



HAL
open science

A simple rounding scheme for multistage optimization

Evripidis Bampis, Dimitris Christou, Bruno Escoffier, Alexander Kononov,
Kim Thang Nguyen

► **To cite this version:**

Evripidis Bampis, Dimitris Christou, Bruno Escoffier, Alexander Kononov, Kim Thang Nguyen. A simple rounding scheme for multistage optimization. *Theoretical Computer Science*, 2022, 907, pp.1–10. 10.1016/j.tcs.2022.01.009 . hal-03542969

HAL Id: hal-03542969

<https://hal.science/hal-03542969>

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

A Simple Rounding Scheme for Multistage Optimization

Evrpidis Bampis¹, Dimitris Christou^{1,5}, Bruno Escoffier^{1,2}, Alexander Kononov^{3,4}, and Nguyen Kim Thang^{1,6}

¹ Sorbonne Université, CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris
{[evripidis.bampis](mailto:evripidis.bampis@lip6.fr), [bruno.escoffier](mailto:bruno.escoffier@lip6.fr)}@lip6.fr

² Institut Universitaire de France

³ Sobolev Institute of Mathematics

⁴ Novosibirsk State University

alvenko@math.ncs.ru

⁵ National Technical University of Athens

dimitrios.christou@hotmail.com

⁶ IBISC, Univ Evry, University Paris-Saclay, France

kimthang.nguyen@univ-evry.fr

Abstract. We consider the *multistage* framework introduced in (Gupta et al., Eisenstat et al., both in ICALP 2014), where we are given a time horizon and a sequence of instances of a (static) combinatorial optimization problem (one for each time step), and the goal is to find a sequence of solutions (one for each time step) reaching a tradeoff between the quality of the solutions in each time step and the stability/similarity of the solutions in consecutive time steps. We first introduce a novel rounding scheme, tailored for multistage problems, that accounts for the moving cost (or stability revenue) of adjacent solutions. Using this rounding scheme, we propose improved approximation algorithms for the multistage variants of Prize-Collecting Steiner Tree and Prize-Collecting Traveling Salesman problems. Furthermore, we introduce a 2-approximation algorithm for multistage Multi-Cut on trees, and we also show that our general scheme can be extended to maximization problems, by providing a 0.75-approximation algorithm for the multistage variant of MaxSat.

Keywords: Multistage optimization · Approximation algorithms · LP-rounding.

1 Introduction

In many applications data are evolving over time (e.g. in data centers [1], video streaming [30], or electricity production [35]). Different models have been introduced in the literature in order to capture the dynamic changes of the data, such as for example, dynamic algorithms [16], temporal graphs [31], etc. Here, we focus on a recently introduced model, the multistage model [15, 26], where we are given a discrete time horizon and a set of instances of a combinatorial optimization problem, one for each time step, and we aim to find a sequence of solutions,

one solution per instance, optimizing the solution quality in each time step and the stability (moving cost or profit) between solutions in consecutive time steps. For a sequence of solutions of a minimization multistage problem, the *service cost* is defined as the sum of the cost of the solutions of each individual instance over the time horizon, while the *moving cost* is defined as the sum of the costs induced by changing the solution in consecutive time steps of the time horizon. The goal is to find a sequence of solutions minimizing the sum of the service cost and the moving cost. Clearly, choosing the best solution in every time step minimizes the service cost, but may induce a huge moving cost. On the other hand, keeping the same solution in every time step will minimize the moving cost, but may induce a huge service cost. Hence, a tradeoff between the service and the moving costs is needed. Surprisingly, some polynomially-time solvable problems in the static case, become very hard in the multistage framework [2, 7, 15, 26]. For instance, while the minimum cost perfect matching problem is polynomially-time solvable in the static case, it becomes hard to approximate even for very restricted cases (when the edges have zero-weights, the time horizon is of only two time steps, or when the graph is bipartite) [7, 26, 12]. Another polynomially-time solvable problem in the static case that becomes hard in the multistage framework is the minimum spanning tree problem [26]. For other problems the situation is much better. This is the case for the multistage variant of the minimum (s, t) -cut problem which remains polynomially-time solvable by a simple reduction to the static (s, t) -cut problem [6]. In this later work, it has been proved that monotone and IP2-non-monotone problems [29] retain their properties in the multistage setting and thus can be solved by efficient approximation algorithms. Furthermore, the authors introduced a new deterministic rounding scheme, called two-threshold rounding scheme, designed for multistage problems in the sense that it is able to take into account both moving costs and service costs when rounding. Using these rounding scheme, they were able to acquire upper bounds on the approximation ratio of the multistage variant of various important optimization problems, including Prize-Collecting Steiner Tree and Prize-Collecting Traveling Salesman problems [3, 5, 11, 25].

1.1 Problem formulation and preliminaries

Linear-programming (LP) based methods are arguably one of the main techniques for the design of approximation algorithms for usual (static) problems, and have already been fruitfully applied in the multistage setting, see e.g. [2, 6, 15].

One of the first problems that have been considered in the multistage framework is the facility location problem. Facility location problem formulations classically use two kinds of variables: variables, say y_j , to express the fact that the facility is open or not, and variables x_{ij} assigning clients to open facilities, with a formulation $\min(c \cdot x + d \cdot y)$ s.t. $(x, y) \in \Delta$, $x_{ij}, y_j \in \{0, 1\}$, where Δ contains feasibility constraints (of the form $x_{ij} \leq y_j$ and $\sum_j x_{ij} = 1$). In the multistage version [15], distance from clients to facility evolve with time, and we have a

sequence of T distance functions. Then, we may modify the solution at each time step $t = 1, \dots, T$ (because of the modification of the distances), but moving a client from one facility (at some time t) to another one (at time $t + 1$) comes at some cost w . Then, globally, the objective function to minimize is the sum of the global *service cost* $\sum_t (c_t x^t + d_t y^t)$, and the global moving cost $w \sum_t \sum_{i,j} |x_{i,j}^t - x_{i,j}^{t+1}|$ (where $x_{i,j}^t = 1$ if client i is affected to facility j at time t). Note that there is a moving cost on variables x but not on variables y .

