Multistage Matchings

Evripidis Bampis
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, Paris, France
evripidis.bampis@lip6.fr

Bruno Escoffier
Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, Paris, France
bruno.escoffier@lip6.fr

Michael Lampis
Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France
michail.lampis@dauphine.fr

Vangelis Th. Paschos
Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France
vangelis.paschos@dauphine.fr

Abstract

We consider a *multistage* version of the Perfect Matching problem which models the scenario where the costs of edges change over time and we seek to obtain a solution that achieves low total cost, while minimizing the number of changes from one instance to the next. Formally, we are given a sequence of edge-weighted graphs on the same set of vertices V, and are asked to produce a perfect matching in each instance so that the total edge cost plus the transition cost (the cost of exchanging edges), is minimized. This model was introduced by Gupta et al. (ICALP 2014), who posed as an open problem its approximability for bipartite instances. We completely resolve this question by showing that *Minimum Multistage Perfect Matching* (Min-MPM) does not admit an $n^{1-\epsilon}$-approximation, even on bipartite instances with only two time steps.

Motivated by this negative result, we go on to consider two variations of the problem. In *Metric Minimum Multistage Perfect Matching* problem (Metric-Min-MPM) we are promised that edge weights in each time step satisfy the triangle inequality. We show that this problem admits a 3-approximation when the number of time steps is 2 or 3. On the other hand, we show that even the metric case is APX-hard already for 2 time steps. We then consider the complementary maximization version of the problem, *Maximum Multistage Perfect Matching* problem (Max-MPM), where we seek to maximize the total profit of all selected edges plus the total number of non-exchanged edges. We show that Max-MPM is also APX-hard, but admits a constant factor approximation algorithm for any number of time steps.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Perfect Matching, Temporal Optimization, Multistage Optimization

Digital Object Identifier 10.4230/LIPIcs.SWAT.2018.7

Acknowledgements This work benefited from the support of the FMJH program "Gaspard Monge in optimization and Operation Research" and from the support to this program from EDF, via the project 2016-1760H/C16/1507 "Stability versus Optimality in Dynamic Environment Algorithms".

1 Introduction

In classical Combinatorial Optimization, given an instance of a problem the goal is to find a solution optimizing the value of the objective function. However, in many applications the

16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018).
Editor: David Eppstein; Article No. 7; pp. 7:1–7:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

instance may change over time and the goal is to find a tradeoff between the quality of the solution in each time step and the stability of the solution in consecutive time steps. As an example, consider an instance of an assignment problem, where the goal is to compute the best assignment of tasks to workers, assuming that we know the cost c_{ij} of performing task j by worker i. In the classical setting, it is possible to choose the assignment that minimizes the total cost in polynomial time. When the costs change over time (as for instance when a worker is not able to do some long task on a very busy day (infinite cost)) the optimal solutions of each time step may differ, inducing a transition cost for setting new task-worker pairs between two consecutive solutions. Hence, the naïve approach of finding a new optimal solution in each time step has the drawback that it does not take care of the penalty (transition cost) that is induced by the changes in the solution.

In this paper we study a multistage version of the *Perfect Matching problem* that follows this motivation and was originally introduced by Gupta, Talwar, and Wieder [11]. In this problem we are given a time horizon: $t = 1, 2, \dots, T$ where for each time t we are given an instance G_t of *Perfect Matching* (that is, an edge-weighted graph) on the same set of vertices V. The goal is to determine a sequence of solutions $S = (M_1, M_2, \dots, M_T)$ that both (1) are near-optimal (*quality*), and (2) induce small transition costs (*stability*). In other words, the goal is to determine a sequence of perfect matchings, one for each stage (time step) t, such that their total cost is small and the solution does not change too radically from one step to the next.

It was shown in [11] that this multistage problem is significantly harder than classical *Perfect Matching*. In fact, it is NP-hard to even approximate the optimal solution within $n^{1-\epsilon}$, for instances with only 8 times steps. Gupta et al. then posed as an explicit question whether the problem becomes easier for bipartite instances. Their work suggests also the question whether this hardness also applies for fewer than 8 steps. The bipartite restriction is especially interesting because Gupta et al. showed that related matroid-based optimization problems remain tractable for $T = 2$, and bipartite *Perfect Matching* can be seen as a matroid intersection problem. One could therefore hope that the matroid structure might make the bipartite case tractable for some small values of T, or at least approximable.

Our main contribution in this paper is to settle this question from [11] in the negative: we show that *Minimum Multistage Perfect Matching* (Min-MPM) is $n^{1-\epsilon}$-inapproximable, even for $T = 2$ time steps, unless $P = NP$. Motivated by this very negative result, we then investigate two other version of the problem: the *Metric Minimum Multistage Perfect Matching* problem (Metric-Min-MPM), where the input is guaranteed to satisfy the triangle inequality, and the *Maximum Multistage Perfect Matching* problem (Max-MPM), where we consider the complementary optimization objective.

Problem definition. Formally, the Min-MPM problem is defined as follows: We are given a sequence $G_1, \dots, G_T$ of T undirected graphs, on the same set of vertices V. At each time step $1 \leq t \leq T$, the graph G_t is given with a cost function c_t on edges: $c_t(e) \in \mathbb{Q}_{\geq 0} \cup \{+\infty\}$. We are also given a transition cost $M \geq 0$. A solution is a sequence $S = (M_1, \dots, M_t)$ where M_t is a perfect matching of G_t. Each solution (sequence) has two costs: a *matching cost* $c(S)$ and a *transition cost* $D(S)$. The goal is to minimize $c(S) + D(S)$. A matching M_t has a matching cost $c_t(M_t)$ which is equal to the sum of the costs of the edges of the perfect matching. The *matching cost* of S is $c(S) = \sum_{t=1}^{T} c_t(M_t)$. The *transition cost* is defined as $D(S) = \sum_{t=1}^{T-1} D_t$, where $D_t = M \cdot |M_{t+1} \setminus M_t|$ is proportional to the number of edges removed between time t and $t + 1$ – which is equal to the number of added edges since the matchings are perfect. Notice that by allowing infinite cost on edges we may assume w.l.o.g. the graphs to be complete.

