since there are $\frac{1}{\mu}$ sets of cut point, for each element $e\in X$ it holds that $e\in\tilde{S}_j$ for at least $\frac{1}{\mu}-\frac{\phi}{\epsilon}$ values of $j\in[\frac{1}{\mu}]$. Thus,

$$\sum_{e\in X}\ \sum_{j\in\left[\frac{1}{\mu}\right]:U_j(e)=\{e\}} v(e)\geq\left(\frac{1}{\mu}-\frac{\phi}{\epsilon}\right)\sum_{e\in X}v(e)=\frac{1}{\mu}(1-\epsilon)\sum_{e\in X}v(e) \tag{6}$$

Next, we bound the second term, the value earned from long elements, $e\in Y$. Consider the set of cut points U_j. Two operators are applied to each long element. First, it is cut and the subset $U_j(e)$ is defined. Second, short elements are discarded from $U_j(e)$. The resulting subset is $\{(i,t_1,t_2)\in U_j(e)\mid t_2-t_1\geq\phi\}$ and therefore we would like to bound the difference

$$\sum_{e\in Y}\left(v(e)-\sum_{e'\in\{(i,t_1,t_2)\in U_j(e)\ \mid\ t_2-t_1\geq\phi\}} v(e')\right)$$

Consider an element $e=(i,t_1,t_2)\in Y$ cut by cut points set U_j. As shown in Equation (3), the loss caused by cutting e at cut point $u\in(t_1,t_2]$ is independent of other cuts that are applied to e and is equal to $\ell(e,u)$. Thus we can consider each cut point separately.

As mentioned above, if $e=(i,t_1,t_2)\in Y$, the second operator discards elements $e'\in U_j(e)$ that are short. Since the distance between each pair of cut points in U_j is at least $\frac{1}{\mu}=\frac{\phi}{\epsilon^2}>\phi$, each such short element e' is either (i,t_1,u) or (i,u,t_2) for some unique cut point $u\in U_j$. In addition, it must hold that either $u-t_1<\phi$ or $t_2-u<\phi$. We associate the value lost by discarding e' to this unique cut point u.

Let $e = (i, t_1, t_2) \in Y$ and $u \in U_j$ be a cut point such that $u \in (t_1, t_2]$, i.e., u cuts e. There are three cases to consider.

1. If $u - t_1 < \phi$, element $e' = (i, t_1, u-1) \in U_j(e)$ is discarded and a loss of $v(e')$ is associated with u in addition to $\ell(e, u)$. Thus the total loss is at most

$$v(e') + \ell(e,u) = \sum_{t=t_1}^{u-1} p_t(i) + \sum_{t=t_1+1}^{u-1} g^+_{i,t} - c^+_{i,t_1} - c^-_{i,u-1} + g^+_{i,u} + c^+_{i,u} + c^-_{i,u-1} \leq$$
$$\leq \sum_{t=t_1}^{u-1} p_t(i) + \sum_{t=t_1+1}^{u} g^+_{i,t} + c^+_{i,u-1} \leq \sum_{t=t_1}^{u+\phi-1} p_t(i) + \sum_{t=t_1+1}^{u+\phi-1} g^+_{i,t}$$

where the equality is due to Equation (2) and the last inequality is due to the profit-cost ratio.

2. If $t_2 - u < \phi$, element $e' = (i, u, t_2) \in U_j(e)$ is discarded and a loss of $v(e')$ is associated with u in addition to $\ell(e, u)$. Thus the total loss is at most

$$v(e') + \ell(e,u) = \sum_{t=u}^{t_2} p_t(i) + \sum_{t=u+1}^{t_2} g^+_{i,t} - c^+_{i,u} - c^-_{i,t_2} + g^+_{i,u} + c^+_{i,u} + c^-_{i,u-1} \leq$$
$$\leq \sum_{t=u}^{t_2} p_t(i) + \sum_{t=u}^{t_2} g^+_{i,t} + c^-_{i,u-1} \leq \sum_{t=u-\phi}^{t_2} p_t(i) + \sum_{t=u-\phi+1}^{t_2} g^+_{i,t}$$

where the equality is due to Equation (2) and the last inequality is due to the profit-cost ratio.