Let us express this in a more general way, and consider any traditional minimization problem (for simplicity of notation, we focus on *minimization* problems in the first part of this article and then extend our results for *maximization* problems as well). As in the previous example, we consider a general multistage setting where a subset of variables, denoted by x , are subject to moving costs, while the other variables, denoted by y , are not. Then, let us consider an *Integer Linear-Program (ILP)* of this minimization problem

$$\min(c \cdot x + d \cdot y) \quad \text{s.t. } (x, y) \in \Delta, \quad x_i, y_j \in \{0, 1\}.$$

where Δ is some set of linear constraints on the sets of variables x, y and c, d are some non-negative cost vectors.

Then, if T is the time horizon (number of time steps), $n_x = |x|$ and $n_y = |y|$, the multistage variant of the problem can be written as the following *ILP*:

$$\begin{cases} \min & \sum_{t=1}^T \sum_{i=1}^{n_x} c_i^t x_i^t + \sum_{t=1}^T \sum_{i=1}^{n_y} d_i^t y_i^t + \sum_{t=2}^T \sum_{i=1}^{n_x} w_i^t |x_i^t - x_i^{t-1}| \\ \text{s.t.} & (x^t, y^t) \in \Delta \quad \forall t \in [T] \\ & x_i^t, y_j^t \in \{0, 1\} \quad \forall t \in [T], \quad \forall i \in [n_x], \quad \forall j \in [n_y] \end{cases} \quad (1)$$

In (1), we have T sets of variables (x^t, y^t) that must satisfy the set of constraints Δ in each time-step and T non-negative cost vectors c^t, d^t, w^t . The x -variables are associated to a movement cost for changing between time-steps $t - 1$ and t that is given by $\sum_{i \in [n_x]} w_i^t |x_i^t - x_i^{t-1}|$. For the purposes of this article, we will focus on multistage minimization problems where the *movement cost* between adjacent solutions can be expressed as a linear combination of the absolute differences between the coordinates of a set of variables x , as this is the case for a large variety of multistage problems. We will refer to the three quantities that define the objective function as the *service cost* w.r.t. to the x -variables, the *service cost* w.r.t. to the y -variables and the *movement cost*.

We first note that (1) is not (formally) an *ILP*-formulation due to the term $|x_i^t - x_i^{t-1}|$. Yet, we can easily overcome this obstacle by substituting this term using an extra variable z_i^t that is subject to the constraints: (i) $z_i^t \geq x_i^t - x_i^{t-1}$ and (ii) $z_i^t \geq x_i^{t-1} - x_i^t$. Then, it is trivial to show that in any optimal solution of (1) it holds that $z_i^t = |x_i^t - x_i^{t-1}|$. For the rest of this work, we will slightly abuse the notation and formulate our *ILP*'s using these absolute values, since they can always be substituted with z -variables without loss of generality.

Our Contribution. We introduce a rounding scheme that is arguably simpler than the two-threshold rounding scheme introduced in [6]. Our method allows

to improve some approximation results presented in [6] and we show how it can be applied to other multistage minimization or maximization problems. Namely, we improve the approximation ratio for multistage Prize-Collecting Steiner Tree from 3.53 to 2.54 and for multistage Prize-Collecting Traveling Salesman from 3.034 to 2.06. Furthermore, by slightly extending our general scheme we are able to acquire a 2-approximation algorithm for multistage Multi-Cut on trees. Finally, we show that our method can also be applied to maximization problems, by presenting a 0.75-approximate algorithm for a multistage variant of MaxSat.

In Table 1.1, we present for each of the problems considered in this article the best known bounds on their approximation ratio in the traditional (static) case, the previously best known upper bounds on their ratio for their multistage extension and our novel bounds that were obtained via the general rounding scheme discussed above.

Problem	<i>Static Hardness</i>	<i>Static Approx.</i>	<i>Multistage Approx. (Previously Known)</i>	<i>Multistage Approx. (This Article)</i>
<i>f-Set Cover</i>	$f - \epsilon$ [32]	f [28]	$2f$ [6]	f
<i>Prize-Collecting ST</i>		1.991 [3]	3.53 [6]	2.54
<i>Prize-Collecting TSP</i>		1.990 [3]	3.034 [6]	2.06
<i>MultiCut on Trees</i>	$2 - \epsilon$ [24]	2 [24]	–	2
<i>Max Sat</i>	$7/8 + \epsilon$ [27]	0.7968 [4]	–	0.75

Table 1. Known results and our contribution

Related Work.

Multistage setting. The multistage model has been introduced by Eisenstat et al. [15] and Gupta et al. [26]. In [15], two multistage versions of the facility location problem have been studied where clients are moving in some metric space over time. Algorithms with logarithmic approximation ratios have been proposed for both versions, namely the *fixed opening cost* version where a facility once opened remains open and where its opening cost is paid once; and the *hourly opening cost* where the opening cost of a facility is paid at every time step when it is open. An et al. [2] proposed an improved constant factor approximation algorithm for the hourly opening cost variant. Fairstein et al. [18] considered the case where only the location of the clients change over time.

Gupta et al. [26] studied the multistage Maintenance Matroid problem (which includes multistage Spanning tree as a subcase) for both the offline and the online settings. They also introduced the multistage Minimum Cost Perfect Matching problem. They showed that it is hard to approximate even for a constant number of stages. In [7], this negative result has been strengthened since it has been proven that the multistage problem is hard to approximate even for bipartite graphs and for a time horizon with only two time steps. Furthermore, different versions of the problem of maintaining a perfect matching on a multistage graph such that the changes between consecutive matchings are minimized, have been

studied in [12], and new complexity results and efficient approximation algorithms have been proposed.

A multistage variant of the Minimum Linear Arrangement problem, which is related to a variant of the List Update problem [36], has been studied in [34], both in the online and offline frameworks. Fotakis et al. considered in [22] the multistage K -Facility Reallocation problem on the real line when the locations of the agents are stage-dependent. In [23], the authors consider the online multistage Facility Location problem and they present an efficient online algorithm.

Multistage maximization problems have also been studied. For instance, in [8], the multistage Max-Min Fair Allocation problem has been considered in the offline and the online settings. For the multistage variant of the Knapsack problem, positive and negative results have been presented in [10]. In [17], an approximation scheme has been proposed for the Generalized Multistage d -Knapsack problem. In [9], general techniques have been introduced for a large family of online multistage problems, called Subset Maximization problems, leading to a characterization of the variants that admit a constant-competitive online algorithm.

Multistage optimization problems also received an increasing attention from a parameterized complexity perspective, where typically moving cost and service cost are interesting and natural parameters. In that perspective, in [19] it is proved that Multistage 2-SAT is NP-hard even in quite restricted cases, and parameterized algorithms (including kernelization) are presented and proved to be asymptotically optimal. Other problems such as path or vertex cover have also been studied, see e.g. [20, 21].

