

New approaches to multi-objective optimization

Fabrizio Grandoni · R. Ravi · Mohit Singh ·
Rico Zenklusen

Received: 30 March 2012 / Accepted: 20 July 2013 / Published online: 8 August 2013
© Springer-Verlag Berlin Heidelberg and Mathematical Optimization Society 2013

Abstract A natural way to deal with multiple, partially conflicting objectives is turning all the objectives but one into budget constraints. Many classical optimization problems, such as maximum spanning tree and forest, shortest path, maximum weight (perfect) matching, maximum weight independent set (basis) in a matroid or in the intersection of two matroids, become NP-hard even with one budget constraint. Still, for most of these problems efficient deterministic and randomized approximation schemes are known. Not much is known however about the case of two or more budgets: filling this gap, at least partially, is the main goal of this paper. In more detail, we obtain the following main results: Using iterative rounding for the first time in multi-objective optimization, we obtain multi-criteria PTASs (which slightly violate the budget constraints) for spanning tree, matroid basis, and bipartite matching with $k = O(1)$ budget constraints. We present a simple mechanism to transform multi-criteria approximation schemes into *pure* approximation schemes for problems whose

A preliminary version of many results in this paper appeared in ESA'09 [17] and ESA'10 [18].

F. Grandoni
IDSIA, University of Italian Switzerland, Manno, Switzerland
e-mail: fabrizio@idsia.ch

R. Ravi
Tepper School of Business, Carnegie Mellon University, Pittsburgh, USA
e-mail: ravi@cmu.edu

M. Singh
School of Computer Science, McGill University, Montreal, Canada
e-mail: mohit@cs.mcgill.edu

R. Zenklusen (✉)
Institute for Operations Research (IFOR), Department of Mathematics, ETH, Zurich, Switzerland
e-mail: rico.zenklusen@ifor.math.ethz.ch

feasible solutions define an independence system. This gives improved algorithms for several problems. In particular, this mechanism can be applied to the above bipartite matching algorithm, hence obtaining a pure PTAS. We show that points in low-dimensional faces of any matroid polytope are almost integral, an interesting result on its own. This gives a deterministic approximation scheme for k -budgeted matroid independent set. We present a deterministic approximation scheme for k -budgeted matching (in general graphs), where $k = O(1)$. Interestingly, to show that our procedure works, we rely on a non-constructive result by Stromquist and Woodall, which is based on the Ham Sandwich Theorem.

Keywords Multi-objective optimization · Multi-budgeted optimization · Approximation algorithms · Combinatorial optimization

Mathematics Subject Classification 90C29 · 90C27

1 Introduction

In many applications, one has to compromise between several, partially conflicting goals. *Multi-Objective Optimization* is a broad area of study in Operations Research, Economics and Computer Science (see [17,36] and references therein). A variety of approaches have been employed to formulate such problems. Here we adopt the Multi-Budgeted Optimization approach [36]: we cast one of the goals as the objective function, and the others as *budget constraints*. More precisely, we are given a (finite) set \mathcal{F} of solutions for the problem, where each solution is a subset S of elements from a given universe E (e.g., the edges of a graph). We are also given a weight function $w : E \rightarrow \mathbb{Q}_+$ and a set of $k = O(1)$ ¹ length functions $\ell^i : E \rightarrow \mathbb{Q}_+$, $1 \leq i \leq k$, that assign a weight $w(S) := \sum_{e \in S} w(e)$ and an i th-length $\ell^i(S) := \sum_{e \in S} \ell^i(e)$, $1 \leq i \leq k$, to every candidate solution S . For each length function ℓ^i , there is a budget $L_i \in \mathbb{Q}_+$. The *k-budgeted optimization problem* can then be formulated as follows:

$$\text{minimize/maximize } w(S) \text{ subject to } S \in \mathcal{F}, \ell^i(S) \leq L_i, 1 \leq i \leq k.$$

We next use *OPT* to denote an optimum solution. A multi-criteria $(\alpha_0, \alpha_1, \dots, \alpha_k)$ -approximation algorithm, $\alpha_i \geq 1$, is a polynomial-time algorithm which produces an α_0 approximate solution S such that $\ell^i(S) \leq \alpha_i L_i$ for all $1 \leq i \leq k$. In particular, $w(S) \geq w(\text{OPT})/\alpha_0$ for a maximization problem, and $w(S) \leq \alpha_0 w(\text{OPT})$ for a minimization one. In a polynomial time approximation scheme (PTAS), $\alpha_0 = 1 + \varepsilon$ for any given constant $\varepsilon > 0$, and all the other α_i 's are 1. In a *multi-criteria* PTAS, all the α_i 's are at most $1 + \varepsilon$. Hence, a multi-criteria PTAS might return slightly infeasible solutions. We sometimes call *pure* a standard PTAS, in order to stress its difference from a multi-criteria PTAS.

Following the literature on the topic, we will focus on the set of problems below:

¹ The assumption that k is a constant is crucial in this paper, since many of the presented algorithms will have a running time that is exponential in k , but polynomial for constant k .

- k -BUDGETED (PERFECT) MATCHING: \mathcal{F} is given by the (perfect) matchings of an undirected graph $G = (V, E)$.
- k -BUDGETED SPANNING TREE (FOREST): \mathcal{F} is given by the spanning trees (forests) of G .
- k -BUDGETED SHORTEST PATH: \mathcal{F} is given by the paths connecting two given nodes s and t in G .
- k -BUDGETED MATROID INDEPENDENT SET (BASIS): \mathcal{F} is given by the independent sets (bases) of a matroid $M = (E, \mathcal{I})$.²
- k -BUDGETED MATROID INTERSECTION INDEPENDENT SET (BASIS): \mathcal{F} is given by the independent sets (bases) in the intersection of two matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$.

We will consider the minimization version of k -BUDGETED SHORTEST PATH. For all the other problems, the minimization version is either trivial or equivalent to its maximization counterpart. Therefore, we will focus only on the maximization version of those problems.

All the above problems are polynomial-time solvable (see, e.g., [23]) in their unbudgeted version ($k = 0$), but become NP-hard [1, 6] even for a single budget constraint ($k = 1$). For the case of one budget ($k = 1$), PTASs are known for SPANNING TREE [35] (see also [21]), SHORTEST PATH [42] (see also [20, 27]), and MATCHING [6]. The approach in [35] easily generalizes to the case of MATROID BASIS. A PTAS is also known for MATROID INTERSECTION INDEPENDENT SET [6]. In the case $2 \leq k = O(1)$, one can use a very general construction by Papadimitriou and Yannakakis [32]. Their technique is based on the construction of ε -approximate Pareto curves, and it can be applied to all the problems whose exact version admits a pseudo-polynomial-time (PPT) deterministic (resp., Monte-Carlo) algorithm. We recall that the *exact version* of a given optimization problem asks for a feasible solution of exactly a given *target* weight. This leads to multi-criteria deterministic (resp., randomized) approximation schemes (with $\alpha_i = 1 + \varepsilon$ for all i). In particular, one can achieve the mentioned approximation for k -BUDGETED SPANNING TREE, k -BUDGETED SHORTEST PATH, and k -BUDGETED (PERFECT) MATCHING.