3. If $t_2 - u \geq \phi$ and $u - t_1 \geq \phi$, no elements are discarded from $U_j(e)$. Thus the only loss is $\ell(e, u)$ and can be bounded by

$$\ell(e,u) = g^+_{i,u} + c^+_{i,u} + c^-_{i,u-1} \leq \sum_{t=u-\phi}^{u+\phi-1} p_t(i) + \sum_{t=u-\phi+1}^{u+\phi-1} g^+_{i,t}$$

Overall we can bound the loss induced by cutting long elements at cut points U_j (due to loss $\ell(e, u)$ and discarded short elements) by

$$\sum_{(i,t_1,t_2)\in Y} \sum_{u\in(t_1,t_2]\cap U_j} \left(\sum_{t=\max\{t_1,u-\phi\}}^{\min\{t_2,u+\phi-1\}} p_t(i) + \sum_{t=\max\{t_1+1,u-\phi+1\}}^{\min\{t_2,u+\phi-1\}} g^+_{i,t} \right)$$

This means that the total value gained from elements that were originally in Y is

$$\sum_{e\in Y} \sum_{j=1}^{\frac{1}{\mu}} \sum_{e'=(i,t_1,t_2)\in u(e,U_j):t_2-t_1\geq\phi} v(e') \geq$$
$$\frac{1}{\mu}\sum_{e\in Y} v(e) - \sum_{j=1}^{\frac{1}{\mu}} \sum_{(i,t_1,t_2)\in Y} \sum_{u\in[t_1+1,t_2]\cap U_j} \left(\sum_{t=\max\{t_1,u-\phi\}}^{\min\{t_2,u+\phi-1\}} p_t(i) + \sum_{t=\max\{t_1+1,u-\phi+1\}}^{\min\{t_2,u+\phi-1\}} g^+_{i,t} \right) \geq$$
$$\frac{1}{\mu}\sum_{e\in Y} v(e) - \sum_{(i,t_1,t_2)\in Y} \sum_{u\in[t_1+1,t_2]} \left(\sum_{t=\max\{t_1,u-\phi\}}^{\min\{t_2,u+\phi-1\}} p_t(i) + \sum_{t=\max\{t_1+1,u-\phi+1\}}^{\min\{t_2,u+\phi-1\}} g^+_{i,t} \right) \geq$$
$$\frac{1}{\mu}\sum_{e\in Y} v(e) - 2\phi \sum_{(i,t_1,t_2)\in Y} \left(\sum_{t\in[t_1,t_2]} p_t(i) + \sum_{t\in[t_1+1,t_2]} g^+_{i,t} \right)$$

where the second inequality follows from the fact that for every $j_1 \neq j_2$ it holds that $U_{j_1} \cap U_{j_2} = \{1, T+1\}$. The last inequality is due to the fact that the profit and gains of element (i, t_1, t_2) is lost at stage $t \in [t_1, t_2]$ if it is cut by a cut point u such that $|u - t| \leq \phi$. Thus, its value is lost in at most 2ϕ instances. Due to the profit-cost ratio, for each long element $e = (i, t_1, t_2) \in Y$ it holds that

$$c^+_{i,t_1} + c^-_{i,t_2} \leq 2\phi \cdot \frac{\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t}}{t_2 - t_1} \leq \epsilon \cdot \sum_{t=t_1}^{t_2} p_t(i) + \epsilon \cdot \sum_{t=t_1+1}^{t_2} g^+_{i,t}$$

By substituting $4\phi(c^+_{i,t_1} + c^-_{i,t_2}) \leq 4\phi\epsilon \left(\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t}\right)$ we get that

$$\begin{aligned}
&\frac{1}{\mu}\sum_{e\in Y} v(e) - 2\phi \sum_{(i,t_1,t_2)\in Y} \left(\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t}\right) = \\
&\sum_{(i,t_1,t_2)\in Y} \left(\frac{1}{\mu}\left(\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t} - c^+_{i,t_1} - c^-_{i,t_2}\right) - 2\phi\left(\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t}\right)\right) \geq \\
&\sum_{(i,t_1,t_2)\in Y} \left(\left(\frac{1}{\mu} - 2\phi - 4\phi\epsilon\right)\left(\sum_{t=t_1}^{t_2} p_t(i) + \sum_{t=t_1+1}^{t_2} g^+_{i,t}\right) - \left(\frac{1}{\mu} - 4\phi\right)\left(c^+_{i,t_1} + c^-_{i,t_2}\right)\right) \geq \\
&\left(\frac{1}{\mu} - 2\phi - 4\phi\epsilon\right)\sum_{e\in Y} v(e) \geq \left(\frac{1}{\mu} - \frac{\phi}{\epsilon}\right)\sum_{e\in Y} v(e)
\end{aligned}$$