Static setting. The Prize-Collecting Metric Traveling Salesman problem that has been introduced in [5]. In [11], a 2.5-approximation algorithm has been presented. In addition, a 3-approximation algorithm for the Prize-Collecting Steiner Tree problem has been proposed. Goemans and Williamson [25] provided 2-approximation algorithms for both problems, based on the primal-dual scheme. Later, Archer et al. [3] devised a $(2 - \epsilon)$ -approximation algorithm, for some $\epsilon > 0$, for both problems. These results are valid for the *rooted* and the *unrooted* variants of both problems. For MaxSat, a classical $3/4$ approximation algorithm has been improved in a sequence of articles, up to a ratio 0.7968 [4]. Garg et al. [24] provided a 2-approximation primal-dual algorithm for the multicut problem in trees, and a reduction from which a matching lower bound of $2 - \epsilon$, under UGC, follows.

2 General Rounding Scheme

In this Section, we will present a general rounding scheme for *multistage minimization problems* that can be formulated as the *ILP* in (1). Then, we will present a direct application of this scheme to obtain an f -approximation algorithm for multistage f -Set Cover.

As mentioned, the hardness of multistage optimization arises from the introduction of a moving cost between consecutive solutions of different instances for

an optimization problem. The general rounding scheme we give allows the use of traditional algorithms for an optimization problem, while achieving theoretical guarantees for the approximation ratio of the solution w.r.t. the moving cost.

On a high level, the general scheme fixes the variables x (which are subset to the moving cost) by using an LP relaxation for the multistage problem and a randomized rounding. Then, x^t being fixed for every t , we have T unrelated instances and we compute y^t independently on each time step.

We will now present the general algorithm scheme, which we refer to as **GS**. **GS** requires one threshold parameter $\alpha \in (0, 1]$ and an algorithm **ALG** that given a solution x^t for the problem produces a solution y^t such that $(x^t, y^t) \in \Delta$.

GS(α, ALG):

1. Solve the continuous relaxation of the multistage problem (the ILP (1) where $x_i^t, y_i^t \in [0, 1]$), to find an optimal solution \bar{x}, \bar{y} .
2. Choose a threshold h uniformly at random in $[0, \alpha]$.
3. For each $t \in [T]$ and $i \in [n_x]$, set $\hat{x}_i^t = 1$ if $\bar{x}_i^t \geq h$, otherwise set $\hat{x}_i^t = 0$.
4. For each $t \in [T]$, produce $\hat{y}^t = \mathbf{ALG}(\hat{x}^t)$.
5. Output (\hat{x}^t, \hat{y}^t) for each $t \in [T]$.

Formally, **ALG** also depends on the instance, that is the set of constraints Δ and the weight vectors c^t, d^t, w^t at time t in order to compute \hat{y}^t . Yet, we will slightly abuse the notation for the convenience of the reader and omit this dependency, writing $\hat{y}^t = \mathbf{ALG}(\hat{x}^t)$.

In order for this general rounding scheme to work, we need to make sure that the rounded values \hat{x} , combined with the output \hat{y} of **ALG** satisfy the linear set of constraints Δ at each time step. In order to achieve this, both the parameter α used to draw the random threshold and the underlying algorithm **ALG** need to be selected appropriately.

The main benefit of this rounding scheme is that it allows the use of a traditional optimization algorithm **ALG** for each (now unrelated) instance, while achieving theoretical guarantees both for the service cost and the moving cost of the x -variables. This can be formalized in the following theorem:

Theorem 1. *Assuming feasibility, the output of $\mathbf{GS}(\alpha, \mathbf{ALG})$ is $\frac{1}{\alpha}$ -approximate to the optimal (fractional) solution both w.r.t. the service cost of the x -variables, and to the moving cost, that is:*

1. $E(\sum_{t=1}^T \sum_{i=1}^{n_x} c_i^t \hat{x}_i^t) \leq \frac{1}{\alpha} \sum_{t=1}^T \sum_{i=1}^{n_x} c_i^t \bar{x}_i^t$.
2. $E(\sum_{t=2}^T \sum_{i=1}^{n_x} w_i^t |\hat{x}_i^t - \hat{x}_i^{t-1}|) \leq \frac{1}{\alpha} \sum_{t=2}^T \sum_{i=1}^{n_x} w_i^t |\bar{x}_i^t - \bar{x}_i^{t-1}|$.

Proof. Fix any time step $t \in [T]$ and any variable index $i \in [n_x]$. By linearity of expectation and the fact that all weights c_i^t, w_i^t are non-negative, it suffices to show that (i) $E(\hat{x}_i^t) \leq \frac{1}{\alpha} \bar{x}_i^t$ and (ii) $E(|\hat{x}_i^t - \hat{x}_i^{t-1}|) \leq \frac{1}{\alpha} |\bar{x}_i^t - \bar{x}_i^{t-1}|$.

If $\bar{x}_i^t \geq \alpha \geq h$ then with probability 1 we have that $\hat{x}_i^t = 1 \leq \frac{\bar{x}_i^t}{\alpha}$. If $\bar{x}_i^t < \alpha$, then \hat{x}_i^t becomes 1 only if $h \leq \bar{x}_i^t$ which happens with probability $\frac{\bar{x}_i^t}{\alpha}$ due to the

uniform selection of the threshold; otherwise it is rounded to 0. So, on expectation we have $E(\hat{x}_i^t) \leq \frac{\bar{x}_i^t}{\alpha}$ as required.

For the moving cost, a simple case-analysis suffices:

- If $\bar{x}_i^t \geq \alpha$ and $\bar{x}_i^{t-1} \geq \alpha$ then both are rounded to 1 and the algorithm does not suffer any moving cost.
- If $\bar{x}_i^t < \alpha$ and $\bar{x}_i^{t-1} < \alpha$ then the algorithm suffers a moving cost of 1 only if these variables are rounded to different values. This happens only if the threshold h is drawn between them. Since h is drawn uniformly at random from $[0, \alpha]$ we get that $E(|\hat{x}_i^t - \hat{x}_i^{t-1}|) = \frac{|\bar{x}_i^t - \bar{x}_i^{t-1}|}{\alpha}$ which satisfied the required property.
- If $\bar{x}_i^t < \alpha$ and $\bar{x}_i^{t-1} \geq \alpha$ (or the symmetrical case which is equivalent), then we have that $\hat{x}_i^{t-1} = 1$ and $\hat{x}_i^t = 0$ only if the threshold h is in $[\bar{x}_i^t, \alpha]$, so on expectation we get $E(|\hat{x}_i^t - \hat{x}_i^{t-1}|) = \frac{\alpha - \bar{x}_i^t}{\alpha} \leq \frac{\bar{x}_i^{t-1} - \bar{x}_i^t}{\alpha}$.

