Budgeted Matching and Budgeted Matroid Intersection via the Gasoline Puzzle*

André Berger † Vincenzo Bonifaci ‡ Fabrizio Grandoni §
Guido Schäfer ¶

Abstract

Many polynomial-time solvable combinatorial optimization problems become NP-hard if an additional complicating constraint is added to restrict the set of feasible solutions. In this paper, we consider two such problems, namely maximum-weight matching and maximum-weight matroid intersection with one additional budget constraint. We present the first polynomial-time approximation schemes for these problems. Similarly to other approaches for related problems, our schemes compute two solutions to the Lagrangian relaxation of the problem and patch them together to obtain a near-optimal solution. However, due to the richer combinatorial structure of the problems considered here, standard patching techniques do not apply. To circumvent this problem, we crucially exploit the adjacency relations on the solution polytope and, surprisingly, the solution to an old combinatorial puzzle.

Keywords: Matching, Matroid Intersection, Budgeted Optimization, Lagrangian relaxation

1 Introduction

Many combinatorial optimization problems can be formulated as follows. We are given a (finite) set $\mathcal{F}$ of feasible solutions and a weight function $w : \mathcal{F} \to \mathbb{Q}$ that assigns a weight $w(S)$ to every feasible solution $S \in \mathcal{F}$. An *optimization problem* Π asks for the computation of a feasible solution $S^* \in \mathcal{F}$ of maximum weight opt_Π, i.e.,

$$\mathrm{opt}_\Pi := \text{maximize } w(S) \text{ subject to } S \in \mathcal{F}. \tag{Π}$$

In this paper, we are interested in solving such optimization problems if the set of feasible solutions is further constrained by a single *budget constraint*. More precisely, we are additionally given a non-negative cost function $c : \mathcal{F} \to \mathbb{Q}^+$ that specifies a cost $c(S)$ for every feasible solution $S \in \mathcal{F}$ and a non-negative budget $B \in \mathbb{Q}^+$. The *budgeted optimization problem* $\bar{\Pi}$ of the above problem Π can then be formulated as follows:

$$\mathrm{opt} := \text{maximize } w(S) \text{ subject to } S \in \mathcal{F},\ c(S) \leq B. \tag{$\bar{\Pi}$}$$

*A preliminary version appeared in *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 5035, Springer, Berlin, 2008, pp. 273–287. This work was done while the first three authors were postdoctoral fellows at TU Berlin. Research supported by the European Regional Development Fund (ERDF).

†Department of Quantitative Economics, Maastricht University, The Netherlands. Email: a.berger@maastrichtuniversity.nl

‡Department of Electrical Engineering, University of L'Aquila, Italy and Department of Computer and Systems Science, Sapienza University of Rome, Italy. Email: bonifaci@dis.uniroma1.it. Work partially supported by the Future and Emerging Technologies Unit of EC (IST priority - 6th FP), under contract no. FP6-021235-2 (project ARRIVAL).

§Department of Computer Science, Systems and Production, University of Rome Tor Vergata, Italy. Email: grandoni@disp.uniroma2.it. Research supported by MIUR under project MAINSTREAM.

¶Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands and Department of Econometrics and Operations Research, VU University Amsterdam, The Netherlands. Email: g.schaefer@cwi.nl.

Even if the original optimization problem Π is polynomial-time solvable, adding a budget constraint typically renders the budgeted optimization problem $\bar{\Pi}$ NP-hard. Problems that fall into this class are, for example, the constrained shortest path problem [3], the constrained minimum spanning tree problem [1], and the constrained minimum arborescence problem [12].

We study the budgeted version of two fundamental optimization problems, namely the maximum-weight matching problem and the maximum-weight matroid intersection problem.

In the *budgeted matching problem*, we are given an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}$ and edge costs $c : E \to \mathbb{Q}^+$, and a budget $B \in \mathbb{Q}^+$. The set $\mathcal{F}$ of feasible solutions corresponds to the set of all matchings in G. (Note that we do not require that the matchings in $\mathcal{F}$ are perfect.) Define the weight of a matching M as the total weight of all edges in M, i.e., $w(M) := \sum_{e \in M} w(e)$. Similarly, the cost of M is defined as $c(M) := \sum_{e \in M} c(e)$. The goal is to compute a matching $M^* \in \mathcal{F}$ of maximum weight $w(M^*)$ among all matchings M in $\mathcal{F}$ whose cost $c(M)$ is at most B.

In the *budgeted matroid intersection problem*, we are given two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$ on a common ground set of elements E (formal definitions will be given in Section 2). Throughout this paper, we assume that a matroid is given implicitly by an *independence oracle* that determines in $O(1)$ time whether a given set $X \subseteq E$ is an independent set of the matroid or not. Moreover, we are given element weights $w : E \to \mathbb{Q}$, element costs $c : E \to \mathbb{Q}^+$, and a budget $B \in \mathbb{Q}^+$. The set of all feasible solutions $\mathcal{F} := \mathcal{F}_1 \cap \mathcal{F}_2$ is defined by the intersection of $\mathcal{M}_1$ and $\mathcal{M}_2$. The weight of an independent set $X \in \mathcal{F}$ is defined as $w(X) := \sum_{e \in X} w(e)$ and the cost of X is $c(X) := \sum_{e \in X} c(e)$. The goal is to compute a common independent set $X^* \in \mathcal{F}$ of maximum weight $w(X^*)$ among all feasible solutions $X \in \mathcal{F}$ satisfying $c(X) \leq B$. Problems that can be formulated as the intersection of two matroids are, for example, matchings in bipartite graphs, arborescences in directed graphs, spanning forests in undirected graphs, etc.

A special case of both budgeted matching and budgeted matroid intersection is the budgeted matching problem on bipartite graphs. This problem is NP-hard by a simple reduction from the knapsack problem. We remark that without the budget constraint the two problems can be solved in polynomial-time (see, e.g., [33]).

Our Contribution

We give the first polynomial-time approximation schemes (PTAS) for the budgeted matching problem and the budgeted matroid intersection problem. For a given input parameter $\epsilon > 0$, our algorithms compute a $(1-\epsilon)$-approximate solution in time $O(m^{O(1/\epsilon)})$, where m is the number of edges in the graph or the number of elements in the ground set, respectively.

The basic structure of our polynomial-time approximation schemes resembles similar approaches for related budgeted optimization problems [29]. By dualizing the budget constraint of $\bar{\Pi}$ and lifting it into the objective function, we obtain for any $\lambda \geq 0$ the Lagrangian relaxation LR(λ).

$$z(\lambda) := \text{maximize } \big(w(S) + \lambda(B - c(S))\big) \text{ subject to } S \in \mathcal{F}. \qquad (\text{LR}(\lambda))$$

Note that the relaxed problem LR(λ) is equivalent to the optimization problem Π with modified *Lagrangian weights* $w_\lambda(e) := w(e) - \lambda c(e)$ for every $e \in E$. Since the original problem Π is polynomial-time solvable, we can compute the optimal *Lagrangian multiplier* $\lambda^* := \arg\min_{\lambda \geq 0} z(\lambda)$ and two optimal solutions S_1 and S_2 to LR(λ^*) such that $c(S_1) \leq B \leq c(S_2)$. (Details will be given in Section 2.) The idea now is to *patch* S_1 and S_2 together to obtain a feasible solution S for $\bar{\Pi}$ whose weight $w(S)$ is at least $(1-\epsilon)$opt. Our patching consists of two phases: an *exchange phase* and an *augmentation phase*.