where the last inequality follows the fact that $\epsilon < \frac{1}{4}$. Overall, we get that

$$\sum_{e\in Y}\sum_{j=1}^{\frac{1}{\mu}} \sum_{e'=(i,t_1,t_2)\in U_j(e): t_2 - t_1 \geq \phi} v(e') \geq \left(\frac{1}{\mu} - \frac{\phi}{\epsilon}\right)\sum_{e\in Y} v(e) \tag{7}$$

Lastly, we bound the third term, gains g^- earned in all cut instance solutions. Consider a cut point set U_j and gain $g^-_{i,t}$ earned in solution $\tilde{S}^*$, i.e., $i \notin S^*_{t-1} \cup S^*_t$. Therefore, any element (i, t_1, t_2) such that $t \in [t_1, t_2]$ or $t - 1 \in [t_1, t_2]$ is not in $\tilde{S}_j$. Thus, unless $t \in U_j$, gain $g^-_{i,t}$ is earned in $\tilde{S}_j$ and in some solution $\tilde{S}_{j,r} \subseteq \tilde{S}_j$ such that $t \in [u^j_r, u^j_{r+1} - 1]$ and $u^j_r \in U_j$. Again, we can use the fact that for every $j_1 \neq j_2$ it holds that $U_{j_1} \cap U_{j_2} = \{1, T+1\}$ and get that

$$\sum_{j=1}^{\frac{1}{\mu}} \sum_{t\in[2,T]\setminus U_j} \sum_{i\notin \tilde{S}_j(t-1)\cap \tilde{S}_j(t)} g^-_{i,t} \geq \left(\frac{1}{\mu} - 1\right)\sum_{t=1}^{T} \sum_{i\notin \tilde{S}^*(t-1)\cap \tilde{S}^*(t)} g^-_{i,t} \tag{8}$$

By substituting inequalities (6),(7) and (8) in Inequality (5) we get that

$$\mathcal{B} \geq \left(\frac{1}{\mu} - \frac{\phi}{\epsilon}\right)\left(\sum_{e\in\tilde{S}^*} v(e) + \sum_{t\in[2,T]} \sum_{i\notin\tilde{S}^*(t-1)\cap\tilde{S}^*(t)} g^-_{i,t}\right) = \left(\frac{1}{\mu} - \frac{\phi}{\epsilon}\right) f_{\mathcal{Q}}((S^*_t)_{t=1}^T)$$

For each set of values, their average is smaller or equal to their maximum value. Thus there must exist at least one set of cut points U_{j^*} such that the sum of values of the solutions $\left(\tilde{S}_{j^*,r}\right)_{r=0}^{k_j-1}$ for its cut instances, $\mathcal{Q}_{U^*_j}$, is at least $\mu \cdot \mathcal{B}$, the average value of a set of solutions for a set of cut instance $\mathcal{Q}_{U_j}$ (for $j = 1, \ldots, \frac{1}{\mu}$). We get that

$$\sum_{e\in\tilde{S}_{j^*}} v(e) + \sum_{t\in[2,T]} \sum_{i\notin\tilde{S}_{j^*}(t-1)\cap\tilde{S}_{j^*}(t)} g^-_{i,t} \geq \mu\mathcal{B} = (1-\epsilon)\, f_{\mathcal{Q}}\left((S^*_t)_{t=1}^T\right)$$

At iteration j^*, in which Algorithm 1 considers the cut instances $\mathcal{Q}_{U_{j^*}}$, algorithm A provides an approximate solution for each cut instance. Thus the value of the solution returned by A for the r-th cut instance is at least

$$\alpha \cdot \left(\sum_{e \in \tilde{S}_{j^*,r}} v(e) + \sum_{t \in [u^j_r+1, u^j_{r+1}-1] : u^j_r \in U_j} \; \sum_{i \notin \tilde{S}_{j^*,r}(t-1) \cup \tilde{S}_{j^*,r}(t)} g^-_{i,t} \right)$$