□

Basically, this theorem bounds the approximation ratio of the solution with regard to the x variables. To summarize, all we need in order to use this general rounding scheme is to select a parameter α (that is as large as possible in order to achieve the lowest possible approximation ratio) and an algorithm **ALG** such that the feasibility of the output is ensured. Observe that some theoretical guarantee about the approximation ratio of **ALG** w.r.t. the y -variables has to be derived in order to obtain a (collective) approximation ratio for the output (\hat{x}, \hat{y}) of **GS**(α, \mathbf{ALG}).

Furthermore, observe that the approximation ratios that stem from this Theorem are w.r.t. the optimal *fractional* solution, which is even stronger than an approximation ratio w.r.t. the optimal *integer* solution.

Finally, let us make a comment on *derandomization*. The only random choice we make is selecting h uniformly at random from $[0, \alpha]$. The way variables are rounded only depends on the comparisons between h and the values of \bar{x}_i^t . In other words, there are at most $nT + 1$ different executions of the randomized algorithms: $h = \bar{x}_i^t$ for any t, i (such that $\bar{x}_i^t \leq \alpha$), and possibly $h = \alpha$ (if all \bar{x}_i^t are smaller than α , then $h = \alpha$ corresponds to the case where all variables are rounded to 0). In other words, **GS** can be easily derandomized (at the price of applying $O(nT)$ times algorithm *ALG*).

Warm-Up: Multistage f -Set Cover. Before we use **GS** to solve more demanding problems, like *Prize-Collecting Problems*, we provide the reader with a simple application of the scheme that gives an f -approximation algorithm for multistage f -Set Cover.

In the (weighted) Set Cover problem, we have as input a ground set C and a collection (S_1, \dots, S_n) of subsets of C , each given with a weight c_i . The goal is to find a collection of minimum weight the union of which is C . In f -Set Cover, each element of the collection appears in at most f sets of the collection. We consider the multistage variant of this problem, where the weights on the n subsets of C

change over time, and a moving cost is associated with choosing set i at time step t but not at time step $t + 1$ (or vice versa). The *ILP* formulation for this problem is (note that there is no y -variable here):

$$\left\{ \begin{array}{l} \min \sum_{t=1}^T \sum_{i=1}^n c_i^t x_i^t + \sum_{t=2}^T \sum_{i=1}^n w_i^t |x_i^t - x_i^{t-1}| \\ \text{s.t.} \left| \begin{array}{l} \sum_{i:e \in S_i} x_i^t \geq 1 \quad \forall t \in [T], \forall e \in C \\ x_i^t \in \{0, 1\} \quad \forall t \in [T], \forall i \in [n] \end{array} \right. \end{array} \right.$$

In [6], the authors provide a $2f$ -approximation algorithm for multistage f -Set Cover. Yet, since the static version of the problem is hard to approximate only within a ratio of $f - \epsilon$ under *UGC* [32], this previous result possible had room for improvement. Indeed, we show that using *GS* allows for an optimal approximation ratio of f to be achieved. Since there are no y -variables in the formulation of the problem, we only need a parameter α that guarantees feasibility in order to use *GS*. This is captured by the following Theorem:

Theorem 2. *$GS(\frac{1}{f}, \emptyset)$ is an f -approximation algorithm for Multistage f -Set Cover.*

Proof. For $\alpha = \frac{1}{f}$ and the fact that there are no y -variables, Theorem 1 immediately guarantees an approximation ratio of f , provided that \hat{x} is feasible. From the linear constraints on the optimal solution and the fact that each element appears in at most f sets of the collection, we know that for any $t \in [T]$ and $e \in C$, there exists at least one set S_i such that $\hat{x}_i^t \geq \frac{1}{f} = \alpha \geq h$. This guarantees that the value will be rounded to 1, and thus the feasibility of the rounded solution that the algorithm outputs.

We note that Vertex Cover is the special case for f -Set Cover when $f = 2$. Thus, our results also translate to a 2-approximation algorithm for multistage Vertex Cover, which is also optimal under *UGC* [32], as an alternative proof of a result in [6].

3 Prize-Collecting problems

In this section, we study the multistage variants of Prize-Collecting Steiner Tree and Prize-Collecting Metric Traveling Salesman that were also considered in [6].

In a Prize-Collecting problem, clients that are represented as vertices v of a graph $G(V, E)$ address a request and come with a penalty c_v for not serving their request. An algorithm has to select some vertices of the graph to serve *some* of the requests, in order to minimize the sum of the service cost (sum of costs d_e of selected edges) and the penalties of not-served vertices. These problems can be formulated via the following *ILP*

$$\left\{ \begin{array}{l} \min \sum_{v \in V} c_v x_v + \sum_{e \in E} d_e y_e \\ \text{s.t.} \left| \begin{array}{l} (x, y) \in \Delta \\ x_v, y_e \in \{0, 1\} \end{array} \right. \end{array} \right.$$

where variable x_v is 1 if vertex v is *not* served and variable y_e is 1 if edge e is selected. In this formulation, c_v denotes the penalty of vertex v and d_e denotes the cost of edge e . The set of linear constraints Δ is selected appropriately for each problem so that (i) the structure of the problem is respected and (ii) the sets of selected edges and not-served vertices coincide. Specifically:

- For Prize-Collecting Steiner Tree (PCST), we have a (static) root $r \in V$ and need to select a set of edges, while minimizing the cost of the selected edges and the penalties of the vertices that are not connected to the root via the selected edges. The set of constraints Δ simply states that x_v cannot become 0 unless there is a path of edges with $y_e = 1$ that connect v to the root r .
- For Prize-Collecting Metric Travelling Salesman Problem (PCMTSP), we assume that the graph $G(V, E)$ is complete and the weights of the edges d_e respect the triangular inequality. Once again, we have a (static) depot $r \in V$ and need to select a set of edges that forms a tour that includes r , while minimizing the cost of the selected edges and the penalties of the vertices that are not visited in the tour. The set of constraints Δ simply states that y_e must form a circular path that includes r and that x_v cannot become 0 unless it is visited by the tour that is formed by edges with $y_e = 1$.