Exchange Phase: Consider the polytope induced by the feasible solutions $\mathcal{F}$ to the original problem Π and let F be the face given by the solutions of maximum Lagrangian weight. This face contains both S_1 and S_2. In the first phase, we iteratively replace either S_1 or S_2 with another vertex on F, preserving the invariant $c(S_1) \leq B \leq c(S_2)$, until we end up with two adjacent solutions. Note that both solutions have objective value $z(\lambda^*) \geq$ opt. However, with respect to their original weights,

we can only infer that $w(S_i) = z(\lambda^*) - \lambda^*(B - c(S_i))$, $i \in \{1, 2\}$. That is, we cannot hope to use these solutions directly: S_1 is a feasible solution for $\bar{\Pi}$ but its weight $w(S_1)$ might be arbitrarily far from opt. In contrast, S_2 has weight $w(S_2) \geq$ opt, but is infeasible.

Augmentation Phase: In this phase, we exploit the properties of adjacent solutions in the solution polytope. For matchings it is known that two solutions are adjacent in the matching polytope if and only if their symmetric difference is an alternating cycle or path X. Analogously, two adjacent extreme points in the common basis polytope of two matroids can be characterized by a proper alternating cycle X in the corresponding exchangeability graph [8, 14]. The idea is to patch S_1 according to a proper subpath X' of X. This subpath X' guarantees that the Lagrangian weight of S_1 does not decrease too much, while at the same time the gap between the budget and the cost of S_1 (and hence also the gap between $w(S_1)$ and $z(\lambda^*)$) is reduced. This way we obtain a feasible solution S whose weight differs from opt by at most the weight of two edges (elements).

Of course, constructing such a solution S alone is not sufficient to obtain a PTAS: the maximum weight of an edge (element) might be comparable to the weight of an optimum solution. However, this problem can be easily overcome by guessing the edges (elements) of largest weight in the optimum solution in a preliminary step.

Surprisingly, the key ingredient that enables us to prove that there always exists a good patching subpath stems from an old combinatorial puzzle which we quote from the book by Lovász [19, Problem 3.21].

> "Along a speed track there are some gas-stations. The total amount of gasoline available in them is equal to what our car (which has a very large tank) needs for going around the track. Prove that there is a gas-station such that if we start there with an empty tank, we shall be able to go around the track without running out of gasoline."

Related Work

For the budgeted matching problem there is an optimal algorithm if the costs are uniform. This problem is equivalent to finding a maximum-weight matching that consists of at most B edges, which can be solved by a reduction to perfect matching.

Naor et al. [25] proposed a fully polynomial-time approximation scheme (FPTAS) for a rather general class of problems, which contains the budgeted matching problem considered here as a special case. However, personal communication [24] revealed that unfortunately the stated result [25, Theorem 2.2] is incorrect.

Papadimitriou and Yannakakis [28] provide a very general and powerful tool to design approximation algorithms for problems with a constant number of objectives, based on the construction of ϵ-approximate Pareto curves. In order to construct such approximate Pareto curves efficiently, a sufficient condition is the existence of a pseudo-polynomial-time algorithm for the *exact* version of the problem considered. The task in the exact version of the problem is to return a feasible solution of exactly some prespecified value. For example, the existence of such a pseudo-polynomial-time algorithm for the minimum spanning tree problem [2] implies a polynomial-time algorithm which returns a $(1 + \epsilon)$-approximate solution violating every budget constraints by at most a factor of $(1+\epsilon)$ for the corresponding multi-objective version. In the case of exact (perfect) matching, there is a pseudo-polynomial-time randomized (Monte Carlo) algorithm [23]. A similar result holds for exact matroid intersection [4]. These results imply that there are randomized fully polynomial-time approximation schemes for the budgeted matching and budgeted matroid intersection problems that compute an (expected) $(1 - \epsilon)$-approximate solution violating the budget constraint by a factor of at most $(1 + \epsilon)$. Similar results can be obtained for a fixed number of budget constraints. We remark that derandomizing the mentioned exact algorithms is a long-standing, important open problem and thus it seems hard to find a deterministic FPTAS (or even PTAS) for the problems considered here using this approach. In contrast, our polynomial-time approximation schemes are deterministic and do not violate the budget constraint.

Budgeted versions of polynomial-time solvable optimization problems have been studied extensively. The best known ones are probably the constrained shortest path problem and the constrained minimum spanning tree problem. Finding a shortest s,t-path P (with respect to weight) between two vertices s and t in a directed graph with edge weights and edge costs such that the total cost of P is at most B appears as an NP-hard problem already in the book by Garey and Johnson [10]. Similarly, finding a minimum weight spanning tree whose total cost is at most some specified value is NP-hard as well [1].

Goemans and Ravi [29] obtain a PTAS for the constrained minimum spanning tree problem by using an approach which resembles our exchange phase. Starting from two spanning trees obtained from the Lagrangian relaxation, they walk along the optimal face (with respect to the Lagrangian weights) of the spanning tree polytope until they end up with two adjacent solutions S_1 and S_2 with $c(S_1) \leq B \leq c(S_2)$. In this polytope, two spanning trees are adjacent if and only if their symmetric difference consists of just two edges. Therefore, the final solution S_1 is a feasible spanning tree whose weight is away from the optimum by the weight of only one edge. In particular, once two such adjacent solutions have been found there is no need for an additional augmentation phase, which is instead crucial for matchings and matroid intersections. The PTAS by Goemans and Ravi [29] also extends to the problem of finding a minimum-weight basis in a matroid subject to a budget constraint. A bicriteria FPTAS for the constrained minimum spanning tree problem has been found by Hong et al. [13]. However, the question whether there exists an FPTAS for the constrained minimum spanning tree problem is open.

Finding constrained minimum arborescences in directed graphs is NP-hard as well. Guignard and Rosenwein [12] apply Lagrangian relaxation to solve it to optimality (though not in polynomial time). Previous work on budgeted optimization problems also includes results on budgeted scheduling [34] and bicriteria results for several budgeted network design problems [21].

Besides the mentioned technique by Papadimitriou and Yannakakis, not much is known about problems with multiple budget constraints. A problem that might be cast into this framework is the degree-bounded minimum spanning tree problem, where the goal is to find a minimum-weight spanning tree such that the degree of each node v respects a prespecified degree bound B_v. This problem can be formulated as a spanning tree problem with a budget constraint for each node v: The cost is one for all edges incident to v and zero otherwise, and the budget is B_v. A lot of work has been devoted to this problem [5, 6, 9, 11, 15, 16, 30, 31], culminating in the recent algorithm by Singh and Lau [35]. Their algorithm computes a spanning tree that violates the degree bound of every node by at most one (additive) and whose weight is at most the weight of an optimal degree-bounded spanning tree. The mentioned results however do not extend to arbitrary cost functions.