Summing over all cut instances in $\mathcal{Q}_{U_{j^*}}$ provides a solution with value at least

$$\alpha \cdot \left(\sum_{e \in \tilde{S}_{j^*}} v(e) + \sum_{t \in [2,T]} \; \sum_{i \notin \tilde{S}_{j^*}(t-1) \cap \tilde{S}_{j^*}(t)} g^-_{i,t} \right) \geq (1-\epsilon) \cdot \alpha \cdot f_{\mathcal{Q}}\left((S^*_t)^T_{t=1} \right) \qquad ◀$$

The correctness of Theorem 1 follows immediately from Theorem 17 and Lemma 10.

3 Hardness Results

In this section we present two hardness results for 1-GMK. First, we show no constant approximation ratio exists for 1-GMK (with unbounded profit-cost ratio), even if there is only one bin per stage. Then, we show that even if we wither down the model by removing the change costs, limiting the time horizon length to $T = 2$, and only having a single bin per stage, the problem still does not admit an EPTAS.

The above results are proved by showing an approximation preserving reduction from d-Dimensional Knapsack (d-KP) and Multidimensional Knapsack. For $d \in \mathbb{N}$, in d-KP we are given a set of items I, each equipped with a profit p_i, as well as a d-dimensional weight vector $\bar{w}^i \in [0,1]^d$. We denote j-th coordinate of $\bar{w}^i$ by $\bar{w}^i_j$. In addition, we are given a single bin equipped with a d-dimensional capacity vector $\bar{W} \in \mathbb{R}^d_{\geq 0}$. A subset $S \subseteq I$ is a feasible solution if $\sum_{i \in S} \bar{w}^i \leq \bar{W}$. The objective is to find a feasible solution S which maximizes $\sum_{i \in S} p_i$.

The Multidimensional Knapsack problem is the generalization of d-KP in which d is not fixed. That is, the input for the problem is a d-KP instance for some $d \in \mathbb{N}$. The solutions and their values are the solution and values of the d-KP instance.

Note that d-KP is a special case of d-MKCP, where the set of MKCs is $\mathcal{K} = (K_j)^d_{j=1}$. The j-th MKC is $K_j = (w_j, B_j, W_j)$, where $w_j(i) = \bar{w}^i_j, B_j = \{b\}$ and $W_j(b) = \bar{W}_j$, where $\bar{W}_j$ is the j-th coordinate of the capacity vector $\bar{W}$. Finally, the profit function $p : I \to \mathbb{R}_{\geq 0}$ is defined as $p(i) = p_i$ for any $i \in I$. For simplicity we will use this notation for d-KP and Multidimensional Knapsack throughout this section.

▶ **Lemma 18.** *There is an approximation preserving reduction from the Multidimensional Knapsack problem to* 1*-GMK with a single bin in each stage.*

Proof. Let $\mathcal{Q} = (I, \mathcal{K}, p)$ be an instance of Multidimensional Knapsack, where $\mathcal{K} = (K_j)^d_{j=1}$. We define an instance of 1-GMK as follows. Define $T = d$, and for $j = 1, \ldots, d$ define $\mathcal{P}_j = (I, (K_j), h)$ with $h(i) = \frac{p(i)}{d}$ for all $i \in I$. The gains vectors are defined as zero vectors, $g^+ = g^- = \vec{0}$. Finally, we define the change cost vectors. For all $i \in I$ we set

$$c^+_{i,t} = \begin{cases} p(i) & t \in [2,d] \\ 0 & \text{otherwise.} \end{cases}, \quad c^-_{i,t} = \begin{cases} p(i) & t \in [1,d-1] \\ 0 & \text{otherwise.} \end{cases}$$

The tuple $\tilde{\mathcal{Q}} = \left((\mathcal{P}_t)^d_{t=1}, g^+, g^-, c^+, c^- \right)$ is a 1-GMK instance with time horizon $T = d$.