We note that, if one requires feasible solutions, several of the mentioned problems are inapproximable already for two budget constraints (see also Sect. 2). More precisely, the corresponding feasibility problem is NP-complete. In particular, this holds for k -BUDGETED SHORTEST PATH, k -BUDGETED PERFECT MATCHING and k -BUDGETED SPANNING TREE (and hence also for k -BUDGETED MATROID BASIS and k -BUDGETED MATROID INTERSECTION BASIS). Furthermore, for these problems we can exchange the role of the objective function with any one of the budget constraints. We can conclude that in any (polynomial-time) $(\alpha_0, \alpha_1, \dots, \alpha_k)$ -approximation algorithm for these problems, at most one α_i can be 1.

² We recall that E is a finite ground set and $\mathcal{I} \subseteq 2^E$ is a nonempty family of subsets of E (*independent sets*) which have to satisfy the following two conditions: (i) $I \in \mathcal{I}$, $J \subseteq I \Rightarrow J \in \mathcal{I}$ and (ii) $I, J \in \mathcal{I}$, $|I| > |J| \Rightarrow \exists z \in I \setminus J : J \cup \{z\} \in \mathcal{I}$. A *basis* is a maximal independent set. For all matroids used in this paper we make the usual assumption that independence of a set can be checked in polynomial time. For additional information on matroids, see e.g. [38, Volume B].

We also remark that, for all the other problems, the set of solutions \mathcal{F} forms an *independence system*. In other terms, for $S \in \mathcal{F}$ and $S' \subseteq S$, we have $S' \in \mathcal{F}$.

Our Results. We obtain the following main results:

(1) Using the iterative rounding framework, we obtain simple deterministic $(1, 1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation algorithm for k -BUDGETED SPANNING TREE and k -BUDGETED MATROID BASIS. This improves on the $(1 + \varepsilon, 1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation algorithms for the same problems in [32], and it is best possible approximation-wise from the above discussion. Furthermore, we obtain a (more involved) deterministic $(1 + \varepsilon, 1 + \varepsilon, \dots, 1 + \varepsilon)$ -approximation algorithm for k -BUDGETED BIPARTITE MATCHING. In contrast, the approach in [32] achieves the same approximation for general graphs, but the algorithm is Monte-Carlo.

The algorithm for k -BUDGETED SPANNING TREE is rather simple; a vertex solution for the natural LP relaxation of the problem is already sparse: it has about k edges more than a spanning tree in its support due to the well-known laminarity of an independent set of tight spanning tree constraints (see, e.g., [14]). We remove all edges corresponding to variables of value zero, relax (remove) all the budget constraints, and solve optimally the residual problem (which is a standard spanning tree problem). A preliminary guessing phase ensures that the k edges not used in the tree do not add much to the approximation bound for any of the budgets. This approach also gives a very simple proof of the earlier result for the case $k = 1$ [35]. An identical approach works also for the more general k -BUDGETED MATROID BASIS problem.

Our algorithm for k -BUDGETED BIPARTITE MATCHING is more involved: after an initial preprocessing phase, where the algorithm removes all edges with large weight and large length, there is a decomposition phase. In that phase, we run an iterative relaxation algorithm which uses the optimal solution of the natural LP formulation to obtain a modified LP solution. The iterative algorithm ensures that the support of the modified solution is a collection of $h \leq k$ vertex disjoint paths. Moreover, each of these paths has small weight and length. In the final combination phase, we combine the solutions on these paths to return one feasible matching. Each path can be decomposed in two matchings. The algorithm picks one matching from each of the paths. While the algorithm is a brute force enumeration over all choices (which are $2^h \leq 2^k$ many), a probabilistic argument is used to show that there exists a choice of a matching from each path which provides a solution with the desired guarantee.

Perhaps even more importantly than these specific results, our main contribution here is to demonstrate that the general framework of iterative rounding can be used to obtain approximation algorithms for various multi-objective optimization problems.

(2) We present a simple but powerful mechanism to transform a multi-criteria PTAS into a pure PTAS for problems whose feasible solutions define an independence system. Similarly, a multi-criteria polynomial randomized time approximation scheme (PRAS) can be transformed into a pure PRAS. The basic idea is as follows. We show that a *good* solution exists even if we scale down the budgets by a small factor. This is done by applying a greedy discarding strategy similar to the greedy algorithm for KNAPSACK. Applying a multi-criteria PTAS (given as a black box) to the scaled problem gives a feasible solution for the original one, of weight *close* to the optimal weight.

To the best of our knowledge, this simple result was not observed before. Indeed, it implies improved approximation algorithms for a number of problems. In particular, we can combine our mechanism with the construction in [32]. For example, using the PPT-algorithm for EXACT FOREST in [5], one obtains a PTAS for k -BUDGETED FOREST. Similarly, the Monte-Carlo PPT-algorithm for EXACT MATCHING in [30] gives a PRAS for k -BUDGETED MATCHING. The Monte-Carlo PPT-algorithms for EXACT MATROID INTERSECTION INDEPENDENT SET in [9], which works in the special case of representable matroids,³ implies a PRAS for the corresponding k -budgeted problem.

Of course, one can also exploit multi-criteria approximation schemes obtained with different techniques. For example, exploiting the multi-criteria PTAS for k -BUDGETED BIPARTITE MATCHING that we present in this paper, one obtains a PTAS for the same problem. Very recently [10], a multi-criteria PRAS for k -BUDGETED MATROID INDEPENDENT SET, based on dependent randomized rounding, has been presented. This implies a PRAS for k -BUDGETED MATROID INDEPENDENT SET.

(3) Based on a different, more direct approach, we obtain a PTAS (rather than a PRAS) for k -BUDGETED MATROID INDEPENDENT SET. The main insight here is a structural property of faces of the matroid polytope⁴ which might be of independent interest. Essentially, we show that points in low-dimensional faces of any matroid polytope are almost integral (i.e., they contain few fractional components). More precisely, if the face has dimension d , then at most $2d$ components are fractional. A PTAS can then easily be derived as follows. We first guess the most expensive elements in the optimum solution, and reduce the problem consequently. Then we compute an optimal (basic) fractional solution: since the relaxation consists of the matroid polytope with k additional linear constraints, the obtained fractional solution lies on a face of the matroid polytope which is at most k -dimensional. Consequently, it has at most $2k$ fractional components. By rounding down such fractional components, we obtain a feasible integral solution with the desired approximation guarantee.

(4) Finally, we present a PTAS for k -BUDGETED MATCHING (in arbitrary graphs). Our PTAS works as follows. Let us confuse a matching M with the associated incidence vector x_M . We initially compute an optimal fractional matching x^* to the natural LP relaxation of k -BUDGETED MATCHING, and express it as a convex combination $x^* = \sum_{j=1}^{k+1} \alpha_j x_j$ of $k+1$ (or less) matchings x_1, \dots, x_{k+1} . Then we exploit a *merging procedure* which, given two matchings x' and x'' with a parameter $\alpha \in [0, 1]$, computes a matching y which is not longer than $z := \alpha x' + (1-\alpha)x''$ with respect to all k lengths, and has comparable weight. This procedure is applied successively: first on the matchings x_1 and x_2 with parameter $\alpha = \alpha_1/(\alpha_1 + \alpha_2)$, hence getting a matching y' . Then, on the two matchings y' and x_3 with parameter $\alpha = (\alpha_1 + \alpha_2)/(\alpha_1 + \alpha_2 + \alpha_3)$, and so on. The resulting matching is feasible and almost optimal, when performing a preliminary guessing step before applying the patching procedure.