To the best of our knowledge, the Gasoline Lemma was first used algorithmically by Lin and Kernighan [18] in the context of devising an efficient heuristic for the traveling salesman problem.

Organization of the Paper

The paper is structured as follows. In Section 2, we give some prerequisites on matroids and Lagrangian relaxation. We then present the PTAS for the budgeted matching problem in Section 3. The PTAS for the budgeted matroid intersection problem is the subject of Section 4. In Section 5 we discuss some open problems.

2 Preliminaries

2.1 Matroids

Let E be a set of elements and $\mathcal{F} \subseteq 2^E$ be a non-empty set of subsets of E. Then $\mathcal{M} = (E, \mathcal{F})$ is a *matroid* if the following holds:

1. If $X \in \mathcal{F}$ and $Y \subseteq X$, then $Y \in \mathcal{F}$.
2. For every $X, Y \in \mathcal{F}$, $|X| = |Y|$, for every $x \in X$ there is a $y \in Y$ such that $X \setminus \{x\} \cup \{y\} \in \mathcal{F}$.

The elements of $\mathcal{F}$ are called *independent sets*. An independent set X is a *basis* of $\mathcal{M}$ if for every $x \in E \setminus X$, $X \cup \{x\} \notin \mathcal{F}$. We assume that $\mathcal{F}$ is represented implicitly by an oracle: for any given $I \subseteq E$, this oracle determines whether $I \in \mathcal{F}$ or not. In the running time analysis, each query to the oracle is assumed to take constant time. It is not hard to show that matroids have the following properties (see e.g. [33] and references therein).

Lemma 2.1. *For any given matroid $\mathcal{M} = (E, \mathcal{F})$:*

1. *(Deletion) For every $E_0 \subseteq E$, $\mathcal{M} - E_0 := (E', \mathcal{F}')$ is a matroid, where $E' := E \setminus E_0$ and $\mathcal{F}' := \{X \in \mathcal{F} : X \cap E_0 = \emptyset\}$.*
2. *(Contraction) For every $E_0 \in \mathcal{F}$, $\mathcal{M}/E_0 := (E', \mathcal{F}')$ is a matroid, where $E' := E \setminus E_0$ and $\mathcal{F}' := \{X \subseteq E \setminus E_0 : X \cup E_0 \in \mathcal{F}\}$.*
3. *(Truncation) For every $q \in \mathbb{N}$, $\mathcal{M}^q := (E, \mathcal{F}^q)$ is a matroid, where $\mathcal{F}^q := \{X \in \mathcal{F} : |X| \leq q\}$.*
4. *(Extension) For every set D, $D \cap E = \emptyset$, $\mathcal{M} + D := (E', \mathcal{F}')$ is a matroid, where $E' := E \cup D$ and $\mathcal{F}' := \{X \subseteq E \cup D : X \cap E \in \mathcal{F}\}$.*

Observe that an oracle for the original matroid implicitly defines an oracle for all the derived matroids above. Given $X \in \mathcal{F}$ and $Y \subseteq E$, the *exchangeability graph* of $\mathcal{M}$ with respect to X and Y is the bipartite graph $\mathbf{ex}_{\mathcal{M}}(X, Y) := (X \setminus Y, Y \setminus X; H)$ with edge set $H = \{(x, y) : x \in X \setminus Y, y \in Y \setminus X, X \setminus \{x\} \cup \{y\} \in \mathcal{F}\}$.

Lemma 2.2 ([17])**.** *(Exchangeability Lemma) Given $X \in \mathcal{F}$ and $Y \subseteq E$, if the graph $\mathbf{ex}_{\mathcal{M}}(X, Y)$ has a unique perfect matching, then $Y \in \mathcal{F}$.*

The *intersection* of two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$ over the same ground set E is the pair $\mathcal{M} = (E, \mathcal{F}_1 \cap \mathcal{F}_2)$. We remark that the intersection of two matroids might not be a matroid, while every matroid $\mathcal{M} = (E, \mathcal{F})$ is the intersection of itself with the trivial matroid $(E, 2^E)$. Lemma 2.1 can be naturally extended to matroid intersections. For example, for a given matroid intersection $(E, \mathcal{F}_1 \cap \mathcal{F}_2)$, by Lemma 2.1.3 $(E, \mathcal{F}_1^q \cap \mathcal{F}_2^q)$ is still the intersection of two matroids, for any $q \in \mathbb{N}$.

Given two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$, the *common basis polytope* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is the convex hull of the characteristic vectors of the common bases. We say that two common bases $X, Y \in \mathcal{F}_1 \cap \mathcal{F}_2$ are *adjacent* if their characteristic vectors are adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$.

2.2 Lagrangian Relaxation

We briefly review the *Lagrangian relaxation* approach; for a more detailed exposition, the reader is referred to [26]. The Lagrangian relaxation of the budgeted optimization problem $\bar{\Pi}$ is given by:

$$z(\lambda) := \text{maximize } \big(w(S) + \lambda(B - c(S))\big) \text{ subject to } S \in \mathcal{F}. \qquad (\text{LR}(\lambda))$$

For any value of $\lambda \geq 0$, the optimal solution to LR(λ) gives an upper bound on the optimal solution of the budgeted problem, because any feasible solution S satisfies $\sum_{e \in S} c(e) \leq B$. The Lagrangian relaxation problem is to find the best such upper bound, i.e., to determine λ^* such that $z(\lambda^*) = \min_{\lambda \geq 0} z(\lambda)$ (see also Figure 1). This can be done in polynomial time by standard techniques whenever the corresponding basic problem Π (and hence LR(λ) for fixed λ) is solvable in polynomial time [32]. Within the same time bound, one can also compute two solutions S_1 and S_2 satisfying $c(S_1) \leq B \leq c(S_2)$, which are optimal with respect to the Lagrangian weights $w_{\lambda^*}(e) := w(e) - \lambda^* c(e)$, $e \in E$. If Π admits a strongly polynomial-time *combinatorial* algorithm (i.e., an algorithm which only compares and sums weights), one can even compute λ^*, S_1 and S_2 in strongly polynomial time by using Megiddo's parametric search technique [22]. This idea has been used, e.g., by Goemans and Ravi [29] to derive strongly polynomial-time algorithms for the constrained minimum spanning tree problem. Since strongly polynomial-time combinatorial algorithms are known for the weighted matching problem and the weighted matroid intersection problem [33], we can use the same approach here in order to obtain strongly polynomial-time algorithms for the corresponding budgeted versions.

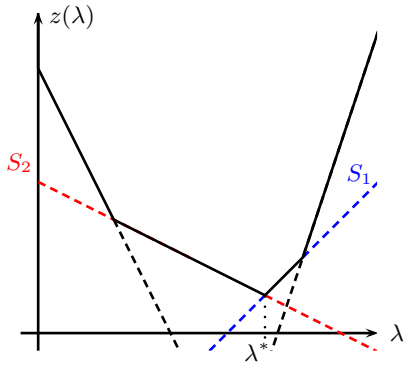

Figure 1: The Lagrangian value $z(\lambda)$ as a function of λ (solid line). Each dashed line represents the Lagrangian value of a specific solution.