Let $(S, \mathcal{A})$ be a feasible solution for $\mathcal{Q}$, where $\mathcal{A} = (A_j)_{j=1}^{d}$. We can easily construct a solution for $\tilde{\mathcal{Q}}$ by setting $\mathcal{A}_j = (A_j)$ for $j = 1, \ldots, d$. Then, $(S_t, \mathcal{A}_t)_{t=1}^{d}$, where $S_j = S$ for $j = 1, \ldots, d$, is a solution for $\tilde{\mathcal{Q}}$. Note that all items are either assigned or not assigned in all stages. Thus the value of the solution is

$$f_{\tilde{\mathcal{Q}}}((S_t)_{t=1}^{d}) = \sum_{t=1}^{d} h(S_t) - \sum_{t=1}^{T}\left(\sum_{i \in S_t \setminus S_{t+1}} c^{-}_{i,t} + \sum_{i \in S_t \setminus S_{t-1}} c^{+}_{i,t}\right) = d \cdot h(S) = p(S)$$

The solution is also feasible as for every $j \in [d]$ it holds that A_j is a feasible assignments for MKC K_j.

Next, let $(S_t, \mathcal{A}_t)_{t=1}^{d}$ be a feasible solution for $\tilde{\mathcal{Q}}$, where $\mathcal{A}_j = (\tilde{A}_j)$ for $j = 1, \ldots, d$. For $j = 1, \ldots, d$ let $B_j = \{b_j\}$. We define the selected items set as $S = \bigcap_{j \in [d]} S_j$ and define the assignments accordingly, $A_j(b_j) = S$ for $j = 1, \ldots, d$ and $\mathcal{A} = (A_j)_{j=1}^{d}$. Consider some $j \in [s]$, the assignment A_j is feasible as $A_j(b_j) \subseteq \tilde{A}_j(b_j)$ and

$$\sum_{i \in A_j(b_j)} w_j(i) \leq \sum_{i \in \tilde{A}_j(b_j)} w_j(i) \leq W_j(b_j)$$

The value of the solution is

$$p(S) = d \cdot h(S) \geq \sum_{t=1}^{d} h(S_t) - \sum_{t=1}^{T}\left(\sum_{i \in S_t \setminus S_{t+1}} c^{-}_{i,t} + \sum_{i \in S_t \setminus S_{t-1}} c^{+}_{i,t}\right) = f_{\tilde{\mathcal{Q}}}((S_t)_{t=1}^{d})$$

where the inequality follows from the construction of $\tilde{Q}$. Furthermore, note that S can be constructed in polynomial time, which concludes the proof. ◀

In [6] Chekuri and Kahanna showed that Multidimensional Knapsack does not admit any constant approximation ratio unless $NP = ZPP$. Theorem 3 follows immediately from the hardness result of [6] and Lemma 18.

We now proceed to the second hardness result.

▶ **Lemma 19.** *There is an approximation preserving reduction from 2-Dimensional Knapsack problem to 1-GMK with time horizon $T = 2$, no change costs and a single bin per stage.*

Proof. Let $\mathcal{Q} = (I, \mathcal{K}, p)$ be an instance of 2-dimensional knapsack, where $\mathcal{K} = (K_1, K_2)$. Also, since p is modular, it holds that $p(S) = \sum_{i \in S} p_i$.

We define an instance of 1-GMK as follows. Set $T = 2$, $\mathcal{P}_1 = (I, (K_1), h)$ and $\mathcal{P}_2 = (I, (K_2), h)$, where h is the zero function, i.e., $h : I \to \{0\}$ such that $\forall i \in I$ it holds that $h(i) = 0$. Since there are only two stages, gains exists only for stage for $t = 2$. Set $g^{+}_{i,2} = p(i)$ and $g^{-}_{i,2} = 0$ for each item $i \in I$. Finally, we set the change cost vectors as $c^{+} = c^{-} = \vec{0}$. The tuple $\tilde{\mathcal{Q}} = ((\mathcal{P}_1, \mathcal{P}_2), g^{+}, g^{-}, c^{+}, c^{-})$ is a 1-GMK instance with time horizon $T = 2$. Note that since all profits, change costs and gains g^{-} are zero we can write the objective function as