³ A matroid $M = (E, \mathcal{I})$ is representable if its ground set E can be mapped in a bijective way to the columns of a matrix over some field, and $I \subseteq E$ is independent in M iff the corresponding columns are linearly independent.

⁴ For some given matroid $M = (E, \mathcal{I})$, the corresponding matroid polytope $P_{\mathcal{I}}$ is the convex hull of the incidence vectors of all independent sets.

An interesting aspect of our procedure is that it relies on a non-constructive theorem of Stromquist and Woodall, which in turn relies on the Ham Sandwich Theorem. The theorem of Stromquist and Woodall implies that some structure exists, which guarantees that the merging procedure we present works.

Related work. There are a few general tools for designing approximation algorithms for budgeted problems. One basic approach is combining *dynamic programming* (which solves the problem for polynomial weights and lengths) with *rounding and scaling* techniques (to reduce the problem to the case of polynomial quantities). This leads for example to the FPTAS for 1- BUDGETED SHORTEST PATH [20,27,42]. Another fundamental technique is the *Lagrangian relaxation method*. The basic idea is relaxing the budget constraints, and lifting them into the objective function, where they are weighted by Lagrangian multipliers. Solving the relaxed problem, one obtains two or more solutions with optimal Lagrangian weight, which can—if needed—be patched together to get a good solution for the original problem. Demonstrating this method, Goemans and Ravi [35] gave a PTAS for 1- BUDGETED SPANNING TREE, which also extends to 1- BUDGETED MATROID BASIS. Inspired by this approach, Correa and Levin [12] presented algorithms for special classes of polynomial-time covering problems with an additional covering constraint. Using the same approach as Goemans and Ravi, with an involved patching step, Berger, Bonifaci, Grandoni, and Schäfer [6] obtained a PTAS for 1- BUDGETED MATCHING and 1- BUDGETED MATROID INTERSECTION INDEPENDENT SET. Their approach does not seem to generalize to the case of multiple budget constraints.

The techniques above apply to the case of one budget. Not much is known for problems with two or more budgets. However, often *multi-criteria* approximation schemes are known, which provide a $(1 + \varepsilon)$ -approximate solution violating the budgets by a factor $(1 + \varepsilon)$. First of all, there is a very general technique by Papadimitriou and Yannakakis [32], based on the construction of ε -approximate Pareto curves. Given an optimization problem with multiple objectives, the *Pareto curve* consists of the set of solutions S such that there is no solution S' which is strictly better than S (in a vectorial sense). Papadimitriou and Yannakakis show that, for any constant $\varepsilon > 0$, there always exists a polynomial-size ε -approximate Pareto curve \mathcal{A} , i.e., a set of solutions such that every solution in the Pareto curve is within a factor of $(1 + \varepsilon)$ from some solution in \mathcal{A} on each objective. Furthermore, this approximate curve can be constructed in polynomial time in the size of the input and $1/\varepsilon$ whenever there exists a PPT algorithm for the associated exact problem. This implies multi-criteria FPTASs for k -BUDGETED SPANNING TREE and k -BUDGETED SHORTEST PATH. Furthermore, it implies a multi-criteria FPRAS for k -BUDGETED (PERFECT) MATCHING. The latter result exploits the Monte-Carlo PPT algorithm for EXACT MATCHING in [30].

The iterative rounding technique was introduced by Jain [22] for approximating survivable network design problems. The basic idea in iterative rounding for covering problems is as follows: Consider an optimal (fractional) vertex (or extreme point or basic feasible) solution to a linear programming relaxation to the problem, and show that there is a variable with high fractional value (e.g. at least 0.5) which can be rounded up to an integer without losing too much (e.g. 2) in the approximation. The method includes this rounded variable in the integral solution and iterates on a reduced prob-

lem where the integral variables are fixed. This method can be enhanced by adding a relaxation step, where one relaxes a constraint that can be ignored without losing too much in the feasibility. The iterative relaxation method has been very successful for approximating degree-constrained network design problems [24,25,39,43] and directed network design problems [4]. Recently, using an iterative randomized rounding approach, Byrka et al. [8] developed an improved approximation algorithm for the Steiner tree problem which was further developed in [16]. In the context of these methods, our paper shows that iterative rounding is a powerful and flexible tool also for approximating multi-objective optimization problems

All mentioned problems are easy in the unbudgeted version. Given an NP-hard unbudgeted problem which admits a ρ -approximation, the *parametric search* technique in [28] provides a multi-criteria $k\rho$ -approximation algorithm violating each budget by a factor $k\rho$ for the corresponding problem with k budgets. This only gives a much weaker k -approximation for each objective for the problems considered here. Other techniques lead to logarithmic approximation factors (see, e.g., [7,33,34]).

Subsequent work. After the conference versions of this paper, relevant progress has been made on some of the problems that we consider here. In [11], a randomized rounding approach was suggested which leads to a PRAS for k -BUDGETED MATROID INTERSECTION. Furthermore, also in [11], a PTAS for k -BUDGETED MATCHING was obtained. This algorithm is based on the derandomization of a PRAS which is obtained by applying Chernoff bounds to a randomized rounding procedure which iteratively merges pairs of matchings along similar lines as we do here. To obtain sufficient concentration, the symmetric difference of two matchings to merge is cut into $\Theta(k \log k / \epsilon^2)$ pieces, and within each of these pieces the edges of one of the two matchings are kept, which is decided randomly. When derandomizing the procedure, all possible $2^{\Theta(k \log k / \epsilon^2)}$ random outcomes have to be checked in each merge iteration. An advantage of the algorithm for k -BUDGETED MATCHING that we present here, is that the running time of our algorithm does not depend on ϵ , apart of the initial guessing step. This is particularly of interest for instances where the weight and lengths of each edge is sufficiently small such that the guessing step can be simplified or even skipped.

Organization. The rest of this paper is organized as follows. In Sect. 2 we discuss the approximability of part of the mentioned problems. In Sect. 3 we present our multi-criteria approximation schemes for k -BUDGETED SPANNING TREE, k -BUDGETED MATROID BASIS and k -BUDGETED BIPARTITE MATCHING. Section 4 contains our pure approximation schemes. In particular, we describe our feasibility mechanism, give a PTAS for k -BUDGETED MATROID INDEPENDENT SET, and a PTAS for k -BUDGETED MATCHING.

2 A simple hardness result

As a warm-up for the reader, we start by observing a few simple facts about the complexity of the mentioned problems. The following simple theorem might be considered as part of folklore.

Theorem 2.1 For $k \geq 2$, it is *NP*-complete to decide whether there is a feasible solution for k -BUDGETED SHORTEST PATH, k -BUDGETED PERFECT MATCHING and k -BUDGETED SPANNING TREE (and hence also for k -BUDGETED MATROID BASIS and k -BUDGETED MATROID INTERSECTION BASIS).