In the Metric-Min-MPM, at each stage c_t obeys the triangle inequality: $c_t(u,v) + c_t(v,w) \geq c_t(u,w)$. Finally, in the Max-MPM version, we consider that $c_t(e)$ is the *profit* obtained by taking edge e (at time t). Then a solution sequence S has a *matching profit* $c(S) = \sum_t c_t(M_t)$. We define the *transition profit* $D(S)$ as $D(S) = \sum_{t \leq T-1} D_t$ where $D_t = M \cdot |M_{t+1} \cap M_t|$ is proportional to the number of edges that *remain* between time t and $t+1$. The goal now is to maximize $c(S) + D(S)$. Notice that in Max-MPM, we may no longer assume that the graphs are complete, since this assumption modifies the problem (we get profit by maintaining an edge, even of profit 0, from one time step to the next one).

Related work. A model that is close to our setting is the reoptimization model of Schieber et al. [15]. In their work, they are given a starting solution and a new instance and the goal is to minimize the sum of the cost of the new instance and of the transition cost. The model of multistage optimization that we use in this work has been studied earlier by Buchbinder et al. [5] and Buchbinder, Chen and Naor [4] for solving a set of fractional problems. Eisenstat et al. [7] studied a similar multistage optimization model for facility location problems. Their main result was a logarithmic approximation algorithm, which was later improved to a constant factor approximation by An et al. [1]. More broadly, many classical optimization problems have been considered in online or semi-online settings, where the input changes over time and the algorithm tries to adjust the solution (re-optimize) by making as few changes as possible. We refer the reader to [2, 3, 6, 10, 13, 14] and the references therein.

As mentioned, Gupta et al. [11] studied the Multistage Maintenance Matroid problem for both the offline and the online settings. Their main result was a logarithmic approximation algorithm for this problem, which includes as a special case a natural multistage version of Spanning Tree. The same paper also introduced the study of Min-MPM, which is the main problem we study here. They showed that the problem becomes hard to approximate even for a constant number of stages. More precisely, they showed the following result (n denotes the number of vertices in the graphs).

▶ **Theorem 1** ([11]). *For any $\epsilon > 0$, Min-MPM is not $n^{1-\epsilon}$-approximable unless $P = NP$. This holds even when the costs are in $\{0, \infty\}$, $M = 1$, and the number of time steps is a constant.*

Theorem 1 is proved for $T = 8$, starting from the fact that 3-colorability is NP-hard in graphs of maximum degree 4 [8]. The authors leave as an open question the approximability of the problem in bipartite graphs, and ask for subcases with better approximatibility behavior.

Our contribution. We answer the open question of [11] by showing that the problem is hard to approximate even for bipartite graphs and for the case of two steps ($T = 2$). Then, we focus on the case where the edge costs are metric within every time step (Metric-Min-MPM). On the negative side, we prove that the problem remains APX-hard even if $T = 2$. On the positive side, we show that Metric-Min-MPM admits a 3-approximation algorithm for two and three stages. Finally, for the maximization version of the problem, Max-MPM, we prove that it admits a constant factor approximation algorithm but is APX-hard.

2 Min-MPM for bipartite graphs

We answer the open question of [11] about the approximability of bipartite Min-MPM.

▶ **Theorem 2.** *For any $\epsilon > 0$, Min-MPM cannot be approximated within a factor of $n^{1-\epsilon}$, even if the input has $T = 2$ time steps, the input graphs are bipartite, $M = 1$ and the costs of edges are in $\{0, \infty\}$, unless P=NP.*

Using infinite costs, the same result immediately holds for bipartite complete graphs, as well as for complete graphs.

Proof. We give a gap-introducing reduction from Perfect 3DM (3-Dimensional Matching), known to be NP-complete [9]. We are given an instance of Perfect 3DM which consists of three sets X, Y, Z, with $|X| = |Y| = |Z| = n$, and a set Q of elements of $X \times Y \times Z$, with $|Q| = m \leq n^3$. We are whether there exists a subset of n pair-wise disjoint elements of Q, or not.

We construct an instance of our problem as follows: first, we create four sets of vertices A, B, C, D with $|A| = |B| = n$ and $|C| = |D| = m$. To ease notation suppose that the elements of our sets X, Y, Z, Q, A, B, C, D are labeled as $\{x_1, \dots, x_n\}$, $\{y_1, \dots, y_n\}$, $\{z_1, \dots, z_n\}$, $\{q_1, \dots, q_m\}$, $\{a_1, \dots, a_n\}$, $\{b_1, \dots, b_n\}$, $\{c_1, \dots, c_m\}$, and $\{d_1, \dots, d_m\}$ respectively.

For any $j \in \{1, \dots, m\}$ we construct a set of $2n^{\lceil \frac{4}{\epsilon} \rceil}$ new vertices. We connect c_j to d_j through a path traversing all these vertices (thus this is a path from c_j to d_j with $2n^{\lceil \frac{4}{\epsilon} \rceil} + 2$ vertices). We set the cost of all the internal edges of these paths for both time-steps to 0.

For all $i \in \{1, \dots, n\}, j \in \{1, \dots, m\}$ we do the following: if $x_i \in q_j$ we set the cost of the edge (a_i, c_j) to 0 in time step 1; if $y_i \in q_j$ we set the cost of the edge (a_i, c_j) to 0 in time step 2; if $z_i \in q_j$ we set the cost of the edge (b_i, d_j) to 0 in both time steps. All other edge costs are set to ∞ (or some other sufficiently large value). This completes the construction. Observe that the new graph has $5n + 3m + 2mn^{\lceil \frac{4}{\epsilon} \rceil}$ vertices, so at most $C \cdot n^{\frac{4}{\epsilon}+4}$ (for some constant C) since $m \leq n^3$. Note also that the new graph is bipartite because the paths that we added from c_j to d_j have odd lengths, hence the bipartition $(A \cup D, B \cup C)$ can be extended to a bipartition of the whole graph.