$$f_{\tilde{\mathcal{Q}}}((S_1, S_2)) = \sum_{i \in S_1 \cap S_2} g^{+}_{i,2}.$$

Let $(S, \mathcal{A})$ be a feasible solution for $\mathcal{Q}$, where $\mathcal{A} = (A_1, A_2)$. We can easily construct a solution for the $\tilde{\mathcal{Q}}$ by setting $\mathcal{A}_1 = (A_1)$ and $\mathcal{A}_2 = (A_2)$. Then, $(S_t, \mathcal{A}_t)_{t=1}^{2}$, where $S_1 = S_2 = S$, is a solution for $\tilde{\mathcal{Q}}$. Note that all items are either assigned in both stages or not assigned in both stages. Thus the value of the solution is

$$f_{\tilde{\mathcal{Q}}}((S, S)) = \sum_{i \in S_1 \cap S_2} g^{+}_{i,2} = \sum_{i \in S} g^{+}_{i,2} = \sum_{i \in S} p_i = p(S)$$

The solution is also feasible as A_1 and A_2 are feasible assignments of K_1 and K_2 (respectively).

Next, let $(S_t, \mathcal{A}_t)_{t=1}^2$ be a feasible solution for $\tilde{\mathcal{Q}}$, where $\mathcal{A}_1 = (\tilde{A}_1)$ and $\mathcal{A}_2 = (\tilde{A}_2)$. Let $B_1 = \{b_1\}$ and $B_2 = \{b_2\}$. We define the selected items set as $S = S_1 \cap S_2$ and define the assignments accordingly, $A_1(b_1) = A_2(b_2) = S$ and $\mathcal{A} = (A_1, A_2)$. Assignment A_1 is feasible as $A_1(b_1) \subseteq \tilde{A}_1(b_1)$ and

$$\sum_{i \in A_1(b_1)} w_1(i) \leq \sum_{i \in \tilde{A}_1(b_1)} w_1(i) \leq W_1(b_1)$$

A similar statement shows that assignment A_2 is feasible as well. The value of the solution is

$$p(S) = \sum_{i \in S} p(i) = \sum_{i \in S} g^+_{i,2} = \sum_{i \in S_1 \cap S_2} g^+_{i,2} = f_{\tilde{\mathcal{Q}}}((S, S))$$

which concludes the proof. ◀

In [19] Kulik and Shachnai showed that there is no EPTAS for 2-KP unless $W[1] = FPT$. Theorem 2 follows immediately from the hardness result of [19] and Lemma 19.

References

1 Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. *ACM Transactions on Algorithms (TALG)*, 13(2):1–20, 2017.

2 Evripidis Bampis, Bruno Escoffier, Michael Lampis, and Vangelis Th Paschos. Multistage matchings. In *16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*, volume 101, pages 7–1, 2018.

3 Evripidis Bampis, Bruno Escoffier, Kevin Schewior, and Alexandre Teiller. Online multistage subset maximization problems. In *European Symposium on Algorithms (ESA)*, volume 144. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

4 Evripidis Bampis, Bruno Escoffier, and Alexandre Teiller. Multistage knapsack. In *Mathematical Foundations of Computer Science (MFCS)*, volume 138, pages 22–1. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

5 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

6 Chandra Chekuri and Sanjeev Khanna. On multidimensional packing problems. *SIAM journal on computing*, 33(4):837–851, 2004.

7 Markus Chimani, Niklas Troost, and Tilo Wiedera. Approximating multistage matching problems. *arXiv preprint arXiv:2002.06887*, 2020.

8 Shichuan Deng, Jian Li, and Yuval Rabani. Approximation algorithms for clustering with dynamic points. In *28th Annual European Symposium on Algorithms (ESA 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

9 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *International Colloquium on Automata, Languages, and Programming*, pages 459–470. Springer, 2014.

10 Yaron Fairstein, Ariel Kulik, Danny Raz, et al. General knapsack problems in a dynamic setting. *arXiv preprint arXiv:2105.00882*, 2021.

11 Yaron Fairstein, Ariel Kulik, and Hadas Shachnai. Modular and submodular optimization with multiple knapsack constraints via fractional grouping. In *29th Annual European Symposium on Algorithms (ESA 2021)*, 2021. (To appear).