Proof It is sufficient to prove the claim for $k = 2$. Consider first 2- BUDGETED SPANNING TREE: the claim for k -BUDGETED MATROID BASIS and, consequently, for k -BUDGETED MATROID INTERSECTION BASIS trivially follows. Let \mathcal{P}^\pm denote our (feasibility) problem, and \mathcal{P}^\pm its variant with arbitrary (i.e., positive and/or negative) lengths. Of course, \mathcal{P}^\pm includes \mathcal{P}^+ as a special case. To see the opposite reduction, observe that a spanning tree contains exactly $n - 1$ edges. Hence, by adding a sufficiently large value M to all the lengths, and adding $(n - 1)M$ to the budgets, one obtains an equivalent problem with non-negative lengths. It is easy to see that \mathcal{P}^\pm includes as a special case the problem $\mathcal{P}^=$ of determining, for a given length function $\ell'(\cdot)$ and target L' , whether there exists a spanning tree T of length $\ell'(S) = L'$: a reduction is obtained by setting $\ell^1(\cdot) = -\ell^2(\cdot) = \ell'(\cdot)$ and $L_1 = -L_2 = L'$. Hence it is sufficient to show that $\mathcal{P}^=$ is *NP*-complete. We do that via the following reduction from PARTITION: given $\alpha_1, \alpha_2, \dots, \alpha_q \in \mathbb{Q}$ and a target $A \in \mathbb{Q}$, determine whether there exists a subset of α_i 's of total value A . Consider graph G_q , consisting of q cycles C_1, C_2, \dots, C_q , with $C_i = (a_i, b_i, c_i, d_i)$ and $c_i = a_{i+1}$ for $i = 1, 2, \dots, q - 1$. Let $\ell'(a_i b_i) = \alpha_i$, $i = 1, 2, \dots, q$, and set to zero all the other lengths. The target is $L' = A$. Trivially, for each spanning tree T and each cycle C_i , the length of $T \cap C_i$ is either 0 or α_i . Hence, the answer to the input partition problem is yes if and only if the same holds for the associated instance of $\mathcal{P}^=$.

Consider now 2- BUDGETED PERFECT MATCHING. Since each perfect matching contains exactly $n/2$ edges, with the same argument and notation as above it is sufficient to prove the *NP*-completeness of the problem $\mathcal{P}^=$ of determining, for a given length function $\ell'(\cdot)$ and target L' , whether there exists a perfect matching M of length $\ell'(M) = L'$. We use a similar reduction from PARTITION as above. The graph is again given by the cycles C_1, \dots, C_q . However, this time each cycle forms a distinct connected component. We use the same lengths ℓ' as above and we again set $L' = A$. It is easy to see that, for each perfect matching M and each cycle C_i , the length of $M \cap C_i$ is either 0 or α_i . The claim follows. Of course, an even simpler reduction is obtained when working with multigraphs, where the used graph can be reduced to distinct connected components each consisting of two parallel edges.

Eventually consider 2- BUDGETED SHORTEST PATH. We restrict our attention to the graph G_q as used for the spanning tree reduction, and let $(s, t) = (a_1, c_q)$. Since any s - t path in this graph uses exactly $2q$ edges, we have by the usual argument that it is sufficient to show the *NP*-completeness of the problem $\mathcal{P}^=$ of determining, for a given length function $\ell'(\cdot)$ and target L' , whether there exists an s - t path P of length $\ell'(P) = L'$. The claim follows by essentially the same reduction as in the spanning tree case. \square

Corollary 2.2 Unless $P = NP$, there is no $(\alpha_0, \alpha_1, \dots, \alpha_k)$ -approximation algorithm with two or more α_i 's equal to 1 for the problems in the claim of Theorem 2.1.

Proof Observe that one can exchange the roles of the objective function with any one of the budget constraints for the mentioned problems. The claim follows from Theorem 2.1. □

3 Multi-criteria approximation schemes

In this section we present our multi-criteria approximation schemes, which slightly violate budget constraints. All these algorithms are based on iterative randomized rounding. We start with the matroid basis case and conclude the section with the much more involved algorithm for bipartite matching.

3.1 k -BUDGETED MATROID BASIS

Consider the following linear programming relaxation (LP-MB) for the problem. There is a variable x_e for each element $e \in E$. For any subset $S \subseteq E$, we denote $x(S) = \sum_{e \in S} x_e$. Here r denotes the rank function of the matroid \mathcal{M} .

$$\begin{aligned}
 \text{(LP-MB)} \quad & \text{maximize} && \sum_{e \in E} w(e) x_e \\
 & \text{subject to} && x(E) = r(E), \\
 & && x(S) \leq r(S), && \forall S \subseteq E \\
 & && \sum_{e \in E} \ell^i(e) x_e \leq L_i, && \forall 1 \leq i \leq k \\
 & && x_e \geq 0, && \forall e \in E.
 \end{aligned}$$

The polynomial time solvability of the linear program (LP-MB) follows from the polynomial time separation of the rank constraints [13]. The following characterization follows from a standard uncrossing argument. A proof is presented for completeness.

We recall that a *chain* is a family \mathcal{F} of sets such that for any $F_1, F_2 \in \mathcal{F}$, we have either $F_1 \subseteq F_2$ or $F_2 \subseteq F_1$.

Furthermore, for any $S \subseteq E$, we denote by $\chi(S) \in \{0, 1\}^E$ the *incidence vector* of S , i.e., $(\chi(S))_e = 1$ for $e \in S$ and $(\chi(S))_e = 0$ for $e \in E \setminus S$.

Lemma 3.1 *Let x be a vertex solution of the linear program (LP-MB) such that $x_e > 0$ for each $e \in E$ and let $\mathcal{T} = \{S \subseteq E \mid x(S) = r(S)\}$ be the set of all tight subset constraints. Then there exists a chain $\mathcal{C} \subseteq \mathcal{T}$ and a subset $J \subseteq \{1 \leq j \leq k \mid \sum_{e \in E} \ell^j(e) x_e = L_j\}$ of tight length constraints such that*

1. *The vectors $\{\chi(S) \mid S \in \mathcal{C}\} \cup \{\ell^i \mid i \in J\}$ are linearly independent,*
2. *$\text{span}(\{\chi(S) \mid S \in \mathcal{C}\}) = \text{span}(\{\chi(S) \mid S \in \mathcal{T}\})$,*
3. *$|\mathcal{C}| + |J| = |E|$.*

Proof Since (LP-MB) is a polytope with $|E|$ variables, any vertex of (LP-MB) can be described as the intersection of $|E|$ constraints of (LP-MB) that are linearly independent and tight with respect to x . Since $x_e > 0$ for $e \in E$, the only constraints of (LP-MB) that can be tight with respect to x are rank constraints, i.e., $x(S) = r(S)$ for some

1. Guess all elements in the optimal solution satisfying $\ell^i(e) \geq \frac{\epsilon}{k}L_i$. Include all such elements in the solution and update the matroid by contracting these elements in the matroid. Delete all other heavy elements e with $\ell^i(e) \geq \frac{\epsilon}{k}L_i$ for any i from \mathcal{M} . Update L_i .
2. Find a vertex solution x of (LP-MB) for the residual problem.
3. Pick any maximum weight basis in the support of x .