2.3 The Gasoline Puzzle

One crucial ingredient in our patching procedure is the solution to the puzzle cited in the introduction. We state it more formally in the following lemma.

Lemma 2.3. *(Gasoline Lemma) Given a sequence of k real numbers $a_0, \ldots, a_{k-1}$ such that $\sum_{j=0}^{k-1} a_j = 0$, there is an index $i \in \{0, \ldots, k-1\}$ such that*

$$\forall h \in \{0, \ldots, k-1\}: \qquad \sum_{j=i}^{i+h} a_j \pmod k \geq 0.$$

Proof. Let $i' \in \{0, \ldots, k-1\}$ be the index for which $\sum_{j=0}^{i'} a_j$ is minimum and set $i := (i'+1) \pmod k$. By the choice of i', we have for every $h \in \{0, \ldots, k-1\}$:

$$\sum_{j=i}^{i+h} a_j \pmod k = \sum_{j=0}^{i+h} a_j \pmod k - \sum_{j=0}^{i'} a_j \pmod k \geq 0. \qquad \square$$

3 A PTAS for the Budgeted Matching Problem

In this section, we present our PTAS for the budgeted matching problem. Suppose we are given a budgeted matching instance $I := (G, w, c, B)$. Let n and m refer to the number of nodes and edges in G, respectively. Moreover, we define $w_{\max} := \max_{e \in E} w(e)$ as the largest edge weight in I. Throughout this section, opt refers to the weight of an optimal solution M^* for I. In order to prove that there exists a PTAS, we proceed in two steps: First we prove that there is an algorithm to compute a feasible solution of weight at least $\text{opt} - 2w_{\max}$. The Gasoline Lemma will play a crucial role in this proof. The claimed PTAS is then obtained by guessing the edges of largest weight in M^* in a preliminary phase and applying the algorithm above.

Lemma 3.1. *There is a polynomial-time algorithm to compute a solution M to the budgeted matching problem of weight $w(M) \geq \text{opt} - 2\, w_{\max}$.*

Proof. As described in Section 2, we first compute the optimal Lagrangian multiplier $\lambda > 0$ and two matchings M_1 and M_2 of maximum Lagrangian weight $w_\lambda(M_1) = w_\lambda(M_2)$ such that $c(M_1) \leq B \leq c(M_2)$. Observe that for $i \in \{1, 2\}$ we have that

$$w_\lambda(M_i) + \lambda\, B \geq w_\lambda(M^*) + \lambda\, B \geq w_\lambda(M^*) + \lambda\, c(M^*) = \text{opt}. \tag{1}$$

We also remark that, by the optimality of M_1 and M_2, $w_\lambda(e) \geq 0$ for all $e \in M_1 \cup M_2$.

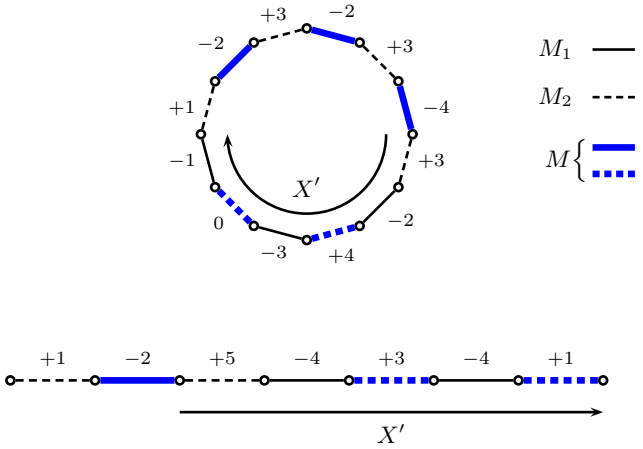

Figure 2: Examples illustrating the construction used in the proof of Lemma 3.1 where the symmetric difference is a cycle (top) or a path (bottom). Each edge x_i is labeled with its value a_i.

We next show how to extract from $M_1 \cup M_2$ a matching M with the desired properties in polynomial time. Intuitively, this will be achieved by exchanging edges along cycles or paths in the symmetric difference of M_1 and M_2. We will distinguish two cases, one in which a whole such cycle or path can be exchanged, and a more complicated case, in which we can only exchange edges along a subpath of a cycle or path in that symmetric difference.

Consider the symmetric difference $M' = M_1 \oplus M_2$. Recall that $M' \subseteq M_1 \cup M_2$ consists of a disjoint union of paths $\mathcal{P}$ and cycles $\mathcal{C}$. We apply the following procedure until eventually $|\mathcal{P} \cup \mathcal{C}| \leq 1$: Take some $X \in \mathcal{P} \cup \mathcal{C}$ and let $A := M_1 \oplus X$. If $c(A) \leq B$ replace M_1 by A. Otherwise replace M_2 by A. Throughout this procedure we maintain the invariant $c(M_1) \leq B \leq c(M_2)$. Observe that in each step, the cardinality of $M_1 \cap M_2$ increases by at least one (in fact by $\min\{|M_2 \cap X|, |M_1 \setminus X|\}$); hence this procedure terminates after at most $O(n)$ steps. Moreover, by the optimality of M_1 and M_2, the Lagrangian weight of the two matchings does not change during the process (i.e., the two matchings remain optimal). To see this note that by the optimality of M_i, $i \in \{1, 2\}$, $w_\lambda(M_i) \geq w_\lambda(M_i \oplus X)$. On the other hand $w_\lambda(M_1) + w_\lambda(M_2) = w_\lambda(M_1 \oplus X) + w_\lambda(M_2 \oplus X)$. It follows that $w_\lambda(A) = w_\lambda(M_1) = w_\lambda(M_2)$.

If at the end of this procedure $c(M_i) = B$ for some $i \in \{1, 2\}$, we are done: M_i is a feasible solution to the budgeted matching problem and

$$w(M_i) = w_\lambda(M_i) + \lambda c(M_i) = w_\lambda(M_i) + \lambda B \geq \text{opt}.$$

Otherwise, $M_1 \oplus M_2$ consists of a unique path or cycle $X = (x_0, x_1, \ldots, x_{k-1}) \subseteq E$ such that $c(M_1 \oplus X) = c(M_2) > B > c(M_1)$.

Note that from the above inequality it also follows that $w(M_1) \geq \text{opt} - \lambda(B - c(M_1))$, i.e., the original weight of an optimal Lagrangian solution that is feasible is close to optimal if its cost is sufficiently close to the budget. The basic idea is to exchange edges along a subpath of X so as to achieve a feasible solution whose cost is close to the budget but still has large Lagrangian weight.