12 Yaron Fairstein, Seffi Joseph Naor, and Danny Raz. Algorithms for dynamic nfv workload. In *International Workshop on Approximation and Online Algorithms*, pages 238–258. Springer, 2018.

13 Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

14 Uriel Feige and Michel Goemans. Approximating the value of two power proof systems, with applications to max 2sat and max dicut. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 182–189. IEEE, 1995.

15 Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. Multistage vertex cover. In *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

16 Till Fluschnik, Rolf Niedermeier, Carsten Schubert, and Philipp Zschoche. Multistage st path: Confronting similarity with dissimilarity in temporal graphs. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

17 Fabrizio Grandoni and Rico Zenklusen. Approximation schemes for multi-budgeted independence systems. In *European Symposium on Algorithms*, pages 536–548. Springer, 2010.

18 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *International Colloquium on Automata, Languages, and Programming*, pages 563–575. Springer, 2014.

19 Ariel Kulik and Hadas Shachnai. There is no eptas for two-dimensional knapsack. *Information Processing Letters*, 110(16):707–710, 2010.

20 George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

21 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

22 Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

A Omitted Proofs and Definition

▶ **Definition 20.** *Let Π_1, Π_2 be two maximization problems. An* approximation factor preserving reduction *from Π_1 to Π_2 consists of two polynomial time algorithms f, g such that for any two instances I_1 of problem Π_1 and $I_2 = f(I_1)$ of problem Π_2 it holds that*

- *$I_2 \in \Pi_2$ and $OPT_{\Pi_2}(I_2) \geq OPT_{\Pi_1}(I_1)$.*
- *for any solution s_2 for I_2, solution $s_1 = g(I_1, s_2)$ is a solution for I_1 and $obj_{\Pi_1}(I_1, s_1) \geq obj_{\Pi_2}(I_2, s_2)$.*

where $OPT_\Pi(I)$ is the value of an optimal solution for instance I of problem Π, and $obj_\Pi(I, s)$ is the value of solution s for instance I of problem Π.

Proof of Lemma 8. Let $\mathcal{Q} = \left((\mathcal{P}_t)_{t=1}^T, g^+, g^-, c^+, c^-\right)$ be an instance of d-GMK, where $\mathcal{P}_t = (I, \mathcal{K}_t, p_t)$ and $\mathcal{K}_t = (K_{t,j})_{j=1}^{d_t}$. Also, let $R(\mathcal{Q}) = \left(E, \tilde{\mathcal{K}}, \tilde{p}, \mathcal{I}\right)$ be the reduced instance of $\mathcal{Q}$. and $\left(S, (\tilde{A}_{t,j})_{t\in[T],\ j\in[d]}\right)$ be a feasible solution for $R(\mathcal{Q})$. We define the solution $(S_t, \mathcal{A}_t)_{t=1}^T$ for $\mathcal{Q}$ as follows. For every stage t we set $S_t = \{i \in I \mid \exists (i, D) \in S : t \in D\}$. For every $t = 1, \ldots, T$, $j = 1, \ldots, d_t$ and bin $b \in B_{t,j}$ (the set of bins in MKC $K_{t,j}$) let $A_{t,j}(b) = \{i \in I \mid \exists (i, D) \in \tilde{A}_{t,j}(b) : t \in D\}$. Observe that the sets $(S_t)_{t=1}^T$ and assignments $(A_{t,j})_{t\in[T],\ j\in[d_t]}$ can be constructed in polynomial time as at most $|I|$ elements can be chosen due to the matroid constraint. The assignment $A_{t,j}$ is an assignment of S_t since

$$S_t = \{i \in I \mid \exists (i, D) \in S : t \in D\} = \bigcup_{b\in B_{t,j}} \{i \in I \mid (i, D) \in \tilde{A}_{t,j}(b) : t \in D\} = \bigcup_{b\in B_{t,j}} A_{t,j}(b),$$

where the second equality follows the feasibility of the solution for $R(\mathcal{Q})$. In addition, $A_{t,j}$ is a feasible assignment for MKC $K_{t,j}$ since for every bin $b \in B_{t,j}$ it holds that

$$\sum_{i \in A_{t,j}(b)} w_{t,j}(i) = \sum_{(i,D) \in \tilde{A}_{t,j}(b) : t \in D} \tilde{w}_{t,j}((i,D)) = \sum_{(i,D) \in \tilde{A}_{t,j}(b)} \tilde{w}_{t,j}((i,D)) \leq W_{t,j}(b)$$

Thus $(S_t, \mathcal{A}_t)_{t=1}^{T}$ is a feasible solution for $\mathcal{Q}$.