In the multistage version we consider here, the cost of edges and penalties evolve with time, in a time horizon $1, \dots, T$. Depending on these costs, we may decide to serve different requests at different time steps. However, changing this decision to serve client/vertex v between $t - 1$ and t comes with a *moving cost* w_v^t . Then, the *ILP* formulation of the problem becomes

$$\begin{cases} \min & \sum_{t=1}^T \sum_{v \in V} c_v^t x_v^t + \sum_{t=1}^T \sum_{e \in E} d_e^t y_e^t + \sum_{t=2}^T \sum_{v \in V} w_v^t |x_v^t - x_v^{t-1}| \\ \text{s.t.} & (x^t, y^t) \in \Delta \quad \forall t \in [T] \\ & x_v^t, y_e^t \in \{0, 1\} \end{cases}$$

where Δ is specified appropriately for multistage PCST and multistage PCMTSP.

Some of the authors of this article have already studied these multistage problems and have provided upper bounds on their approximation ratio. Using the same approach, we show that our novel rounding scheme presented in this work can improve the approximation ratio for both of these problems.

We will use **GS** to solve these problems: initially, we solve their continuous relaxation to get optimal fractional solutions (\bar{x}, \bar{y}) . Then, we select an appropriate α and a random threshold $h \in [0, \alpha]$ to round \bar{x} to integer solutions \hat{x}^t for each time-step t . This step fixes the sets of vertices that we need to serve at each time-step and now we need to specify an algorithm that decides on edges that have to be selected in order to serve these requests (independently at each time step).

Note that once the set of served vertices is fixed, choosing the edges to serve these clients is a classical Steiner tree problem (for PCST) and a classical TSP problem (for PCMTSP). Then, for PCST, we will use the Steiner tree algorithm **ST** from [25] that is 2-approximate for the Steiner Tree problem, and for PCMTSP

we will use Christofides' algorithm TSP from [13] which is $\frac{3}{2}$ -approximate for Metric TSP. The approximation of both of these algorithms is w.r.t. the optimal *fractional* solution to the problem, which is a property that we are going to use. The feasibility of the output is ensured, since for any set of vertices that we need to serve (as a result of the rounding on the x -variables) these algorithms will output an appropriate set of edges. Furthermore, the approximation ratio w.r.t. the penalties and the moving cost is $\frac{1}{\alpha}$ from Theorem 1. Thus, we only need to argue about the cost of the edges.

Let (\bar{x}, \bar{y}) be the optimal fractional solution for either PCST or PCMTSP, α be the parameter of **GS** (to be specified), h be the random threshold. Let also (\hat{x}, \hat{y}) denotes the solution output by the algorithm, i.e., \hat{x} obtained in the rounded scheme and \hat{y} the set of edges obtained using the Steiner tree algorithm from [25] (for PCST) or Christofides' algorithm (for PCMTSP) on the set of chosen served vertices.

The following property was proven in [6] (see the proofs of Theorem 1 and 3 there):

Lemma 1. [6] *If $\bar{x}_v^t \geq 1 - \beta$ for all v such that $\hat{x}_v^t = 1$, then $\sum_{e \in E} d_e^t \hat{y}_e^t \leq \lambda \sum_{e \in E} d_e^t \bar{y}_e^t$, where $\lambda = \frac{2}{\beta}$ for PCST and $\lambda = \frac{3}{2\beta}$ for PCMTSP.*

We are now ready to present the main theorems of this section.

Theorem 3. *With $\alpha = 1 - e^{-1/2}$, **GS**(α, ST) gives a $\frac{1}{1 - e^{-1/2}} \leq 2.55$ approximation algorithm for multistage Prize-Collecting Steiner tree.*

Proof. Fix any $t \in T$. By the definition of the rounding scheme, if $\hat{x}_v^t = 1$ then $\bar{x}_v^t \geq h$. Using Lemma 1 (with $\beta = 1 - h$) we get:

$$\sum_{e \in E} d_e^t \hat{y}_e^t \leq 2 \sum_{e \in E} d_e^t \frac{\bar{y}_e^t}{1 - h}$$

By computing the expectation on the random variable h that is drawn uniformly in $[0, \alpha]$ we get:

$$E \left(\sum_{e \in E} d_e^t \hat{y}_e^t \right) \leq 2 \sum_{e \in E} d_e^t \bar{y}_e^t \frac{1}{\alpha} \int_0^\alpha \frac{dh}{1 - h} \leq -\frac{2}{\alpha} \ln(1 - \alpha) \sum_{e \in E} d_e^t \bar{y}_e^t$$

Thus, combining with the results of Theorem 1, the approximation ratio of **GS**(α, ST) is $\max(\frac{1}{\alpha}, -\frac{2}{\alpha} \ln(1 - \alpha))$ and the Theorem follows from balancing these terms by choosing $\alpha = 1 - e^{-1/2}$. \square

Theorem 4. *With $\alpha = 1 - e^{-2/3}$, **GS**(α, TSP) gives a $\frac{1}{1 - e^{-2/3}} \leq 2.056$ approximation algorithm for the multistage Prize-Collecting Metric Travelling Salesman Problem.*

Proof. The proof is identical to that of Theorem 3 with the only difference that the ratio for the cost of edges is $\frac{3}{2(1-h)}$. Then, **GS**(α, TSP) is $\max(\frac{1}{\alpha}, -\frac{3}{2\alpha} \ln(1 - \alpha))$ -approximate, and the Theorem follows from choosing $\alpha = 1 - e^{-2/3}$. \square

4 Multi-Cut on Trees

In this section, we will provide a 2-approximation algorithm for multistage Multi-Cut on Trees, using a slight modification of our general rounding scheme. In the traditional multi-cut problem, we are given an undirected graph $G = (V, E)$ with weights $c : E \rightarrow \mathbb{R}_+$ and a set of k -pairs $(s_i, t_i) \in V \times V$. A multi-cut $C \subseteq E$ is any set of edges such that s_i and t_i are disconnected in $G' = (V, E \setminus C) \forall i \in [k]$. Our objective is to determine a multi-cut on the graph with minimum weight.

For $k \geq 3$, it has been shown that the problem is NP -hard. Even when G is restricted to trees, it has been shown from a reduction from vertex cover that the multi-cut problem is NP -hard to approximate within 1.3 of the optimal solution [14]. Furthermore, under UGC , the problem is hard to approximate within $(2 - \epsilon)$ of the optimal solution for any $\epsilon > 0$ [33].