Suppose that the original instance has a set $Q' \subseteq Q$ such that $|Q'| = n$ and no element of $X \cup Y \cup Z$ appears in two elements of Q'. We obtain a multistage matching as follows: For each $q_j \in Q'$ such that $q_e = (x_{i_1}, y_{i_2}, z_{i_3})$ we use the edge (a_{i_1}, c_j) in step 1, the edge (a_{i_2}, c_j) in step 2, and the edge (b_{i_3}, d_j) in both time steps. Note that this fully specifies how the vertices of $A \cup B$ are matched. We now complete the matching by selecting a set of edges from the paths connecting each c_j to d_j: if $q_j \in Q'$, then both c_j, d_j have been matched to $A \cup B$ in both time steps, and we select in both time steps the unique perfect matching of the path connecting them; if $q_j \notin Q'$, then neither c_j, d_j is matched to $A \cup B$ in either time step, so we select the perfect matching on the path from c_j to d_j, including these two vertices. Observe that the cost of all edges we use is 0, while we only change at most n edges from one time step to the other, hence the total transition cost is at most nM.

Suppose that the original instance does not have a solution and consider any multistage matching in the new instance. We will show that it must make at least $n^{\frac{4}{\epsilon}}$ changes from one time step to the other. We will say that $q_j \in Q$ is selected in time step 1, if in that time step c_j is matched to an element of A. If q_j is selected in time step 1, then d_j is matched to an element of B in that time step, otherwise it would be impossible to have a perfect matching on the path connecting c_j to d_j. If some q_j is selected in time step 1, but not in time step 2, then the solution must change all internal edges on the perfect matching on the path from c_j to d_j, hence it makes at least $n^{\frac{4}{\epsilon}}$ changes, and we are done. What remains therefore to show is that if the solution maintains the set of selected q_j in the two time steps, then we can construct a solution to the original instance. Indeed, since all of $A \cup B$ is matched, we have n selected q_j's. Each element of $C \cup D$ has at most one edge connecting it to $A \cup B$ in each step, hence if it is selected this edge must be used. But if we select q_{j_1}, q_{j_2} that overlap, then two selected elements will have a common neighbor in $A \cup B$ and will therefore not be matched, contradiction.

Since the new graph has N vertices with $n^{\frac{1}{\epsilon}} \leq N \leq Cn^{\frac{4}{\epsilon}+4}$ vertices, it is NP-hard to distinguish if the optimal is at most $nM \leq N^{\epsilon}M$ or at least $n^{\frac{4}{\epsilon}}M \geq N^{1-\epsilon}M/C$. ◀

3 Metric-Min-MPM

We consider in this section that c_t obeys the triangle inequality: $c_t(u,v)+c_t(v,w) \geq c_t(u,w)$. In particular, the graph is complete. As seen before, the problem is hard to approximate even if there are only 2 time steps with general costs. We show here that while the problem is APX-hard in the metric case even with only 2 time steps (Section 3.1), it admits a 3-approximation algorithm in this case (2 time steps), see Section 3.2. We then extend this last result to the case of 3 time steps in Section 3.3.

3.1 APX-hardness for 2 time steps

In the case of 2 time steps the following result is proved.

▶ **Theorem 3.** Metric-Min-MPM *is APX-hard, even if the input has $T = 2$ time steps.*

Proof. We give a gap-preserving reduction from Max 3DM. We are given an instance of Max 3DM which consists of three sets X, Y, Z, with $|X| = |Y| = |Z| = n$, a set Q of elements of $X \times Y \times Z$, with $|Q| = m$, and an integer k. We are asked if there exists a subset of k pair-wise disjoint elements of Q. We assume that n, m and k are even (if not simply make two independent copies of the initial instance). This problem is APX-hard even if the occurence of each element is bounded above by a constant $C = 3$ [12]. Note that in this case the optimum value is at least $m/7$ (greedy algorithm; at most 6 incompatible triplets are removed when a triplet is chosen). So m, n and k are linearly related ($3n \geq m \geq k \geq m/7 \geq n/21$).

We construct an instance of Metric-Min-MPM as follows: first, we create five sets of vertices X, Y, Z, G, D with $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_n\}$, $Z = \{z_1, \dots, z_n\}$, $G = \{g_1, \dots, g_m\}$ and $D = \{d_1, \dots, d_m\}$.

The graph is complete, and we set the following costs:

- At time step 1, Z is seen as a single point very far from the rest of the graph: (z_i, z_j) has cost 0 for $z_i, z_j \in Z$, and (z_i, v) has infinite cost for $z_i \in Z$, $u \notin Z$.
- The same is done for X at time 2.
- The m edges (g_i, d_i) have cost 1 at both time steps.
- For each triplet $q_i = (x_j, y_p, z_s)$: at time 1 edges (x_j, g_i) and (d_i, y_p) have cost a (a is a sufficiently large constant, to be specified later), and, for the triangle inequality to hold, (x_j, d_i) and (g_i, y_p) have cost $a+1$. Similarly at time 2: (z_s, g_i) and (d_i, y_p) have cost a, and, for the triangle inequality to hold, (z_s, d_i) and (g_i, y_p) have cost $a+1$.

All non yet defined costs are equal to $2a$. The transition cost is $M = 1$. Figure 1 gives an illustration of the construction.

Note that the triangle inequality holds in both time steps.

We show that (1) if there is a 3DM of size k then there exists a solution of Metric-Min-MPM whose total cost is at most $2m + 4an - k/2$, and (2) conversely from a solution of the multistage problem of total cost z we can construct a 3DM of size at least $2(2m + 4an - z)$. This proves APX-hardness since a is a constant, and m, n and k are linearly related.