In more detail, consider the sequence

$$a_0 = \delta(x_0)\, w_\lambda(x_0), \quad a_1 = \delta(x_1)\, w_\lambda(x_1), \quad \ldots, \quad a_{k-1} = \delta(x_{k-1})\, w_\lambda(x_{k-1}),$$

where $\delta(x_i) = 1$ if $x_i \in M_2$ and $\delta(x_i) = -1$ otherwise. (Note that, if X is a path, x_0 and x_{k-1} might both belong to either M_1 or M_2). This sequence has total value $\sum_{j=0}^{k-1} a_j = 0$, because of the optimality of M_1 and M_2. By the Gasoline Lemma, there must exist an edge x_i, $i \in \{0, 1, \ldots, k-1\}$, of X such that for any cyclic subsequence $X' = (x_i,\ x_{(i+1) \pmod k},\ \ldots,\ x_{(i+h) \pmod k})$, where $h \in \{0, \ldots, k-1\}$, we have that

$$0 \leq \sum_{j=i}^{i+h} a_j \pmod k = \sum_{e \in X' \cap M_2} w_\lambda(e) - \sum_{e \in X' \cap M_1} w_\lambda(e). \tag{2}$$

Consider the longest such subsequence X' satisfying $c(M_1 \oplus X') \leq B$ (see Figure 2 for an example). Note that X' consists of either one or two alternating paths (the latter case only occurs if X is a path whose first and last edge belong to X'). Let $e_1 = x_i$. Without loss of generality, we can assume $e_1 \in M_2$ (X' might start with one or two edges of M_1 with Lagrangian weight zero, in which case the next edge in M_2 is a feasible starting point of X' as well). Observe that $M_1 \oplus X'$ is not a matching unless X is a path and e_1 its first edge. However, $M := (M_1 \oplus X') \setminus \{e_1\}$ is always a matching. Moreover, $c(M) = c(M_1 \oplus X') - c(e_1) \leq c(M_1 \oplus X') \leq B$. That is, M is a feasible solution to the budgeted matching problem.

It remains to lower bound the weight of M. We have

$$\begin{aligned} w(M_1 \oplus X') &= w_\lambda(M_1 \oplus X') + \lambda\, c(M_1 \oplus X') \\ &= w_\lambda(M_1 \oplus X') + \lambda\, B - \lambda\,(B - c(M_1 \oplus X')) \\ &\geq w_\lambda(M_1) + \lambda\, B - \lambda\,(B - c(M_1 \oplus X')) \\ &\geq \mathrm{opt} - \lambda\,(B - c(M_1 \oplus X')), \end{aligned}$$

where the first inequality follows from (2) and the second inequality follows from (1).

Let $e_2 = x_{(i+h+1) \pmod k}$. The maximality of X' implies that $c(e_2) > B - c(M_1 \oplus X') \geq 0$. Moreover, by the optimality of M_1 and M_2, the Lagrangian weight of any edge $e \in M_1 \cup M_2$ is non-negative, and thus $0 \leq w_\lambda(e_2) = w(e_2) - \lambda\, c(e_2)$. Altogether $\lambda\,(B - c(M_1 \oplus X')) \leq \lambda\, c(e_2) \leq w(e_2)$ and hence $w(M_1 \oplus X') \geq \mathrm{opt} - w(e_2)$. We can thus conclude that

$$w(M) = w(M_1 \oplus X') - w(e_1) \geq \mathrm{opt} - w(e_2) - w(e_1) \geq \mathrm{opt} - 2\, w_{\max}. \qquad \square$$

Theorem 3.2. *There is a deterministic algorithm that, for every $\epsilon > 0$, computes a solution to the budgeted matching problem of weight at least $(1-\epsilon) \cdot opt$ in time $O(m^{2/\epsilon + O(1)})$, where m is the number of edges in the graph.*

Proof. Let $\epsilon \in (0, 1)$ be a given constant. Assume that the optimum matching M^* contains at least $p := \lceil 2/\epsilon \rceil$ edges. (Otherwise the problem can be solved optimally by brute force.) Consider the following algorithm. Initially, we guess the p heaviest (with respect to weights) edges M_H^* of M^*. Then we remove from the graph G the edges in M_H^*, all edges incident to M_H^*, and all edges of weight larger than the smallest weight in M_H^*. We also decrease the budget by $c(M_H^*)$. Let I' be the resulting budgeted matching instance. Note that the maximum weight of an edge in I' is $w'_{\max} \leq w(M_H^*)/p \leq \epsilon w(M_H^*)/2$. Moreover, $M_L^* := M^* \setminus M_H^*$ is an optimum solution for I'. We compute a matching M' for I' using the algorithm described in the proof of Lemma 3.1. Eventually, we output the feasible solution $M := M_H^* \cup M'$.

For a given choice of M_H^* the running time of the algorithm is dominated by the time to compute the two solutions M_1 and M_2. This can be accomplished in $O(m^{O(1)})$ time using Megiddo's parametric search technique [22]. Hence the overall running time of the algorithm is $O(m^{p+O(1)})$, where the m^p factor is due to the guessing of M_H^*. By Lemma 3.1, $w(M') \geq w(M_L^*) - 2\, w'_{\max}$. It follows that

$$\begin{aligned} w(M) = w(M_H^*) + w(M') &\geq w(M_H^*) + w(M_L^*) - 2\, w'_{\max} \\ &\geq w(M^*) - \epsilon\, w(M_H^*) \geq (1-\epsilon)\, w(M^*). \quad \square \end{aligned}$$

4 A PTAS for the Budgeted Matroid Intersection Problem

In this section we will develop a PTAS for the budgeted matroid intersection problem. As in the PTAS for the budgeted matching problem, we will first show how to find a feasible common independent set of two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$ and $\mathcal{M}_2 = (E, \mathcal{F}_2)$ of weight at least $\mathrm{opt} - 2w_{\max}$, where $w_{\max}$ is the weight of the heaviest element. The PTAS will then follow similarly as in the previous section.

Like in the matching case, we initially use Megiddo's parametric search technique to obtain the optimal Lagrangian multiplier $\lambda \geq 0$ and two solutions $X, Y \in \mathcal{F}_1 \cap \mathcal{F}_2$, $c(X) \leq B \leq c(Y)$, that are optimal with respect to the Lagrangian weights $w_\lambda(e) = w(e) - \lambda\, c(e)$, $e \in E$. Notice that

neither X nor Y will contain any element e such that $w_\lambda(e) < 0$. Furthermore, both solutions can be used to derive upper bounds on the optimum solution. In fact, let I^* be the optimum solution of weight $\text{opt} = w(I^*)$. For $Z \in \{X, Y\}$,

$$w_\lambda(Z) + \lambda B \geq w_\lambda(I^*) + \lambda B \geq w_\lambda(I^*) + \lambda c(I^*) = \text{opt}. \tag{3}$$

If X and Y have different cardinalities, say $|X| < |Y|$, we extend $\mathcal{M}_1$ and $\mathcal{M}_2$ according to Lemma 2.1.4 by adding $|Y| - |X|$ dummy elements D of weight and cost zero, and then we replace X by $X \cup D$. (Dummy elements will be discarded when the final solution is returned.) Of course, this does not modify the weight of the optimum solution nor the weight and cost of X. Finally, using Lemma 2.1.3 we truncate the two matroids to $q := |X| = |Y|$. The solutions X and Y will now be maximum-weight common bases of each one of the two truncated matroids.