Lastly, consider the value of the solution for $\mathcal{Q}$. It holds that

$$\begin{aligned}
&f_{\mathcal{Q}}\left((S_t)_{t=1}^{T}\right) = \\
&\sum_{t=1}^{T}\sum_{i \in S_t} p_t(i) + \sum_{t=2}^{T}\left(\sum_{i \in S_{t-1} \cap S_t} g^+_{i,t} + \sum_{i \notin S_{t-1} \cup S_t} g^-_{i,t}\right) - \sum_{t=1}^{T}\left(\sum_{i \in S_t \setminus S_{t-1}} c^+_{i,t} + \sum_{i \in S_t \setminus S_{t+1}} c^-_{i,t}\right) = \\
&\sum_{(i,D) \in S}\left(\sum_{t \in D} p_t(i) + \sum_{t \in D : t-1 \in D} g^+_{i,t} + \sum_{t \notin D : t-1 \notin D} g^-_{i,t} - \sum_{t \in D : t-1 \notin D} c^+_{i,t} - \sum_{t \in D : t+1 \notin D} c^-_{i,t}\right) = \\
&\tilde{p}(S)
\end{aligned}$$

◀

Proof of Corollary 13. Let $\mathcal{Q}$ be an instance of d-GMK, U be a set of cut points. Also, let $\mathcal{Q}_U = (q_j)_{j=0}^{k-1} = \left((\mathcal{P}_t)_{t=u_j}^{u_{j+1}-1}, g^+, g^-, c^+, c^-\right)_{j=0}^{k-1}$ be the corresponding tuple of cut instances, and $\left((S_t, \mathcal{A}_t)_{t=u_j}^{u_{j+1}-1}\right)_{j=0}^{k-1}$ be a tuple of feasible solutions for the cut instances.

We define the solution $(S_t, \mathcal{A}_t)_{t=1}^{T}$ for $\mathcal{Q}$. It is easy to see that the assignments $\mathcal{A}_t$ to $\mathcal{K}_t$ are all feasible assignments of S_t. In addition, it holds that

$$\begin{aligned}
&f_{\mathcal{Q}}\left((S_t)_{t=1}^{T}\right) = \\
&\sum_{t=1}^{T} p_t(S_t) + \sum_{t=2}^{T}\left(\sum_{i \in S_{t-1} \cap S_t} g^+_{i,t} + \sum_{i \notin S_{t-1} \cup S_t} g^-_{i,t}\right) - \sum_{t=1}^{T}\left(\sum_{i \in S_t \setminus S_{t+1}} c^-_{i,t} + \sum_{i \in S_t \setminus S_{t-1}} c^+_{i,t}\right) \geq \\
&\sum_{j=0}^{k-1}\left(\sum_{t=u_j}^{u_{j+1}-1} p_t(S_t) + \sum_{t=u_j+1}^{u_{j+1}-1}\left(\sum_{i \in S_{t-1} \cap S_t} g^+_{i,t} + \sum_{i \notin S_{t-1} \cup S_t} g^-_{i,t}\right)\right. \\
&\qquad \left. - \sum_{t=u_j+1}^{u_{j+1}-1}\left(\sum_{i \in S_t \setminus S_{t+1}} c^-_{i,t} + \sum_{i \in S_t \setminus S_{t-1}} c^+_{i,t}\right) - \sum_{i \in S_{u_{j+1}-1}} c^-_{i,u_{j+1}} - \sum_{i \in S_{u_j}} c^+_{i,u_j}\right) = \\
&\sum_{j=0}^{k-1} f_{q_j}\left((S_t)_{t=u_j}^{u_{j+1}-1}\right)
\end{aligned}$$

where f_{q_j} is the objective functions of cut instance q_j. This proves that a cut solution has a higher value than the sum of solutions for cut instance from which it was created. ◀