Let us first prove (1), and suppose that we have a 3DM of size k, say (for ease of notation) $q_1, \dots, q_k$ where $q_i = (x_i, y_i, z_i)$. Then we define a solution S of the multistage matching as follows:

- We take the $(m-k)$ edges of triplets (g_j, d_j) *not* in the 3DM, at both time steps 1 and 2;
- For $q_i, 1 \leq i \leq k$: we take edges (x_i, g_i) at time 1, (z_i, g_i) at time 2, and (y_i, d_i) at time 1 and 2.
- We match together the $(n-k)$ remaining vertices of Y, choosing the same $\frac{n-k}{2}$ edges at both time steps.

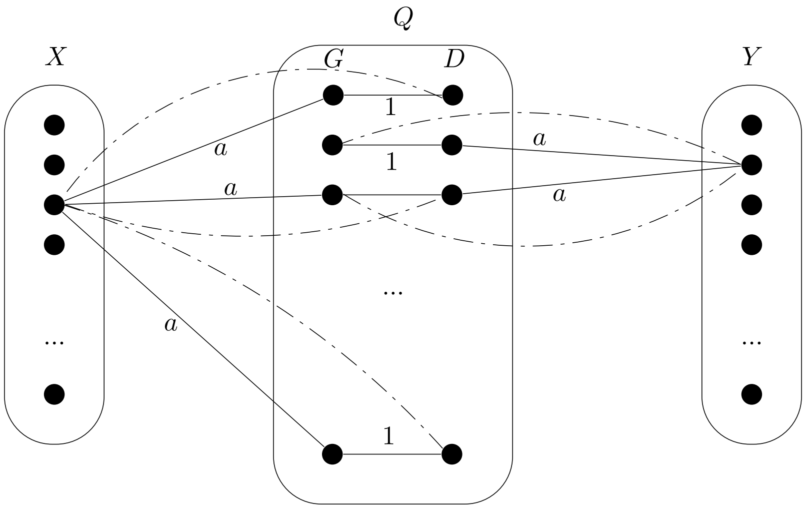

Figure 1 An illustration of the reduction at time $t = 1$, without representing Z - the construction is symmetric for time step $t = 2$. The third element of X is in the first, third and last triplet of Q. The second element of Y is in the second and third triplet. The dashed edges have costs $a + 1$. Not represented edges have cost $2a$..

- We match together the $(n - k)$ remaining vertices of X at time 1. At time 2 we keep these $\frac{n-k}{2}$ edges and match the remaining k vertices of X together.
- We do the same for Z.

We get a solution (M_1, M_2) whose costs are:

- At time 1, the matching cost is $(m - k) + 2ak + 2a\frac{n-k}{2} + 2a\frac{n-k}{2} = m + 2an - k$;
- The matching cost at time 2 is the same.
- The number of modifications is $3k/2$: k edges (x_i, g_i) become (z_i, g_i), and $k/2$ edges in Z disappear at time 1 ($k/2$ edges appear in X at time 2).

In all, (M_1, M_2) has cost $2m + 4an - k/2$.

Conversely, suppose that we have a solution (M_1^0, M_2^0) of total cost z for the instance of Metric-Min-MPM. We first structure this solution using local modifications, and then show how to derive a matching from it.

- *Replacement 1.* First, suppose that M_1^0 takes (at time 1) an edge (x_j, g_i) of cost $2a$ - so x_j is not in the i-th triplet q_i of Q. Then d_i is matched with a vertex v with an edge of cost at least a. By replacing (at time 1) (x_j, g_i) and (d_i, v) by (x_j, v) and (g_i, d_i) we get a matching cost for these two edges at most $2a + 1$ instead of (at least) $3a$. Even considering that the transition cost may have increased by two, this replacement does not increase the cost of the solution for $a \geq 3$. The same argument applies for an edge (x_j, d_i) (time step 1), an edge (y_j, d_i) or (y_j, g_i) (time step 1 or 2) and for an edge (z_j, g_i) or (z_j, d_i) in M_2^0.
- *Replacement 2.* Now, suppose that M_1^0 takes an edge of cost $2a$ in $G \cup D$, say (g_i, g_j) with $i \neq j$ (the very same argument works for the 2 other cases (g_i, d_j) and (d_i, d_j)). Let v and w be the neighbors of d_i and d_j in M_1^0. By replacing the three edges (g_i, g_j), (d_i, v) and (d_j, w) by (g_i, d_i), (g_j, d_j) and (v, w), we get a matching cost at most $(2a + 2)$ instead of (at least) $4a$. Even considering that the transition cost may have increased by three, this replacement does not increase the cost of the solution for $a \geq 5/2$. The same holds for M_2^0.

- *Replacement 3.* Last, suppose that edges (y_j, g_i) and (y_s, d_i) are both taken at time 1 and 2. This costs $2(a + a + 1) = 4a + 2$. Then we can take instead edges (g_i, d_i) and (y_j, y_s) at both time steps, with the same cost $2 + 2(2a) = 4a + 2$.

In this way, we transform (M_1^0, M_2^0) into a solution (M_1, M_2) of cost at most z such that:

- No g_i (and no d_i) is matched using an edge of cost $2a$ (replacements 1 and 2).
- g_i and d_i cannot be both matched to the same vertices at time 1 and 2, unless they are matched together (replacement 3).

We now show how to find a 3DM from this solution (M_1, M_2). Let:

- N_x and N_z be respectively the number of edges in $X \times (G \cup D)$ at time 1 and in $Z \times (G \cup D)$ at time 2.
- N_y^1 and N_y^2 be respectively the number of edges in $Y \times (G \cup D)$ at time 1 and time 2, among which λ_1 (resp., λ_2) are of cost $a + 1$.
- N_y be the number of edges in $Y \times (G \cup D)$ that are taken at both times 1 and 2.