We focus on the case where G is a tree. In this case, there is only one path P_i between vertices s_i and t_i and thus, the problem reduces to selecting at least one edge from every path P_i . Then, the problem can be formulated as the following integer program:

$$\begin{cases} \min. & \sum_{e \in E} c_e x_e \\ \text{s.t.} & \sum_{e \in P_i} x_e \geq 1, \quad i = 1, \dots, k \\ & x_e \in \{0, 1\}, \quad e \in E \end{cases}$$

where $x_e = 1$ means that edge $e \in E$ belongs to the multi-cut. A way to achieve an approximation ratio of 2, which is optimal on trees under UGC , is to use LP-rounding.

Specifically, the algorithm⁷ that achieves this results first solves the continuous relaxation of the problem to obtain optimal fractional solutions $\bar{x}_e \in [0, 1]$, picks any vertex $r \in V$ as the "root" and compute $d(r, u)$ to be equal to the sum of values \bar{x}_e on the edges e of the path between r and any vertex u .

We say that an edge $e = (u, v) \in E$ is "cut" by some $a > 0$ iff $d(r, u) \leq a$ and $d(r, v) > a$. The algorithm picks a random $R \in [0, \frac{1}{2}]$ and rounds to 1 only the edges that are cut by $R + \frac{k}{2}$ for some $k \in \mathbb{N}$. It can be shown that this rounding scheme gives a feasible solution, and that the probability that edge e is selected is at most $2\bar{x}_e$, leading to an approximation ratio of 2.

Multistage Multi-Cut on Trees We consider the multistage version of multi-cut on trees where we have T costs functions on edges. Since the decision variables are on the edges, we consider a moving cost for changing an edge (in or out of the multi-cut) between consecutive solutions. We can easily write the multistage multi-cut problem as an integer program.

$$\begin{cases} \min. & \sum_{t=1}^T \sum_{e \in E} c_e^t x_e^t + \sum_{t=2}^T \sum_{e \in E} w_e^t |x_e^t - x_e^{t-1}| \\ \text{s.t.} & \sum_{e \in P_i} x_e^t \geq 1, & i = 1, \dots, k, \quad t = 1, \dots, T \\ & x_e^t \in \{0, 1\}, & e \in E, \quad t = 1, \dots, T \end{cases}$$

⁷ Folklore, see <https://www.cs.cmu.edu/~anupamg/adv-approx/lecture18.pdf>

We use a modification of GS, where in Step 2 we use as rounding scheme the one of the static problem given before: pick $R \in [0, \frac{1}{2}]$ at random and then at each time step select edges that are "cut" by $R, R + \frac{1}{2}$, etc. This will give feasibility and a 2-approximation w.r.t. the service cost as in the traditional case. What we need to do now is bound the moving cost. To be able to do this, we pick the same R for each time step. Let us call this algorithm Mod-GS.

Theorem 5. *Mod-GS is 2-approximate for multistage multi-cut on trees.*

Proof. By the discussion above, we only have to bound the moving cost. The proof is similar to the one of Theorem 1.

1. If $\bar{x}_e^{t-1} \geq \frac{1}{2}$ and $\bar{x}_e^t \geq \frac{1}{2}$ then edge e is rounded to 1 (included) in both time-steps and the moving cost is 0 and (clearly) at most the optimal.
2. If $\bar{x}_e^{t-1} < \frac{1}{2}$ and $\bar{x}_e^t \geq \frac{1}{2}$ then edge e is rounded to 1 at time t and to 1 at time $t-1$ with probability $2\bar{x}_e^{t-1}$. Thus, the expected moving cost is $(1 - 2\bar{x}_e^{t-1}) = 2(\frac{1}{2} - \bar{x}_e^{t-1}) \leq 2(\bar{x}_e^t - \bar{x}_e^{t-1})$ and thus the moving cost is at most 2 times the moving cost of the optimal solution.
3. If $\bar{x}_e^{t-1} \geq \frac{1}{2}$ and $\bar{x}_e^t < \frac{1}{2}$ we use the same analysis as in 2.
4. If $\bar{x}_e^{t-1} \leq \bar{x}_e^t < \frac{1}{2}$ since we use the same R randomly picked from $[0, \frac{1}{2}]$ we know that x_e^t and x_e^{t-1} are rounded to different values with probability $\frac{|x_e^t - x_e^{t-1}|}{\frac{1}{2}}$ which is exactly 2 times the moving cost of the optimal solution. □

5 Multistage Maximum Satisfiability

Until this point, we have studied exclusively multistage minimization problems. However, as we show, the general rounding scheme we have proposed can easily be transferred to maximization settings as well. To illustrate this, we study a multistage variant of the Max-Sat problem in this section.

In the traditional (weighted) Max-Sat problem, we are given a set of clauses C_1, C_2, \dots, C_m that are logical disjunctions of some boolean variables x_1, x_2, \dots, x_n . Each clause C_i comes with a nonnegative weight c_i . Our goal is to determine an assignment for the boolean variables x_i , such that the sum of the weights of the satisfied clauses is maximized.

We consider the problem in a multistage framework by having T weight-functions c^t (c_j^t being the weight of clause C_j at time t). We also have a stability revenue $w_i^t(1 - |x_i^t - x_i^{t-1}|)$ that awards the algorithm with w_i^t for not changing the assignment of a boolean variable x_i between time steps $t-1$ and t .

We note that this setting also captures the case where the set of clauses can change over time. In fact, this is equivalent to having a (static) set of all the clauses that appear throughout the instance and simply assigning weight $c_p^t = 0$ to clause C_p if it does not appear in time step t .

An *ILP* formulation for the multistage problem is the following:

$$\begin{cases} \max & \sum_{t=1}^T \sum_{i=1}^m c_i^t y_i^t + \sum_{t=2}^T \sum_{i=1}^n w_i^t (1 - |x_i^t - x_i^{t-1}|) \\ \text{s.t.} & (x^t, y^t) \in \Delta \quad \forall t \in [T] \\ & x_i^t, y_i^t \in \{0, 1\} \end{cases}$$

Once again, we can use z -variables to substitute the absolute difference of x -values in the formulation of the problem. $x_i^t = 1$ if at time t the boolean value of variable x_i is TRUE, and $y_i^t = 1$ if the assignment at time t satisfies clause C_i . For the set of constraints Δ , we use the traditional modeling of the problem: based on the structure of a clause C_i , y_i is allowed to take the value 1 only if at least one of the literals is served.