Fig. 1 Algorithm for k -BUDGETED MATROID BASIS

set $S \subseteq E$, and length constraints. In general, there may be many choices of $|E|$ linearly independent and tight constraints that define x . Actually, among all tight constraints with respect to x one can choose any maximal subset of linearly independent constraints to define x . We first choose a maximal number of linearly independent and tight rank constraints, which can be represented by a family of tight sets $\mathcal{C} \subseteq \mathcal{T}$. Notice that by choosing a maximal family we have $\text{span}(\{\chi(S) \mid S \in \mathcal{C}\}) = \text{span}(\{\chi(S) \mid S \in \mathcal{T}\})$. The chosen tight rank constraints define a face of the matroid polytope. By [38, p. 778], any face of the matroid polytope can be defined by a family of tight rank constraints corresponding to sets \mathcal{C} that form a chain (together with possibly tight nonnegativity constraints, which does not apply to our case since $x > 0$). Hence, we can choose \mathcal{C} to be a chain. We complete the chosen rank constraints with a maximal number of tight length constraints that are linearly independent among each other and with respect to the chosen rank constraints. We represent those constraints by the indices of the chosen length constraints $J \subseteq \{1 \leq j \leq k \mid \sum_{e \in E} \ell^j(e)x_e = L_j\}$. Hence, $\{\chi(S) \mid S \in \mathcal{F}\} \cup \{\ell^i \mid i \in J\}$ are linearly independent vectors by constructions. Since they uniquely define x , we have $|\mathcal{C}| + |J| = |E|$, thus completing the proof. \square

Consider the algorithm for k -BUDGETED MATROID BASIS in Fig. 1. We first perform a pruning step to *guess* all elements in the optimal solution with i th-length at least $\frac{\epsilon L_i}{k}$ for any $1 \leq i \leq k$. Then we solve the linear program (LP-MB) for the residual problem and remove all elements which the linear program sets to zero. We then select the maximum weight basis under weight function w ignoring the rest of the length functions. Observe that the last step is equivalent to relaxing all the k length constraints and solving the integral linear program for the matroid basis problem.

Theorem 3.2 *For any $\epsilon > 0$, there exists an algorithm for k -BUDGETED MATROID BASIS, $k = O(1)$, which returns a basis B with $\ell^i(B) \leq (1 + \epsilon)L_i$ for each $1 \leq i \leq k$, and $w(B) \geq w(OPT)$, where OPT is a maximum-weight basis that satisfies all length constraints. The running time of the algorithm is $O(m^{O(k^2/\epsilon)})$.*

Proof Consider the algorithm described in Fig. 1, whose running time trivially satisfies the claim. First observe that the support of a vertex solution to (LP-MB) on a matroid with $r(E) = n$ has at most $n + k$ elements. In fact, from Lemma 3.1, we have $|E| = |\mathcal{C}| + |J|$. But $|\mathcal{C}| \leq r(E)$ since \mathcal{C} is a chain and $x(C)$ equals a distinct integer between 1 and $r(E)$ for each $C \in \mathcal{C}$. Also $|J| \leq k$ proving the claim. Let L'_i be the i th budget of the residual problem solved in step 2 of the algorithm. Observe that the weight of the basis returned is at least the weight of the LP-solution and hence is at

least $w(OPT)$. Now, we show that the i th-length is at most $L'_i + \epsilon L_i$. Observe that any basis must contain $r(E)$ elements out of the $r(E) + k$ elements in the support. Hence, the longest i th-length basis differs from the minimum i th-length basis by at most $k \cdot \frac{\epsilon}{k} L'_i = \epsilon L'_i$. But the minimum i th-length basis has i th-length at most the length of the fractional basis which is at most L'_i . The claim follows. \square

3.2 k -BUDGETED BIPARTITE MATCHING

In this section we present a multi-criteria PTAS for k -BUDGETED BIPARTITE MATCHING.

We formulate the following linear programming relaxation (LP-BM) for the problem. We use $\delta(v)$ to denote the set of edges incident to $v \in V$.

$$\begin{aligned}
 \text{(LP-BM)} \quad & \text{maximize} && \sum_{e \in E} w(e) x_e \\
 & \text{subject to} && \sum_{e \in \delta(v)} x_e \leq 1, && \forall v \in V \\
 & && \sum_{e \in E} \ell^i(e) x_e \leq L_i, && \forall 1 \leq i \leq k \\
 & && x_e \geq 0, && \forall e \in E.
 \end{aligned}$$

Consider the algorithm for k -BUDGETED BIPARTITE MATCHING in Fig. 2. Our algorithm works in three phases.

In the *Preprocessing Phase*, the algorithm guesses all the edges in OPT of weight at least $\delta w(OPT)$ or i th-length at least δL_i for some i . Here δ is a proper function of ϵ and k . This guessing can be performed in time polynomial in n (but exponential in δ). The algorithm then includes all the guessed edges in the solution, and deletes the remaining heavy edges and all edges incident to vertices which have already been matched by guessed edges. It also reduces the L_i 's accordingly. After this phase $w(e) \leq \delta w(OPT)$ and $\ell^i(e) \leq \delta L_i$ for each edge e .

In the *Decomposition Phase* our algorithm computes over a series of pruning and iterative steps, a solution to the k -budgeted matching problem on a reduced graph that is eventually a collection of paths. In Step (c), we discard nodes of degree 0 or of degree 3 or higher so as to leave only paths and cycles; Finally, one edge from each cycle is removed in this step. In Step (e), we further break each path into subpaths of bounded total weight and length. This pruning is useful in the later Combination Phase when we choose one of the two matchings in each path: the bounded difference ensures that one such combination is near optimal. The use of vertex solutions in all the residual problems ensures that the total number of edges thrown away in all the above stages is roughly of the order of the extra budget constraints in the problem which is $O(k/\gamma)$ for a parameter $\gamma \simeq O(\epsilon/\sqrt{k})$. Finally, we output a feasible fractional vertex solution x^g to the LP with the following properties:

- (1) The support of x^g is a collection of vertex disjoint paths S_1, \dots, S_h where $h \leq k$.
- (2) x^g is a $(1 + \epsilon/4)$ -approximate solution.

Preprocessing

- (a) Let $\delta = \epsilon^2 / (36k\sqrt{2k \ln(k+2)})$. Guess all the edges e in OPT such that $w(e) \geq \delta w(OPT)$ or $\ell^i(e) \geq \delta L_i$ for some i , and add them to the solution. Reduce the problem consequently.

Decomposition

- (b) Compute the optimal fractional vertex solution x^b to LP-BM for the reduced problem. As long as there is an integral variable, reduce the problem appropriately and iterate.
- (c) Remove all the nodes of degree zero and of degree at least 3, and all the edges incident to the removed nodes. Compute an optimal fractional vertex solution x^c to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate. Finally, remove one edge from each remaining cycle.
- (d) Compute an optimal fractional vertex solution x^d to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate.
- (e) Let $\gamma = \epsilon / (2\sqrt{2k \ln(k+2)})$. As long as there is a path $P = (e_1, e_2, \dots, e_t)$ in the support of x^d such that $w(P) > \gamma w(x^d)$ or $\ell^i(P) > \gamma \ell^i(x^d)$ for some i , find a minimal prefix $P' = (e_1, e_2, \dots, e_{t'})$ of P satisfying the condition above and remove $e_{t'}$ from the graph.
- (f) Compute an optimal fractional vertex solution x^f to the problem LP-BM in the remaining graph. As long as there is an integral variable, reduce the problem appropriately and iterate.
- (g) Let P_1, P_2, \dots, P_q be the set of paths induced by x^f . Return the subpaths S_1, S_2, \dots, S_h formed after deleting the internal nodes whose matching constraints are not tight with respect to x^f . Return the solution x^g which is x^f induced on the edges in S_i for each $1 \leq i \leq h$.