At time 1, besides these $N_x + N_y^1$ edges and the $n/2$ edges of cost 0 (vertices of Z), the other edges of (M_1, M_2) have cost either 1 (edges (g_i, d_i)) or $2a$. Since $N_x + N_y^1$ vertices in $G \cup D$ are already matched at time 1, there are at most $\frac{2m - N_x - N_y^1}{2}$ edges of cost 1 at time 1. Similarly, there are at most $\frac{2m - N_z - N_y^2}{2}$ edges of cost 1 at time 2.

Then, computing the matching cost of (M_1, M_2) we have

$$\begin{aligned} c(M_1, M_2) \quad &\geq \quad a\left(N_x + N_z + N_y^1 + N_y^2\right) + \lambda_1 + \lambda_2 + \frac{4m - N_x - N_z - N_y^1 - N_y^2}{2} \\ &\quad +2a\left(\frac{n - N_x + n - N_y^1 + n - N_z + n - N_y^2}{2}\right) \\ &\geq \quad 2m + 4na + \lambda_1 + \lambda_2 - \frac{N_x + N_z + N_y^1 + N_y^2}{2}. \end{aligned}$$

Now, note that at time 1 at least $N_x + N_y^1 - N_y + \frac{N_z}{2}$ edges disappear, so $D(M_1, M_2) \geq N_x + N_y^1 - N_y + \frac{N_z}{2}$. Similarly, at least $N_z + N_y^2 - N_y + \frac{N_x}{2}$ edges appear at time 2. So $D(M_1, M_2) \geq N_z + N_y^2 - N_y + \frac{N_x}{2}$. Then,

$$D(M_1, M_2) \geq \frac{N_x + N_z + N_y^1 + N_y^2}{2} - N_y + \frac{N_x + N_z}{4}.$$

This gives:

$$z \geq c(M_1, M_2) + D(M_1, M_2) \quad \geq \quad 2m + 4na + \lambda_1 + \lambda_2 - N_y + \frac{N_x + N_z}{4}.$$

Now, consider the set of indices i such that edge (y_j, d_i) is taken at both time steps, or edge (y_j, g_i) is taken at both time steps. Since, thanks to the preprocessing, for a given i this cannot concern both d_i or g_i, we know that there are exactly N_y such indices (edges). Since there are $\lambda_1 + \lambda_2$ edges of cost $a + 1$ between Y and $G \cup D$, among these N_y indices at least $N_y - (\lambda_1 + \lambda_2)$ are such that: (1) edge (d_i, y_j) is used at both time steps (2) an edge (x_s, g_i) of cost a is used at time 1 (since no edge of cost $2a$ is used for vertices in G) and (3) an edge (z_p, g_i) of cost a is used at time 2.

In other words these at least $N_y - (\lambda_1 + \lambda_2)$ indices correspond to triplets of a 3DM. So we have a 3DM of size (at least) $k = N_y - (\lambda_1 + \lambda_2)$. Then, $N_x \geq N_y - (\lambda_1 + \lambda_2) = k$ and similarly $N_z \geq k$, so $\frac{N_z + N_x}{4} \geq \frac{k}{2}$. All together, we get

$$z \geq 2m + 4an - k + \frac{k}{2} = 2m + 4an - \frac{k}{2}. \qquad ◀$$

3.2 A 3-approximation algorithm for 2 time steps

We now devise an approximation algorithm. Informally, this algorithm first guesses the number k of edges that an optimal solution keeps between steps 1 and 2. Then it computes a set of k edges with low matching cost that it maintains between time 1 and 2. Finally, it completes this set of k edges into two perfect matchings, in such a way that, using the triangle inequality, the matching cost does not increase too much.

Formally, the algorithm `Metric2` runs the following procedure for k from 0 to $n/2$.

1. Let G_{1+2} be the graph where the edge costs are $c(u,v) = c_1(u,v) + c_2(u,v)$. Compute a minimum cost matching M^k of size exactly k in G_{1+2}.
2. Compute a minimum cost perfect matching M_1 in G_1, and a minimum cost perfect matching M_2 in G_2.
3. Consider the symmetric difference of the two matchings M^k and M_1 in G_1. This is a (vertex disjoint) set of paths $P_1, \ldots, P_p$ and cycles. Define M_1^k as M^k plus the p edges linking the first vertex and last vertex of each path P_j.
4. Do the same to get M_2^k.
5. Consider $S^k = (M_1^k, M_2^k)$.

`Metric2` outputs the best solution S^k.

▶ **Theorem 4.** *`Metric2` is a (polytime) 3-approximation algorithm for* METRIC-MIN-MPM *when $T = 2$.*

Proof. We first prove that S^k is a feasible solution, i.e., M_i^k is a perfect matching of G_i. Since M_i is a perfect matching, in all paths P_j the first and last edges belong to M_i. Hence the first and last vertices are not covered by M^k, so M_1^k is a matching. Every other vertex is covered by M^k, so the matching is perfect.

Now, let us prove the claimed approximation ratio. Let us denote $S^* = (M_1^*, M_2^*)$ be an optimal solution, and consider S^k where $k = |M_1^* \cap M_2^*|$.