A well-known result for Max-Sat is that we can get a $\frac{3}{4}$ -approximation solution by combining the solutions of two different algorithms [37]: the first one is based on randomized rounding of an optimal solution of a continuous relaxation, and the second one simply fixes the truth variables at random independently with probability 1/2. We will show that the same holds for the multistage variation of the problem.

Let us first consider the first one, which consists in applying GS with $\alpha = 1$. A slight modification of the rounding scheme is that we now draw a separate threshold h_i for each boolean variable x_i and use it for every time step. Since the threshold is drawn uniformly in $[0, \alpha]$, this basically means that each variable is randomly set to TRUE with probability that is equal to the optimal solution of the relaxed linear program. This directly gives the set of satisfied clauses at each time step - so variables y representing clauses can be immediately deduced once the truth variables x_i^t have been fixed.

Lemma 2. *GS with $\alpha = 1$ is a $(1-1/e)$ -approximation algorithm for Multistage Max Sat.*

Proof. While Theorem 1 holds for minimization problems, using the exact same arguments we can easily get that for $\alpha = 1$ it holds that $E(|\hat{x}_i^t - \hat{x}_i^{t-1}|) \leq |\bar{x}_i^t - \bar{x}_i^{t-1}|$. Hence, the expected stability revenue is at least as large as the one of the optimal solution.

For the *service revenue*, using standard arguments we get that the expected service revenue is at least $\sum_t \sum_p c_p^t \cdot \left(1 - \left(1 - \frac{1}{|C_p|}\right)^{|C_p|}\right) \bar{y}_p^t$. Indeed, the probability that C_p is satisfied at step t is $P = 1 - \prod_{x_i \in C_p} (1 - \bar{x}_i^t) \prod_{\bar{x}_i \in C_p} \bar{x}_i^t$. Using the geometric-arithmetic means inequality $\prod_{i=1}^b \alpha_i^{1/b} \leq \sum_{i=1}^b \alpha_i / b$, we get:

$$\begin{aligned} P &\geq 1 - \left(\frac{\sum_{x_i \in C_p} (1 - \bar{x}_i^t) + \sum_{\bar{x}_i \in C_p} \bar{x}_i^t}{|C_p|} \right)^{|C_p|} \\ &= 1 - \left(1 - \frac{\sum_{x_i \in C_p} \bar{x}_i^t + \sum_{\bar{x}_i \in C_p} (1 - \bar{x}_i^t)}{|C_p|} \right)^{|C_p|} \geq 1 - \left(1 - \frac{\bar{y}_p^t}{|C_p|} \right)^{|C_p|} \end{aligned}$$

As $1 - (1 - t/k)^k \geq 1 - (1 - 1/k)^k t$ for $0 \leq t \leq 1$, we get that $P \geq (1 - (1 - 1/|C_p|)^{|C_p|} \bar{y}_p^t)$, leading to the claimed lower bound on the service revenue.

Now, as $(1 - 1/k)^k \leq 1/e$ for any integer $k \geq 1$ the result follows. \square

Now, we generalize to the multistage setting the (trivial) classical algorithm which consists of choosing x_i to true with probability $1/2$, independently on each variable. In the multistage setting, we do the same but do not flip x_i^t and x_i^{t+1} independently. Rather, we choose x_i^1 at random, and then fix $x_i^t = x_i^1$ for any t . Doing this we get full stability revenue. Let us call *AlgoMaxSat* the algorithm which outputs the best solution between this randomized algorithm, and the one in Lemma 2.

Theorem 6. *AlgoMaxSat is a 3/4-approximation algorithm for Multistage Max Sat.*

Proof. Let us first focus on the stability revenue: the trivial rounding gets full revenue (no variable changes over time), while, as shown in the proof of Lemma 2, the stability revenue in *GS* is at least the one of the optimal fractional solution. So in both cases we have no loss with respect to the stability revenue.

For the service revenue, the analysis is the same as in the static case. In the trivial rounding, since the boolean variables are assigned values at random, the probability that the clause C_p is satisfied at time t is exactly $(1 - \frac{1}{2^{|C_p|}})$ and thus, the expected service revenue of the algorithm $E_1 = \sum_t \sum_p c_p^t \cdot (1 - \frac{1}{2^{|C_p|}})$.

Recall that the one of *GS* is at least $E_2 = \sum_t \sum_p c_p^t \cdot \left(1 - \left(1 - \frac{1}{|C_p|}\right)^{|C_p|}\right) \bar{y}_p^t$.

Using the fact that, for any integer $\ell \geq 1$, $\left(1 - \left(1 - \frac{1}{\ell}\right)^\ell\right) + \left(1 - \frac{1}{2^\ell}\right) \geq 3/2$, we get that the expected service revenue of *AlgoMaxSat* is at least

$$\frac{E_1 + E_2}{2} \geq \sum_t \sum_p c_p^t \frac{3\bar{y}_p^t}{4}$$

\square

6 Conclusion

We have presented some approximation algorithms for various multistage problems using a general rounding scheme.

In the formulation we considered, the set of feasible solutions is static, only the weights in the objective function evolve with time. However, the results presented in this work could also apply to the more general setting where the set of linear constraints changes over time, that is we have $(x^t, y^t) \in \Delta_t$. For instance, in the f -set cover problem elements may be added or removed from sets. As long as the frequency remain at most f , the result holds. Similarly, literals can be added or removed from a clause in the MaxSat problem.

One might also consider the case where the set of variables is not fixed. For instance, a new set may appear at some time step in f -set cover. Our formulation

can also capture this more dynamic setting. For example, if a new variable x_k is added at time-step t , then we can assume that this variable always existed and simply had $c_k^\tau = d_k^\tau = 0$ for all $\tau < t$ and $w_k^\tau = 0$ for all $\tau \leq t$. Since we focus on the offline case where the whole sequence is known in advance, this can happen without loss of generality.

As possible extensions and future works, a first direction would be to consider other mathematical programming techniques in the multistage setting. For instance, SDP techniques can be applied to the multistage setting, and approximation results for problems as Max Cut or Max 2Sat generalize to multistage versions of the problem.

A second direction would be to consider other ways to take into account moving costs in the multistage framework. For instance, if the moving cost is not put into the objective function but considered as a constraint (we have a budget on this cost), can LP-based techniques lead to interesting approximation algorithms in this case as well?