In the following, we will show how to derive from X and Y a feasible solution of weight at least $\text{opt} - 2w_{\max}$. This is done in two steps. First (Section 4.1), we extract from $X \cup Y$ two adjacent common bases, one below and the other over the budget, with the same (optimal) Lagrangian weight of X and Y. Then (Section 4.2) we apply the Gasoline Lemma to a proper auxiliary graph to compute the desired approximate solution.

4.1 Finding Adjacent Common Bases

The following lemma characterizes two adjacent common bases in the common basis polytope of two matroids.

Lemma 4.1 ([8, 14])**.** *Assume we have two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$, $\mathcal{M}_2 = (E, \mathcal{F}_2)$ and two common bases $X, Y \in \mathcal{F}_1 \cap \mathcal{F}_2$. Then X and Y are adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$ if and only if the following conditions hold:*

1. *The exchangeability graph $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$ has a unique perfect matching M_1.*
2. *The exchangeability graph $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$ has a unique perfect matching M_2.*
3. *The union $M_1 \cup M_2$ forms a cycle.*

The following corollary of Lemma 4.1 will help us to deal with contracted matroids.

Corollary 4.2. *Assume we have two matroids $\mathcal{M}_1 = (E, \mathcal{F}_1)$, $\mathcal{M}_2 = (E, \mathcal{F}_2)$ and two common bases $X, Y \in \mathcal{F}_1 \cap \mathcal{F}_2$. Moreover, let $Z \in \mathcal{F}_1 \cap \mathcal{F}_2$ and $Z \subseteq X \cap Y$. Then X and Y are adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$ if and only if $X \setminus Z$ and $Y \setminus Z$ are adjacent extreme points in the common basis polytope of $\mathcal{M}_1/Z$ and $\mathcal{M}_2/Z$.*

Proof. First note, that X is a basis of $\mathcal{M}_i$ if and only if $X \setminus Z$ is a basis of $\mathcal{M}_i/Z$ $(i = 1, 2)$ by Lemma 2.1.2. The same holds for Y. Moreover, as $Z \subseteq X \cap Y$, the exchangeability graphs $\mathbf{ex}_{\mathcal{M}_i}(X, Y)$ and $\mathbf{ex}_{\mathcal{M}_i/Z}(X \setminus Z, Y \setminus Z)$ $(i = 1, 2)$ are the same, since they are defined on the symmetric difference of X and Y. The claim then follows immediately from Lemma 4.1. □

Remember that X and Y are maximum-weight common bases of $\mathcal{M}_1$ and $\mathcal{M}_2$ with respect to the Lagrangian weights w_λ, and that $c(X) \leq B \leq c(Y)$. Since our solution will be a subset of $X \cup Y$, let us delete the elements $E' = E \setminus (X \cup Y)$ according to Lemma 2.1.1. In order to do a similar patching procedure as for the matching problem, we would like X and Y to be adjacent extreme points in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$. The following lemma will help us to find such two adjacent common bases which are also of maximum weight with respect to w_λ.

Lemma 4.3. *There is a polynomial-time algorithm that finds a third maximum-weight common basis A with respect to w_λ, such that $X \neq A \neq Y$ and $X \cap Y \subseteq A \subseteq X \cup Y$, or determines that no such basis exists.*

Proof. Let $Z = X \cap Y$. Without loss of generality, let $X \setminus Y = \{x_1, \ldots, x_r\}$ and $Y \setminus X = \{y_1, \ldots, y_r\}$. For $1 \leq i, j \leq r$ we now try to find a maximum-weight common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$ that does not

contain x_i and y_j. Denote by $\mathcal{M}_1^{ij} = \mathcal{M}_1/Z - \{x_i, y_j\}$ and $\mathcal{M}_2^{ij} = \mathcal{M}_2/Z - \{x_i, y_j\}$ the matroids resulting from the contraction of Z (Lemma 2.1.2) and the deletion of x_i and y_j (Lemma 2.1.1).

Consider the following (polynomial-time) algorithm. For every $1 \leq i, j \leq r$ compute a maximum-weight common basis A_{ij} of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$. If there is an A_{ij} satisfying $|A_{ij}| = r$ and $w_\lambda(A_{ij}) = w_\lambda(X \setminus Z)$, then $A = A_{ij} \cup Z$ is the desired third basis: A_{ij} is a common basis of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$, and since $|A| = |A_{ij}| + |Z| = |X|$, it is also a common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$. Also, $X \neq A \neq Y$ since x_i and y_j are not present in $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$.

If none of the A_{ij}'s satisfies $|A_{ij}| = r$ and $w_\lambda(A_{ij}) = w_\lambda(X \setminus Z)$, then no common basis A of $\mathcal{M}_1$ and $\mathcal{M}_2$ with the desired properties exists. In fact, assume that there is such a third maximum-weight basis A. Choose i and j such that $x_i, y_j \notin A$. Note that such indices must exist since $X \neq A \neq Y$. Then $A \setminus Z$ is a common basis of $\mathcal{M}_1^{ij}$ and $\mathcal{M}_2^{ij}$. Since A_{ij} is a maximum-weight such common basis, $w_\lambda(A_{ij}) \geq w_\lambda(A \setminus Z)$. Moreover $|A_{ij}| = |A \setminus Z| = r$, and thus $A_{ij} \cup Z$ is a common basis of $\mathcal{M}_1$ and $\mathcal{M}_2$, implying $w_\lambda(A_{ij} \cup Z) \leq w_\lambda(A)$. Hence $w_\lambda(A_{ij}) \leq w_\lambda(A \setminus Z)$. We conclude that $w_\lambda(A_{ij}) = w_\lambda(A \setminus Z) = w_\lambda(X \setminus Z)$. □

We can now apply Lemma 4.3 as follows. As long as we find a third basis A, we replace X by A if $c(A) \leq B$, and Y by A otherwise. In either case, the cardinality of the intersection of the new X and Y increases by at least one. Hence this process ends after at most $O(m)$ rounds.

At the end of the process, X and Y must be adjacent in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$. In fact, $X \setminus Y$ and $Y \setminus X$ are maximum-weight common bases of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$ and there is no other maximum-weight common basis A' of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$, as otherwise $A = A' \cup (X \cap Y)$ would have been found by the algorithm from Lemma 4.3. Now as $X \setminus Y$ and $Y \setminus X$ are the only two maximum-weight common bases, they must also be adjacent on the optimal face of the common basis polytope of $\mathcal{M}_1/(X \cap Y)$ and $\mathcal{M}_2/(X \cap Y)$. Therefore, by Corollary 4.2, X and Y are adjacent in the common basis polytope of $\mathcal{M}_1$ and $\mathcal{M}_2$.

4.2 Merging Adjacent Common Bases

Let X and Y be the two adjacent solutions obtained at the end of the process described in the previous section. Note that if either $c(X) = B$ or $c(Y) = B$, we obtain a feasible solution that is optimal also with respect to the original weights, in which case we are done. In the following, we assume that $c(X) < B < c(Y)$. Without loss of generality, we also assume that $X \setminus Y = \{x_1, x_2, \ldots, x_r\}$ and $Y \setminus X = \{y_1, y_2, \ldots, y_r\}$.