Since at least M^k is common between M_1^k and M_2^k, at least k edges are maintained between time 1 and 2 in S^k, as in S^*. So:

$$D(S^k) \leq D(S^*). \tag{1}$$

Now, let us prove that:

$$c_1(M_1^k) + c_2(M_2^k) \leq 3c_1(M_1^*) + 3c_2(M_2^*). \tag{2}$$

Thanks to the triangle inequality, in a path $P = (v_0, v_1, \ldots, v_t)$, $c_i(v_0, v_t) \leq \sum_j c_i(v_j, v_{j+1})$: when adding edges (v_0, v_t) we add in total at most the total length of the paths, hence at most $c_i(M_i) + c_i(M^k)$. So $c_i(M_i^k) \leq c_i(M_i) + 2c_i(M^k)$. Using that $c_i(M_i) \leq c_i(M_i^*)$, we get:

$$c_1(M_1^k) + c_2(M_2^k) \leq c_1(M_1^*) + c_2(M_2^*) + 2(c_1(M^k) + c_2(M^k)).$$

By optimality of M^k and since S^* has k common edges between times 1 and 2, these k common edges induce a cost in S^* at least $c_1(M^k) + c_2(M^k)$. Then:

$$c_1(M_1^k) + c_2(M_2^k) \leq c_1(M_1^*) + c_2(M_2^*) + 2(c_1(M_1^*) + c_2(M_2^*))$$

and Equation 2 follows. From Equations 1 and 2 we derive:

$$c(S) + D(S) \leq 3c(S^*) + D(S^*).$$

The result immediately follows. ◀

3.3 A 3-approximation algorithm for 3 time steps

We now extend the previous result to the case of $T = 3$. As previously, if an optimal solution preserves in total k edges (operates in total $n-k$ modifications between time steps 1 and 2, and 2 and 3) we would like to first compute a set of k 'preserved' edges inducing a low cost, and then to complete this set as perfect matchings in each of the time steps. Now things get more complex since an edge can be preserved between steps 1 and 2, between steps 2 and 3, or during the whole process. It seems hard to mimic an optimal solution on these 3 types of edges (while inducing a low matching cost), but this difficulty can be overcome as follows.

Let G be the graph with edge cost $w = \min\{c_1 + c_2 + c_3, c_1 + c_2 + M, c_2 + c_3 + M\}$. If the minimum is $c_1 + c_2 + c_3$ (resp., $c_1 + c_2 + M$, $c_2 + c_3 + M$) we say that the edge is of type 1 (resp., 2, 3). Intuitively, edges of type 1 will be taken in steps 1, 2 and 3, edges of type 2 (resp., 3) will be taken in steps 1 and 2 (resp., 2 and 3). We present a 3-approximation algorithm `Metric3`. It runs the following procedure for k from 0 to $n/2$.

1. Compute a minimum cost matching M^k of size exactly k in G. Denote M_1^k the set of edges of M^k of type 1 or 2, $M_2^k = M^k$ and M_3^k the set of edges of M^k of type 1 or 3.
2. Compute a minimum cost perfect matching M_i in G_i, $i = 1, 2, 3$.
3. Consider the symmetric difference of the two matchings M_i^k and M_i in G_i. This is a (vertex disjoint) set of paths $P_1, \ldots, P_p$ and cycles. Define $M_i'^k$ as the set of p edges linking the first vertex and last vertex of each path P_j.
4. Consider $S^k = (M_1^k \cup M_1'^k, M_2^k \cup M_2'^k, M_3^k \cup M_3'^k)$.

Then `Metric3` outputs the best solution S^k.

▶ **Theorem 5.** *`Metric3` is a (polytime) 3-approximation algorithm for* Metric-Min-MPM *when $T = 3$.*

Proof. We first note that, as in the case for $T = 2$ time steps, $M_i^k \cup M_i'^k$ is a perfect matching of G_i, so S^k is a feasible solution.

Now let us deal with the approximation ratio. Let $S^* = (M_1^*, M_2^*, M_3^*)$ be an optimal solution. Let us consider the set $H = (M_1^* \cap M_2^*) \cup (M_2^* \cap M_3^*)$ of edges in S^* that are in (at least) two consecutive steps. Note that H is a matching (it is included in M_2^*). Consider S^k where $k = |H|$. We now prove the following result:

▶ **Lemma 6.** $D(S^k) + \sum_i c_i(M_i^k) \leq D(S^*) + c(S^*)$.

Proof. To prove this, let $k_1 = |M_1^* \cap M_2^* \cap M_3^*|$ be the number of edges in S^* that are taken at each of the 3 time steps. Hence, $k - k_1$ edges are taken at (only) 2 consecutive time steps. So there are $(n/2 + n/2 - 2k_1 - (k - k_1))$ modifications in total, and:

$$D(S^*) = M(n - k - k_1). \tag{3}$$

Recall that in G, $w = min\{c_1 + c_2 + c_3, c_1 + c_2 + M, c_2 + c_3 + M\}$. k_1 edges of H are present on the 3 time steps (matching cost $c_1 + c_2 + c_3$), while $k - k_1$ are present in two consecutive time steps (matching cost $c_1 + c_2$ or $c_2 + c_3$).

$$w(H) \leq c(S^*) + M(k - k_1). \tag{4}$$

Similarly, let λ_1 be the number of edges of type 1 in M^k. There are $(k - \lambda_1)$ edges of type 2 or 3, hence

$$w(M^k) = \sum_i c_i(M_i^k) + M(k - \lambda_1). \tag{5}$$

Indeed, in G cost c_1 applies to edges of type 1 and 2 ($c_1(M_1^k)$), cost c_2 applies to all edges of M^k ($c_2(M_2^k)$), cost c_3 applies to edges of type 1 and 3 ($c_3(M_3^k)$), and cost M to the $(k - \lambda_1)$ edges of type 2 and 3.

Also, the number of preserved edges in S^k is at least $k + \lambda_1$, so:

$$D(S^k) \leq M(n - k - \lambda_1). \tag{6}$$

Since H is a matching, in G we have $w(H) \geq w(M^k)$. This gives using Equations 4 and 5:

$$\sum_i c_i(M_i^k) + M(k - \lambda_1) \leq c(S^*) + M(k - k_1)$$

so $\sum_i c_i(M_i^k) \leq c(S^*) + M(\lambda_1 - k_1)$. Then using Equations 3 and 6 we get:

$$\sum_i c_i(M_i^k) + D(S^k) \leq c(S^*) + M(\lambda_1 - k_1) + M(n - k - \lambda_1) = c(S^*) + M(n - k - k_1)$$

$$= c(S^*) + D(S^*)$$

which concludes the proof of Lemma 6. ◀

Now, by triangle inequality, and the fact that $c_i(M_i) \leq c_i(M_i^*)$, we know that:

$$c_i(M_i'^k) \leq c_i(M_i^*) + c_i(M_i^k). \tag{7}$$

Then, from Lemma 6 and Equation 7 we get:

$$\begin{aligned} c(S^k) + D(S^k) &= \sum_i (c_i(M_i^k) + c_i(M_i'^k)) + D(S^k) \leq \sum_i (2c_i(M_i^k) + c_i(M_i^*)) + D(S^k) \\ &\leq c(S^*) + 2\left(\sum_i c_i(M_i^k) + D(S^k)\right) \leq 3c(S^*) + 2D(S^*). \end{aligned}$$