Acknowledgements This work was partially funded by the grant ANR-19-CE48-0016 from the French National Research Agency (ANR). The research of the fourth author is supported by the Russian Science Foundation, grant 21-41-09017.

References

1. Albers, S.: On energy conservation in data centers. In: proc. SPAA. pp. 35–44 (2017)
2. An, H., Norouzi-Fard, A., Svensson, O.: Dynamic facility location via exponential clocks. *ACM Trans. Algorithms* **13**(2), 21:1–21:20 (2017)
3. Archer, A., Bateni, M., Hajiaghayi, M., Karloff, H.J.: Improved approximation algorithms for prize-collecting steiner tree and TSP. *SIAM J. Comput.* **40**(2), 309–332 (2011)
4. Avidor, A., Berkovitch, I., Zwick, U.: Improved approximation algorithms for MAX NAE-SAT and MAX SAT. In: proc. WAOA 2005. *Lecture Notes in Computer Science*, vol. 3879, pp. 27–40. Springer (2005)
5. Balas, E.: The prize collecting traveling salesman problem. *Networks* **19**(6), 621–636 (1989)
6. Bampis, E., Escoffier, B., Kononov, A.: LP-based algorithms for multistage minimization problems. In: proc. WAOA 2020. *Lecture Notes in Computer Science*, vol. 12806, pp. 1–15. Springer (2020)
7. Bampis, E., Escoffier, B., Lampis, M., Paschos, V.T.: Multistage matchings. In: proc. SWAT. pp. 7:1–7:13 (2018)
8. Bampis, E., Escoffier, B., Mladenovic, S.: Fair resource allocation over time. In: proc. AAMAS. pp. 766–773 (2018)
9. Bampis, E., Escoffier, B., Schewior, K., Teiller, A.: Online multistage subset maximization problems. In: proc. ESA. pp. 11:1–11:14 (2019)
10. Bampis, E., Escoffier, B., Teiller, A.: Multistage knapsack. In: proc. MFCS. pp. 22:1–22:14 (2019)
11. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.P.: A note on the prize collecting traveling salesman problem. *Math. Prog.* **59**, 413–420 (1993)

12. Chimani, M., Troost, N., Wiedera, T.: Approximating multistage matching problems. In: *proc IWOCA 2021. Lecture Notes in Computer Science*, vol. 12757, pp. 558–570. Springer (2021)
13. Christofides, N.: Worst-case analysis of a new heuristics for the traveling salesman problem. In: Report 388, Graduate School of Industrial Administration, Carnegie-Mellon Univ. (1976)
14. Dinur, I., Safra, S.: The importance of being biased. *Electron. Colloquium Comput. Complex.* (104) (2001)
15. Eisenstat, D., Mathieu, C., Schabanel, N.: Facility location in evolving metrics. In: *proc. ICALP*. pp. 459–470 (2014)
16. Eppstein, D.: Offline algorithms for dynamic minimum spanning tree problems. *J. Algorithms* **17**(2), 237–250 (1994)
17. Fairstein, Y., Kulik, A., Naor, J.S., Raz, D.: General knapsack problems in a dynamic setting. In: *proc. APPROX/RANDOM 2021. LIPIcs*, vol. 207, pp. 15:1–15:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021)
18. Fairstein, Y., Naor, S., Raz, D.: Algorithms for dynamic NFV workload. In: *proc. WAOA 2018. Lecture Notes in Computer Science*, vol. 11312, pp. 238–258. Springer (2018)
19. Fluschnik, T.: A multistage view on 2-satisfiability. In: *proc. CIAC 2021. Lecture Notes in Computer Science*, vol. 12701, pp. 231–244. Springer (2021)
20. Fluschnik, T., Niedermeier, R., Rohm, V., Zschoche, P.: Multistage vertex cover. In: *proc. IPEC. LIPIcs*, vol. 148, pp. 14:1–14:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
21. Fluschnik, T., Niedermeier, R., Schubert, C., Zschoche, P.: Multistage s-t path: Confronting similarity with dissimilarity in temporal graphs. In: *proc. ISAAC 2020. LIPIcs*, vol. 181, pp. 43:1–43:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
22. Fotakis, D., Kavouras, L., Kostopanagiotis, P., Lazos, P., Skoulakis, S., Zarifis, N.: Reallocating multiple facilities on the line. *Theor. Comput. Sci.* **858**, 13–34 (2021)
23. Fotakis, D., Kavouras, L., Zakyntinou, L.: Online facility location in evolving metrics. *Algorithms* **14**(3), 73 (2021)
24. Garg, N., Vazirani, V.V., Yannakakis, M.: Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica* **18**(1), 3–20 (1997)
25. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. *SIAM J. Comput.* **24**(2), 296–317 (1995)
26. Gupta, A., Talwar, K., Wieder, U.: Changing bases: Multistage optimization for matroids and matchings. In: *proc. ICALP*. pp. 563–575 (2014)
27. Håstad, J.: Some optimal inapproximability results. *J. ACM* **48**(4), 798–859 (2001)
28. Hochbaum, D.S.: Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput.* **11**(3), 555–556 (1982). <https://doi.org/10.1137/0211045>, <https://doi.org/10.1137/0211045>
29. Hochbaum, D.S.: Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *European Journal of Operational Research* **140**(2), 291–321 (2002)
30. Joseph, V., de Veciana, G.: Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In: *proc. IEEE INFOCOM*. pp. 567–575 (2012)
31. Kempe, D., Kleinberg, J.M., Kumar, A.: Connectivity and inference problems for temporal networks. In: Yao, F.F., Luks, E.M. (eds.) *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, May 21–23, 2000, Portland, OR, USA. pp. 504–513. ACM (2000)

32. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.* **74**(3), 335–349 (2008)
33. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.* **74**(3), 335–349 (2008)
34. Olver, N., Pruhs, K., Sitters, R., Schewior, K., Stougie, L.: The itinerant list-update problem. In: *proc. WAOA*. pp. 310–326 (2018)
35. Rottner, C.: *Combinatorial Aspects of the Unit Commitment Problem*. Ph.D. thesis, Sorbonne Université (2018)
36. Sleator, D.D., Tarjan, R.E.: Amortized efficiency of list update and paging rules. *Commun. ACM* **28**(2), 202–208 (1985)
37. Williamson, D.P., Shmoys, D.B.: *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edn. (2011)

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:

On the behalf of the authors of the article,
Bruno Escoffier