Combination

- (h) Let M_j and \bar{M}_j be the two matchings partitioning S_j . Return the matching M' satisfying the following properties: (i) For each S_j , $M' \cap S_j \in \{M_j, \bar{M}_j\}$; (ii) $w(M') \geq (1 - \epsilon/2)w(x^g)$ and $\ell^i(M') \leq (1 + \epsilon/2)\ell^i(x^g)$ for all i .

Fig. 2 Algorithm for k -BUDGETED BIPARTITE MATCHING

- (3) For each S_i , the degree constraints of the vertices of S_i are tight except for its endpoints.
- (4) For each S_j , $w \cdot x^g(S_j) \leq \gamma w(OPT)$ and $\ell^i \cdot x^g(S_j) \leq \gamma L_i$ for each $1 \leq i \leq k$ and $1 \leq j \leq h$ where $\gamma = \epsilon / (2\sqrt{2k \ln(k+2)})$.

In the final *Combination Phase*, the paths S_1, \dots, S_h are used to compute an approximate feasible (integral) solution. The algorithm enumerates over all the 2^h matchings which are obtained by taking, for each S_i , one of the two matchings which partition S_i . This enumeration takes polynomial time since $h \leq k = O(1)$. A probabilistic argument is used to show that one of these matchings satisfies the claimed approximation guarantee of the algorithm.

Analysis. We now analyze the three phases of the algorithm, bounding the corresponding approximation guarantee and running time. Consider first the *Preprocessing*

Phase. In order to implement Step (a), we have to consider all the possible choices, and run the algorithm for each choice. Observe that there are at most $(k + 1)/\delta$ such heavy edges in the optimal solution, and hence the number of possibilities is $O(m^{(k+1)/\delta}) = O(m^{O(k^2 \sqrt{k \log k}/\epsilon^2)})$. The algorithm generates a different subproblem for each possible guess of the edges. In the following we will focus on the run of the algorithm where the guessed edges correspond to an optimal solution.

Consider now the *Decomposition Phase*. We prove that the output of this phase satisfies the four properties stated above. Observe that by construction the algorithm returns a collection of edge disjoint paths whose interior vertices have tight degree constraints. Properties (3) and (4) follow by construction. We now argue that the number of paths is bounded by k , proving Property (1).

Lemma 3.3 *The number h of subpaths in Step (g) is upper bounded by k .*

Proof Consider the solution x^f . The number of variables $|E| = \sum_{i=1}^q |P_i|$ is upper bounded by the number of tight constraints. Let q' be the number of internal nodes whose matching constraint is not tight in x^f . Note that the matching constraints at the endpoints of each path are not tight. Hence the number of tight constraints is at most $\sum_{i=1}^q (|P_i| - 1) - q' + k = |E| - q - q' + k \geq |E|$, from which $q + q' \leq k$. Observe that, by definition, the number h of subpaths is exactly $q + q'$ (we start with q subpaths, and create a new subpath for each internal node whose matching constraint is not tight). The claim follows. \square

Clearly, solution x^g satisfies all the constraints. We next argue that the weight of x^g is nearly optimal. In Steps (c), (e) and (g) we remove a subset of edges whose optimal fractional value is larger than zero in the step considered. In the following lemma we bound the number of edges removed. Due to the Preprocessing Phase, the weight of these edges is negligible, which implies that the consequent worsening of the approximation factor is sufficiently small. This proves Property (2).

Lemma 3.4 *The algorithm removes at most*

1. $7k$ edges in Step (c);
2. $(k + 1)/\gamma$ edges in Step (e);
3. $2k$ edges in Step (g).

Proof (1) In the beginning of Step (c), all variables are strictly fractional. Thus, every vertex in V_1 , the set of vertices with tight degree constraints, has degree at least two. Let E be the residual edges. Note that $|E| \leq |V_1| + k$ since the number of tight constraints is at most $|V_1| + k$. Let H be the set of nodes of degree at least 3. Observe that

$$\begin{aligned}
 2(|V_1| + k) &\geq 2|E| \geq \sum_{v \in V_1} \text{deg}(v) \geq \sum_{v \in V_1 \setminus H} 2 \\
 &+ \sum_{v \in H} \text{deg}(v) \implies \sum_{v \in H} \text{deg}(v) \leq 2|H| + 2k.
 \end{aligned}$$

Since $\text{deg}(v) \geq 3$ for each $v \in H$ we have $|H| \leq 2k$. Thus $\sum_{v \in H} \text{deg}(v) \leq 6k$.

After removing nodes of degree 0 and at least 3, the graph consists of a set of paths and cycles. Let C_1, C_2, \dots, C_q be the set of cycles, and P_1, P_2, \dots, P_r be the set of paths. We next show that $q \leq k$ with an analogous counting argument, and hence at most k more edges are removed. Since G is bipartite, each cycle C_i must be even and therefore the corresponding matching constraints must be dependent. Moreover, the endpoints of each path cannot correspond to tight matching constraints. Thus the total number of edges over all such even cycles is $|E| = \sum_{i=1}^q |C_i| + \sum_{j=1}^r |P_j|$ while the number of tight and independent degree constraints and budget constraints is at most $\sum_{i=1}^q (|C_i| - 1) + \sum_{j=1}^r (|P_j| - 1) + k = |E| - q - r + k$. Since the number of variables is at most the number of tight and independent constraints at any vertex solution, we obtain that the number of cycles and paths altogether is at most k .

- (2) Each minimal subpath P' considered in Step (e) satisfies either $w(P') > \gamma w(x^d)$ or $\ell^i(P') > \gamma \ell^i(x^d)$ for some i . Since the edges of P' are not considered any more in the following iterations of Step (e), the condition $w(P') > \gamma w(x^d)$ can be satisfied at most $1/\gamma$ times. Similarly for the condition $\ell^i(P') > \gamma \ell^i(x^d)$. It follows that the number of minimal subpaths, and hence the number of edges remove, is upper bounded by $(k + 1)/\gamma$.
- (3) Let V' be the set of internal vertices for which the degree constraints are not tight in the paths P_1, \dots, P_q and let V_1 be the set of vertices with tight degree constraints. Thus, we have $deg(v) \geq 2$ for each $v \in V_1 \cup V'$. But the total number of edges is at most $|V_1| + k$. Thus we have $|V'| \leq k$. We remove exactly two edges for each vertex in V' obtaining the claimed bound.

□

Each of the steps (b) to (g) is run polynomially many times and takes polynomial time. Hence the overall running time of the Decomposition Phase is polynomial.