The result follows. ◀

4 Max-MPM

In the maximization version, we consider that $c_t(e)$ is the *profit* obtained by taking edge e (at time t). Then a solution sequence S has a matching profit $c(S) = \sum_t c_t(M_t)$. We define the transition profit $D(S)$ as $D(S) = \sum_{t \leq T-1} D_t$ where $D_t = M \cdot |M_{t+1} \cap M_t|$ is proportional to the number of edges that *remain* between time t and $t+1$. The goal now is to maximize $c(S) + D(S)$. Recall that in the maximization version we may no longer assume that the graphs are complete.

4.1 APX-hardness for 2 time steps

We first show that Max-MPM, even in the case of 2 time steps is APX-hard.

▶ **Theorem 7.** Max-MPM *is APX-hard even if $T = 2$.*

Proof. As previously, we consider the maximum 3DM problem in the case where the occurrence of each element is bounded by 3, hence the optimal value, the number of triplets and the size of the ground sets are linearly related.

Given three sets X, Y, Z each of size n, and m triplets q_i of $X \times Y \times Z$, we build two graphs G_1 and G_2 with $n' = 2m + 4n$ vertices:

- 4 sets D, E, F, G of size n;
- 2 sets $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_m\}$ of size m.

Vertices of D will represent elements of X, vertices of E and F elements of Y (twice), vertices of G elements of Z. Each triplet q_i is represented by one edge (a_i, b_i) in both graphs. It has cost 0.

If a triplet q_i is (x_j, y_k, z_l) then:

- In G_1 we put edges (d_j, a_i) and (b_i, e_k), both with cost M';
- In G_2 we put edges (f_k, a_i) and (b_i, z_l), both with cost M'.

Note that vertices in F, G have degree 0 in G_1, vertices in D, E have degree 0 in G_2.

We fix $M' = \frac{M+1}{4}$, and $M \geq 3$.

Let us show that there is a 3DM of size (at least) k if and only if there is a solution of profit at least $Mm + k$.

Suppose first that there is a set S of k independent triplets. Then we build matchings (M_1, M_2) as follows:

- if q_i is not in S, we take (a_i, b_i) both in M_1 and M_2. This gives transition profit $M(m-k)$.
- if $q_i = (x_j, y_k, z_l)$ is in S, then we take in M_1 the two edges (d_j, a_i) and (b_i, e_k), and in M_2 the two edges (f_k, a_i) and (b_i, z_l). This gives a matching profit $4kM'$.

Note that since any element of X, Y, Z is in at most one triplet of S, vertices in D, E, F, G are adjacent to at most one chosen edge. In other words M_1 and M_2 are matchings.

The profit of the solution is $4kM' + M(m-k) = k(M+1) + M(m-k) = Mm + k$.

Suppose now that there is a solution (M_1, M_2) of profit at least $Mm + k$. Suppose first that there is an edge (a_i, b_i) which is in M_1 but not in M_2. Then we get no transition profit for this edge. In M_2 we have taken at most one edge incident to a_i, and one edge incident to b_i, with matching profit at most $2M'$. Since these edges are not in G_1 they cannot give transition profit. So we can put in M_2 the edge (a_i, b_i) and remove the edges incident to a_i and b_i (if any). The profit increases by $M - 2M' = M/2 - 1/2 \geq 0$.

So we can assume that M_1 and M_2 have the same set of edges between A and B. Suppose now that there are two edges (a_i, b_i) and (a_s, b_s) both not in M_1 (equiv. not in M_2) corresponding to two intersecting triplets. Suppose for instance that x_j is in both triplets. This means that in M_1 we cannot take both edges (c_j, a_i) and (c_j, a_s), for instance (c_j, a_s) is not in M_1. Then we can add (a_s, b_s) is M_1 and M_2, and remove the (at most) 3 incident edges. This increases profit by $M - 3M' \geq 0$.

So, the set of edges (a_i, b_i) not in M_1 (or not in M_2) corresponds to a set of independent triplets. Let t the number of such edges. Since M_1 is a matching, besides these edges between A and B, there is at most two edges for each (a_i, b_i) not in M_1. Similarly, there is at most two edges in M_2 for each (a_i, b_i) not in M_2. So the matching profit is at most $4tM'$, and the transition profit is $M(m-t)$. The profit is $M(m-t) + 4tM' = Mm + t \geq Mm + k$. So $t \geq k$. ◀

4.2 Constant factor approximation algorithms

▶ **Theorem 8.** Max-MPM *is 1/2-approximable. If $T = 2$ it is 2/3-approximable, if $T = 3$ it is* 3/5*-approximable.*

Proof. Note that if the graphs are assumed to be complete (bipartite complete) then the ratio 1/2 is easily achievable. Indeed, consider two solutions:

- The first one S_1 consisting of the same perfect matching M_0 at all time steps;
- The second one S_2 consisting of a matching $\hat{M}_t$ of maximum profit on G_t for each t.

Output the best one.

Let $S^* = (M_1^*, \ldots, M_T^*)$ be an optimal solution. Clearly the profit of S_1 is at least the transition profit $D(S^*)$ of S^*. Also, $c(M_i^*) \leq c(\hat{M}_i)$ so the matching profit of S^* is at most the one of S_2. The ratio 1/2 follows.