Lemma 4.4. *Given X and Y with the properties above, there is a polynomial-time algorithm which computes a common independent set $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$ such that $c(X') \leq B$ and $w(X') \geq \mathrm{opt} - 2w_{\max}$.*

Proof. We exploit Lemma 4.1 to obtain two unique perfect matchings M_1 in $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$ and M_2 in $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$. Without loss of generality, we can assume that the elements of $X \setminus Y$ and $Y \setminus X$ are denoted such that $M_1 = \{x_1y_1, \ldots, x_ry_r\}$ and $M_2 = \{y_1x_2, y_2x_3, \ldots, y_rx_1\}$. Let $(x_1, y_1, x_2, y_2, \ldots, x_r, y_r)$ be the corresponding cycle. Assign to the each edge x_jy_j a weight $\delta_j := w_\lambda(y_j) - w_\lambda(x_j)$, and weight zero to the remaining edges. Clearly, $\sum_{j=1}^{r} \delta_j = 0$, since X and Y have the same maximum Lagrangian weight. By the Gasoline Lemma, there must exist an edge of the cycle such that the partial sum of the δ-weights of each subpath starting at that edge is non-negative. Without loss of generality, assume x_1y_1 is such an edge. Thus for every $i \in \{1, \ldots, r\}$, $\sum_{j=1}^{i} \delta_j \geq 0$. Find the largest $k \in \{1 \ldots, r\}$ such that

$$c(X) + \sum_{j=1}^{k} \big(c(y_j) - c(x_j)\big) \leq B.$$

Since $c(Y) > B$, we have $k < r$ and by construction

$$c(X) + \sum_{j=1}^{k} \big(c(y_j) - c(x_j)\big) > B - c(y_{k+1}) + c(x_{k+1}).$$

We now show that $X' := X \setminus \{x_1, \ldots, x_{k+1}\} \cup \{y_1, \ldots, y_k\}$ satisfies the claim. By the choice of k, $B - c(y_{k+1}) < c(X') \leq B$. Also, since $\sum_{j=1}^{k} \delta_j \geq 0$, we have

$$w_\lambda(X') \geq w_\lambda(X) - w_\lambda(x_{k+1}) \geq w_\lambda(X) - w_{\max}.$$

We next bound the weight of X':

$$\begin{aligned} w(X') &= w_\lambda(X') + \lambda c(X') = w_\lambda(X') + \lambda B - \lambda(B - c(X')) \\ &\geq w_\lambda(X) + \lambda B - w_{\max} - \lambda c(y_{k+1}) \geq w_\lambda(X) + \lambda B - 2w_{\max} \\ &\geq \text{opt} - 2w_{\max}. \end{aligned}$$

Above we used the fact that $w_\lambda(e) \geq 0$ for all $e \in Y$, so in particular $w_\lambda(y_{k+1}) = w(y_{k+1}) - \lambda c(y_{k+1}) \geq 0$, implying $w_{\max} \geq w(y_{k+1}) \geq \lambda c(y_{k+1})$. The last inequality follows from (3).

It remains to prove that $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$. Consider the set $X' \cup \{x_{k+1}\}$: its symmetric difference with X is the set $\{x_1, \ldots, x_k\} \cup \{y_1, \ldots, y_k\}$. Recall that $x_i y_i$ is an edge of M_1. Thus, for $i \leq k$, it is also an edge of $\mathbf{ex}_{\mathcal{M}_1}(X, X' \cup \{x_{k+1}\})$ so that this graph has a perfect matching. On the other hand this perfect matching must be unique, otherwise M_1 would not be unique in $\mathbf{ex}_{\mathcal{M}_1}(X, Y)$. Thus by the Exchangeability Lemma $X' \cup \{x_{k+1}\} \in \mathcal{F}_1$. Similarly, consider the set $X' \cup \{x_1\}$: its symmetric difference with X is the set $\{x_2, \ldots, x_{k+1}\} \cup \{y_1, \ldots, y_k\}$. For $i \leq k$, $y_i x_{i+1}$ is an edge of M_2. Thus $\mathbf{ex}_{\mathcal{M}_2}(X, X' \cup \{x_1\})$ has a perfect matching, and it has to be unique, otherwise M_2 would not be unique in $\mathbf{ex}_{\mathcal{M}_2}(X, Y)$. Thus by the Exchangeability Lemma $X' \cup \{x_1\} \in \mathcal{F}_2$. We have thus shown that $X' \cup \{x_{k+1}\} \in \mathcal{F}_1$ and $X' \cup \{x_1\} \in \mathcal{F}_2$. As a consequence, $X' \in \mathcal{F}_1 \cap \mathcal{F}_2$. □

Theorem 4.5. *There is a deterministic algorithm that, for every $\epsilon > 0$, computes a solution to the budgeted matroid intersection problem of weight at least $(1 - \epsilon) \cdot opt$ in time $O(m^{2/\epsilon + O(1)})$, where m is the number of elements.*

Proof. Let $\epsilon \in (0, 1)$ be a given constant. Assume that the optimum solution contains at least $p := \lceil 2/\epsilon \rceil$ elements (otherwise the problem can be solved optimally by brute force). We first guess the p elements of largest weight in the optimal solution. Using contraction (Lemma 2.1.2) we remove these elements from both matroids, and using deletion (Lemma 2.1.1) we also remove all elements that have a larger weight than any of the contracted elements. We decrease the budget by the cost of the guessed elements and apply the above algorithm to the resulting instance. Finally, we add back the guessed elements to the returned solution. The final solution has weight at least $\text{opt} - 2w'_{\max}$, where $w'_{\max}$ is the largest weight of the elements that remained after the guessing step. Since $\text{opt} \geq (2/\epsilon) w'_{\max}$, we obtain a solution of weight at least $(1 - \epsilon)\text{opt}$. Similarly to the matching case, the running time of the algorithm is $O(m^{p+O(1)})$, where the m^p factor is due to the guessing. □

5 Concluding Remarks and Open Problems

There are several problems that we left open. One natural question is whether we can apply our patching technique to other budgeted problems. Apparently, the main property that we need is that two adjacent solutions are characterized by a proper path or cycle. This is for example the case for maximum independent sets in claw-free graphs (which follows, for example, from [7]). Indeed, our approach can be used to derive a PTAS for the budgeted maximum independent set problem in this special graph class.

Another natural question is whether our techniques can be extended to the case of multiple budget constraints. The difficulty here is that the Gasoline Lemma alone seems unable to fill in the cost-budget-gap for several budget constraints simultaneously.

In our approach we crucially exploit the fact that removing edges/elements from a feasible solution preserves feasibility. An interesting problem is extending our techniques to problems that do not exhibit this property, such as for example the budgeted version of the maximum perfect matching problem.