Consider eventually the *Combination Phase*. As described earlier, the running time of this phase is bounded by $O(2^k n^{O(1)})$. The following lemma, which is the heart of our analysis, shows that a subset M' satisfying Properties (i) and (ii) defined in the *Combination Phase* (h) always exists. Henceforth the algorithm always returns a solution. Although we use a randomized argument to prove the lemma, the algorithm is completely deterministic and enumerates over all solutions. Recall that M_j and \bar{M}_j are the two matchings which partition subpath S_j .

Lemma 3.5 *In Step (h) there is always a set of edges M' satisfying Properties (i) and (ii).*

Proof Consider the following packing problem

$$\begin{aligned}
 (\text{PACK}) \quad & \text{maximize} \quad \sum_{j=1}^h (y_j w(M_j) + (1 - y_j) w(\bar{M}_j)) \\
 & \text{subject to} \quad \sum_{j=1}^h (y_j \ell^i(M_j) + (1 - y_j) \ell^i(\bar{M}_j)) \leq L_i, \quad \forall 1 \leq i \leq k \\
 & \quad \quad \quad y_j \in \{0, 1\}, \quad \quad \quad \forall 1 \leq j \leq h.
 \end{aligned}$$

We can interpret the variables y_j in the following way: $M' \cap S_j = M_j$ if $y_j = 1$, and $M' \cap S_j = \bar{M}_j$ otherwise. Given a (possibly fractional and infeasible) solution y to PACK, we use $w(y)$ and $\ell^i(y)$ as shortcuts for $\sum_{j=1}^h (y_j w(M_j) + (1 - y_j) w(\bar{M}_j))$ and $\sum_{j=1}^h (y_j \ell^i(M_j) + (1 - y_j) \ell^i(\bar{M}_j))$, respectively.

We first show that the solution x^g can be interpreted as a feasible solution y^g to the linear relaxation of PACK as follows. Consider each subpath S_j . By definition, each matching constraint at an internal node of S_j is tight. This implies that all the edges e of M_j (resp., \bar{M}_j) have the same value $x_e^g =: y^g$ (resp., $x_e^g =: 1 - y^g$). Thus, we have $w(y^g) = w(x^g)$.

Now, we construct an integral solution y' in the following manner. Independently, for each path S_i , select M_i with probability y_i^g and \bar{M}_i with probability $1 - y_i^g$. Note that $E[w(y')] = w(y^g)$ and $E[\ell^i(y')] = \ell^i(y^g) \leq L_i$ for all i . In order to prove the claim, it is sufficient to show that, with positive probability, one has

$$w(y') \geq (1 - \epsilon/2)w(x^g) \quad \text{and} \quad \ell^i(y') \leq (1 + \epsilon/2)\ell^i(x^g) \text{ for all } i.$$

This implies that a matching satisfying (i) and (ii) always exists, and hence the algorithm will find it.

By Step (e), switching one variable of y' from 1 to 0 or vice versa can change the cost and i th-length of y' at most by $\gamma w(x^g)$ and $\gamma \ell^i(x^g)$, respectively. Using the method of bounded differences (see, e.g., [29]):

$$\begin{aligned} Pr(w(y') < E[w(y')] - t) &\leq e^{-\frac{t^2}{2h(\gamma w(x^g))^2}} \quad \text{and} \\ Pr(\ell^i(y') > E[\ell^i(y')] + t) &\leq e^{-\frac{t^2}{2h(\gamma \ell^i(x^g))^2}}. \end{aligned}$$

Recalling that $E[w(y')] \geq w(x^g)$, $h \leq k$, and setting $t = \epsilon/2 \cdot w(x^g) = \gamma w(x^g)\sqrt{2k \ln(k+2)}$,

$$\begin{aligned} Pr(w(y') < w(x^g) - \epsilon/2 \cdot w(x^g)) &\leq Pr(w(y') < E[w(y')] - \gamma w(x^g)\sqrt{2k \ln(k+2)}) \\ &\leq e^{-(\gamma w(x^g))^2 2k \ln(k+2)/2h(\gamma w(x^g))^2} \\ &\leq e^{-\ln(k+2)} = \frac{1}{k+2}. \end{aligned}$$

Similarly, for all i ,

$$Pr(\ell^i(y') > \ell^i(x^g) + \epsilon/2 \cdot \ell^i(x^g)) \leq \frac{1}{k+2}.$$

From the union bound, the probability that y' does not satisfy Property (ii) is therefore at most $\frac{k+1}{k+2} < 1$. The claim follows. □

Theorem 3.6. *For any $\epsilon > 0$, there exists a deterministic algorithm for k -BUDGETED BIPARTITE MATCHING, $k = O(1)$, which returns a matching M of weight $w(M) \geq (1 - \epsilon)w(OPT)$ and length $\ell^i(M) \leq (1 + \epsilon)L_i$ for each $1 \leq i \leq k$. The running time of the algorithm is $O(n^{O(k^2 \sqrt{k \log k / \epsilon^2})})$.*

Proof Consider the above algorithm, whose running time is trivially as in the claim. It is easy to see that the solution returned is a matching. Moreover a solution is always returned by Lemma 3.5. The approximation guarantee of the algorithm follows from the properties of the Decomposition step and Lemma 3.5. \square

4 Pure approximation schemes for independence systems

In this section we present our pure approximation schemes (which do not violate any budget constraint) when the solution space \mathcal{F} is an independence system, i.e. if $F \in \mathcal{F}$ and $F' \subseteq F$ then $F' \in \mathcal{F}$. We start by describing our feasibilization mechanism to turn multi-criteria approximation schemes into pure approximation schemes. We then present our deterministic approximation scheme for matroid independent set. We conclude the section with a deterministic approximation scheme for matchings (in general graphs) with k budget constraints, where $k = (1)$ as usual.

4.1 A feasibilization mechanism

Since we deal with independence systems, minimization problems are trivial (the empty solution is optimal). Therefore, we will consider maximization problems only. Analogous to terminology used in matroid theory, for an independence system \mathcal{F} on some ground set E and any $I \in \mathcal{F}$, we call the independence system $\{S \in E \setminus I \mid S \cup I \in \mathcal{F}\}$ on ground set $E \setminus I$ a *contraction* of \mathcal{F} . Similarly, for any $I \subseteq E$, the independence system $\{S \in E \setminus I \mid S \in \mathcal{F}\}$ is called a *restriction* of \mathcal{F} . Combination of contractions and restrictions are called *minors*.

We say that a family \mathcal{F} of independence systems is *self-reducible* if it is closed under taking minors. Self-reducibility is a natural property for independence system, examples include feasible solutions to knapsack problems, graphic matroids, linear matroids, matchings, and bipartite matchings.

Theorem 4.1 (*Feasibilization*) *Let \mathcal{F} be a self-reducible family of independence systems. Suppose that we are given an algorithm \mathcal{A} which, for any constant $\delta > 0$ and k -budgeted optimization problem \mathcal{P}_{ind} on an independence system $\mathcal{F} \in \mathcal{F}$, computes in polynomial time a solution $S \in \mathcal{F}$ to the k -budgeted maximization problem on \mathcal{F} of cost (resp., expected cost) at least $(1 - \delta)$ times the optimum in \mathcal{F} , violating each budget by a factor of at most $(1 + \delta)$. Then there is a PTAS (resp., PRAS) for \mathcal{P}_{ind} .⁵*

Proof Let $\varepsilon \in (0, 1]$ be a given constant, with $1/\varepsilon \in \mathbb{N}$. Consider the following algorithm. Initially we guess the $h = k/\varepsilon$ elements E_H of OPT of largest weight, and reduce the problem consequently, hence getting a problem \mathcal{P}' . Then we scale down all the budgets by a factor $(1 - \delta)$, and solve the resulting problem \mathcal{P}'' by means of \mathcal{A} , where $\delta = \varepsilon/(k + 1)$. Let E_L be the solution returned by \mathcal{A} . We finally output $E_H \cup E_L$.