If the graphs are not assumed to be complete things get harder since one cannot trivially optimize the transition profit by keeping a perfect matching along the multistage process.

Let us consider three consecutive time steps $t-1, t, t+1$. Let us consider the graph G'_t which is the same as G_t up to the profit on edges, which is now $c'_t(e)$ where:

1. $c'_t(e) = c_t(e) + 2M$ if e is in G_{t-1} and G_{t+1};
2. otherwise, $c'_t(e) = c_t(e) + M$ if e is in G_{t-1} or G_{t+1};
3. otherwise $c'_t(e) = c_t(e)$.

Let us consider a matching M'_t of maximum profit in G'_t.

▶ **Lemma 9.** *$c'_t(M'_t) \geq D_{t-1}(S^*) + c_t(M_t^*) + D_t(S^*)$.*

Proof. Let us consider the profit of M_t^* on G'_t. Since the set of edges preserved from time $t-1$ to time t is included in M_t^*, the profit $D_{t-1}(S^*)$ appears in the profit of M_t^* on G'_t ($+M$ on each common edges between the two consecutive graphs). This is also the case for $D_t(S^*)$, for the same reason. Of course, the profit $c_t(e)$ appears as well. Since M'_t is of maximum profit, the Lemma follows. ◀

Because of Lemma 9, choosing the matching M'_t at time steps $t-1$, t and $t+1$ in a solution generates a profit at least $D_{t-1}(S^*) + c_t(M_t^*) + D_t(S^*)$.

Note that, with similar arguments, if two times steps $t, t+1$ are involved, we can compute a matching H_i that we take at time steps $t, t+1$ generating a profit at least $c_t(M_t^*) + D_t(S^*)$. Symmetrically, we can compute a matching H'_i that we take at time steps $t, t+1$ generating a profit at least $c_{t+1}(M_t^*) + D_t(S^*)$.

Now we consider the following 2 solutions:

- S_1 consists of choosing H_1 at steps $1, 2$, H_3 at step $3, 4, \ldots$. If T is even then we are done, otherwise we take an optimal matching $\hat{M}_T$ at step T.
- S_2 consisting of choosing an optimal matching $\hat{M}_1$ at step 1, then H_2 at steps $2, 3$, H_4 at steps $4, 5, \ldots$. If T is even we take an optimal matching $\hat{M}_T$ at step T.

Output the best of these two solutions. Then: S_1 covers the transition profit of an optimal solution D_t for t odd, plus the matching profits for t odd. S_2 covers the transition profit of an optimal solution D_t for t even, plus the matching profits for t even. The ratio 1/2 follows.

Improvement for $T = 3$. The previous solutions S_1 and S_2 have profit (respectively) at least $c_1(S^*) + D_1(S^*) + c_3(S^*)$ and $c_1(S^*) + D_2(S^*) + c_2(S^*)$. S_3 takes $\hat{M}_1$ at step 1 and H'_2 at time steps 2 and 3, with profit at least $c_1(S^*) + D_2(S^*) + c_3(S^*)$; S_4 takes H'_1 at steps 1 and 2, and $\hat{M}_3$ at step 3, with profit at least $D_1(S^*) + c_2(S^*) + c_3(S^*)$. S_5 uses M'_2 at the 3 steps with profit at least $D_1(S^*) + c_2(S^*) + D_2(S^*)$ (thanks to Lemma 9). Take the best of these 5 solutions, and the ratio follows.

Improvement for $T = 2$. Simply take 3 solutions: S_1 is defined as previously, with profit at least $c_1(S^*) + D_1(S^*)$. S_2 takes H'_1 at both steps with profit at least $D_1(S^*) + c_2(S^*)$. S_3 consists of one optimal matching at step 1, and an optimal matching at step 2, with profit at least $c_1(S^*) + c_2(S^*)$. The ratio 2/3 follows. ◀

5 Concluding remarks

Following the results of Section 3, we leave as an open question the existence of a constant factor approximation algorithm for the metric case for a number of time steps bigger than 3. Also, we considered here an off-line version of the problem where the whole set of instances is known in advance. It would be worth investigating the on-line case where data are not known in advance.

References

1 Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. *ACM Trans. Algorithms*, 13(2):21:1–21:20, 2017.

2 Barbara M. Anthony and Anupam Gupta. Infrastructure leasing problems. In *IPCO*, volume 4513 of *Lecture Notes in Computer Science*, pages 424–438. Springer, 2007.

3 Nicolas K. Blanchard and Nicolas Schabanel. Dynamic sum-radii clustering. In *WALCOM*, volume 10167 of *Lecture Notes in Computer Science*, pages 30–41. Springer, 2017.

4 Niv Buchbinder, Shahar Chen, and Joseph Naor. Competitive analysis via regularization. In *SODA*, pages 436–444. SIAM, 2014.

5 Niv Buchbinder, Shahar Chen, Joseph Naor, and Ohad Shamir. Unified algorithms for online learning and competitive analysis. *Math. Oper. Res.*, 41(2):612–625, 2016.

6 Edith Cohen, Graham Cormode, Nick G. Duffield, and Carsten Lund. On the tradeoff between stability and fit. *ACM Trans. Algorithms*, 13(1):7:1–7:24, 2016.

7 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *ICALP (2)*, volume 8573 of *Lecture Notes in Computer Science*, pages 459–470. Springer, 2014.

8 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete problems. In *STOC*, pages 47–63. ACM, 1974.

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

10 Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: Maintaining a constant-competitive steiner tree online. *SIAM J. Comput.*, 45(1):1–28, 2016.

11 Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 563–575. Springer, 2014.

12 Viggo Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991.

13 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The power of recourse for online MST and TSP. *SIAM J. Comput.*, 45(3):859–880, 2016.

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

15 Baruch Schieber, Hadas Shachnai, Gal Tamir, and Tami Tamir. A theory and algorithms for combinatorial reoptimization. *Algorithmica*, 80(2):576–607, 2018.