Finally, an interesting open problem is whether there are fully polynomial-time approximation schemes (FPTAS) for the problems considered here. (Note that, already with two budget constraints, our problems are strongly NP-hard via reduction from multi-dimensional knapsack [20]). We conjecture that budgeted matching is not strongly NP-hard. However, finding an FPTAS for that problem might be a very difficult task. In fact, consider the following *exact perfect matching problem*: Given an undirected graph $G = (V, E)$, edge weights $w : E \to \mathbb{Q}$, and a parameter $W \in \mathbb{Q}$, find a perfect matching of weight exactly W, if any. This problem was first posed by Papadimitriou and Yannakakis [27]. For polynomial weights, the problem admits a polynomial-time Monte Carlo algorithm [4, 23]. Hence, it is very unlikely that exact perfect matching with polynomial weights is NP-hard (since this would imply RP=NP). However, after several decades, the problem of finding a deterministic algorithm to solve this problem is still open.

Interestingly, for polynomial weights and costs, the budgeted matching problem is equivalent to the exact perfect matching problem. Let the *budgeted perfect matching problem* be the variant of the budgeted matching problem, where we additionally require that the computed matching is perfect.

Lemma 5.1. *For polynomial weights and costs, the following problems are polynomially equivalent: (a) exact perfect matching; (b) budgeted perfect matching; (c) budgeted matching.*

Proof. Without loss of generality, we assume that all weights and costs are non-negative integers. Let $w_{\max}$ and $c_{\max}$ be the largest weight and cost, respectively.

$(a) \Rightarrow (b)$: Let (G, w, W) be an exact perfect matching instance. Solve the budgeted perfect matching instance (G, w, c, B), where $B = W$ and $c(e) = w(e)$ for every edge e. If the solution returned has weight smaller than $B = W$, the original problem is infeasible. Otherwise, the solution computed is a perfect matching of G of weight W.

$(b) \Rightarrow (c)$: Let (G, w, c, B) be a budgeted perfect matching instance. Consider the budgeted matching instance (G, w', c, B), where $w'(e) = w(e) + (n/2+1)w_{\max}$ for every edge e. The original problem is feasible if and only if the maximum matching M^* of the new problem contains $n/2$ edges, i.e., M^* is a perfect matching.

$(c) \Rightarrow (a)$: Let (G, w, c, B) be a budgeted matching instance. For two given W^* and B^*, consider the exact perfect matching instance (G, w', W'), where $W' = (n/2+1)c_{\max}W^* + B^*$ and $w'(e) = (n/2+1)c_{\max}w(e) + c(e)$ for every edge e. The problem (G, w', W') is feasible if and only if there is a matching of weight W^* and cost B^* in the original problem. By trying all the (polynomially many) possible values for W^* and B^*, we obtain the desired solution to the original problem. □

As a consequence, a (deterministic) FPTAS for budgeted matching would solve the mentioned long-standing open problem.

Acknowledgements

The authors would like to thank Jochen Könemann, Rajiv Raman, and Martin Skutella for helpful discussions. Moreover, we would like to thank R. Ravi and Satoru Iwata for pointing out that our techniques can be used to obtain a PTAS for the budgeted independent set problem in claw-free graphs, and Refael Hassin for indicating that the Gasoline Lemma was used in [18].

References

[1] V. Aggarwal, Y. P. Aneja, and K. P. K. Nair. Minimal spanning tree subject to a side constraint. *Computers & Operations Research*, 9(4):287–296, 1982.

[2] F. Barahona and W. R. Pulleyblank. Exact arborescences, matchings and cycles. *Discrete Applied Mathematics*, 16:91–99, 1987.

[3] J. E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394, 1989.

[4] P. Camerini, G. Galbiati, and F. Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13:258–273, 1992.

[5] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. What would Edmonds do? Augmenting paths and witnesses for degree-bounded MSTs. In *Proc. 8th Int. Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 26–39, 2005.

[6] K. Chaudhuri, S. Rao, S. Riesenfeld, and K. Talwar. A push-relabel algorithm for approximating degree bounded MSTs. In *Proc. 33rd Int. Colloquium on Automata, Languages and Programming*, pages 191–201, 2006.

[7] V. Chvátal. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18:138–154, 1975.

[8] A. Frank and É. Tardos. Generalized polymatroids and submodular flows. *Mathematical Programming*, 42:489–563, 1988.

[9] M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *Journal of Algorithms*, 17:409–423, 1994.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[11] M. X. Goemans. Minimum bounded-degree spanning trees. In *Proc. 47th IEEE Symp. on Foundations of Computer Science*, pages 273–282, 2006.

[12] M. Guignard and M. B. Rosenwein. An application of Lagrangean decomposition to the resource-constrained minimum weighted arborescence problem. *Networks*, 20(3):345–359, 1990.

[13] S.-P. Hong, S.-J. Chung, and B. H. Park. A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. *Operations Research Letters*, 32(3):233–239, 2004.

[14] S. Iwata. On matroid intersection adjacency. *Discrete Mathematics*, 242:277–281, 2002.

[15] J. Könemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree bounded minimum spanning trees. *SIAM Journal on Computing*, 31:1783–1793, 2002.

[16] J. Könemann and R. Ravi. Primal-dual meets local search: approximating MSTs with nonuniform degree bounds. *SIAM Journal on Computing*, 34:763–773, 2005.

[17] S. Krogdahl. The dependence graph for bases in matroids. *Discrete Mathematics*, 19:47–59, 1977.

[18] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[19] L. Lovász. *Combinatorial Problems and Exercises*. North-Holland, 1979.

[20] M. J. Magazine and M. S. Chern. A note on approximation schemes for multidimensional knapsack problems. *Mathematics of Operations Research*, 9:244–247, 1984.

[21] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Bicriteria network design problems. In *Proc. 22nd Int. Colloquium on Automata, Languages and Programming*, pages 487–498, 1995.

[22] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4(4):414–424, 1979.

[23] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[24] J. Naor, H. Shachnai, and T. Tamir. Personal communication, 2007.

[25] J. Naor, H. Shachnai, and T. Tamir. Real-time scheduling with a budget. *Algorithmica*, 47(3):343–364, 2007.

[26] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, 1988.

[27] C. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.

[28] C. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proc. 41st IEEE Symp. on Foundations of Computer Science*, pages 86–92, 2000.

[29] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem (extended abstract). In *Proc. 5th Scandinavian Workshop on Algorithms and Theory*, pages 66–75, 1996.

[30] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Many birds with one stone: Multi-objective approximation algorithms. In *Proc. 25th ACM Symp. on the Theory of Computing*, pages 438–447, 1993.

[31] R. Ravi and M. Singh. Delegate and Conquer: An LP-based approximation algorithm for minimum degree MSTs. In *Proc. 33rd Int. Colloquium on Automata, Languages and Programming*, pages 169–180, 2006.

[32] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[33] A. Schrijver. *Combinatorial Optimization. Polyhedra and Efficiency*. Springer, 2003.

[34] D. B. Shmoys and É. Tardos. Scheduling unrelated machines with costs. In *Proc. 4th Symposium on Discrete Algorithms*, pages 448–454, 1993.

[35] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. 39th ACM Symp. on the Theory of Computing*, pages 661–670, 2007.