⁵ Notice that it suffices to assume that \mathcal{F} is closed under contractions, since a restriction can be emulated by setting the weights of the elements to be removed to zero.

Let OPT' and OPT'' be the optimum solution to problems \mathcal{P}' and \mathcal{P}'' , respectively. We also denote by L'_i and L''_i the i th budget in the two problems, respectively. Let w_{max} be the largest weight in \mathcal{P}' and \mathcal{P}'' . We observe that trivially: (a) $w(OPT) = w(E_H) + w(OPT')$ and (b) $w_{max} \leq w(E_H)/h$.

Let us show that (c) $w(OPT'') \geq w(OPT')(1 - k\delta) - kw_{max}$. Consider the following process: for each length function i , we remove from OPT' the element e with smallest ratio $w(e)/\ell^i(e)$ until the remaining elements of OPT' have i th length $\leq (1 - \delta)L'_i$. Let E_i be the set of elements removed. More formally, we number the elements of $OPT' = \{e_1, e_2, \dots, e_q\}$ such that $w(e_1)/\ell^i(e_1) \leq w(e_2)/\ell^i(e_2) \leq \dots \leq w(e_q)/\ell^i(e_q)$. Hence, $E_i = \{e_1, \dots, e_r\}$, where $r \in \{0, \dots, q\}$ is the smallest index such that $\ell^i(\{e_{r+1}, \dots, e_q\}) \leq (1 - \delta)L'_i$. We now show $w(E_i) \leq \delta w(OPT') + w_{max}$. This is trivially true if $E_i = \emptyset$, hence, we assume without loss of generality $r \geq 1$. Since $\ell^i(OPT') \leq L'_i$ and r is the smallest index with $\ell^i(\{e_{r+1}, \dots, e_q\}) \leq (1 - \delta)L'_i$, or equivalently $\ell^i(\{e_1, \dots, e_r\}) \geq \ell^i(OPT') - (1 - \delta)L'_i$, we get

$$\ell^i(\{e_1, \dots, e_{r-1}\}) < \ell^i(OPT') - (1 - \delta)L'_i \leq \delta \ell^i(OPT'). \tag{1}$$

Furthermore, notice that for any four reals $A, B, a, b > 0$ with $\frac{A}{B} \geq \frac{a}{b}$, we have $\frac{a+A}{b+B} \geq \frac{a}{b}$. Applying this inequality repeatedly, we obtain

$$\frac{w(e_1)}{\ell^i(e_1)} \leq \frac{w(\{e_1, e_2\})}{\ell^i(\{e_1, e_2\})} \leq \dots \leq \frac{w(OPT')}{\ell^i(OPT')},$$

and in particular

$$\frac{w(\{e_1, \dots, e_{r-1}\})}{\ell^i(\{e_1, \dots, e_{r-1}\})} \leq \frac{w(OPT')}{\ell^i(OPT')}. \tag{2}$$

Hence,

$$\begin{aligned} w(E_i) &= w(e_r) + w(\{e_1, \dots, e_{r-1}\}) \\ &\stackrel{(2)}{\leq} w_{max} + \ell^i(\{e_1, \dots, e_{r-1}\}) \cdot \frac{w(OPT')}{\ell^i(OPT')} \\ &\stackrel{(1)}{\leq} w_{max} + \delta w(OPT'), \end{aligned}$$

as claimed.

It follows that $OPT' - \cup_i E_i$ is a feasible solution for \mathcal{P}' of weight at least $w(OPT')(1 - \delta k) - kw_{max}$, proving (c).

We observe that E_L is feasible for \mathcal{P}' since, for each i , $\ell^i(E_L) \leq (1 + \delta)L''_i = (1 + \delta)(1 - \delta)L'_i \leq L'_i$. As a consequence, the returned solution $E_H \cup E_L$ is feasible. Moreover, when \mathcal{A} is deterministic, we have

$$\begin{aligned}
 w(E_H) + w(E_L) &\geq w(E_H) + (1 - \delta)w(OPT'') \\
 &\stackrel{(c)}{\geq} w(E_H) + (1 - \delta)(w(OPT')(1 - \delta k) - kw_{max}) \\
 &\stackrel{(b)}{\geq} (1 - k/h)w(E_H) + (1 - \delta(k + 1))w(OPT') \\
 &\geq (1 - \varepsilon)(w(E_H) + w(OPT')) \stackrel{(a)}{=} (1 - \varepsilon)w(OPT).
 \end{aligned}$$

The same bound holds in expectation when \mathcal{A} is randomized. □

Corollary 4.2 *There are PTASs for k -BUDGETED FOREST and k -BUDGETED BIPARTITE MATCHING. There are PRASs for k -BUDGETED MATCHING, k -BUDGETED MATROID INDEPENDENT SET, and k -BUDGETED MATROID INTERSECTION in representable matroids.*

Proof The result about bipartite matching follows from the multi-criteria PTAS in previous section. All the other results follow from known multi-criteria PTASs and PRASs [5,9,10,30,32]. □

4.2 A PTAS for k -BUDGETED MATROID INDEPENDENT SET

Again, we denote by $r(S) = \max\{|J| \mid J \subseteq S, J \in \mathcal{I}\}$ the rank function of a matroid $M = (E, \mathcal{I})$. Furthermore, $P_{\mathcal{I}} = \{x \geq 0 \mid x(S) \leq r(S) \forall S \subseteq E\}$ denotes the *matroid polytope* which is the convex hull of the characteristic vectors χ_I of the independent sets $I \in \mathcal{I}$.

Theorem 4.3 *Let $M = (E, \mathcal{I})$ be a matroid and let F be a face of dimension d of the matroid polytope $P_{\mathcal{I}}$. Then any $x \in F$ has at most $2d$ non-integral components. Furthermore, the sum of all fractional components of x is at most d .*

Proof Let $m = |E|$. We assume that the matroid polytope has full dimension, i.e., $\dim(P_{\mathcal{I}}) = m$, or equivalently, every element $e \in E$ is independent. This can be assumed w.l.o.g. since if $\{e\} \notin \mathcal{I}$ for some $e \in E$, then we can reduce the matroid by deleting element e .

By [38, p. 778], any d -dimensional face F of a polymatroid, which is a generalization of a matroid polytope, can be described as follows

$$F = \{x \in P_{\mathcal{I}} \mid x(e) = 0 \forall e \in N, x(A_i) = r(A_i) \forall i \in \{1, \dots, k\}\},$$

where $A_1 \subsetneq A_2 \subsetneq \dots \subsetneq A_k \subseteq E$, and $N \subseteq E$ with $|N| + k = m - d$.

We prove the claim by induction on the number of elements of the matroid. The theorem clearly holds for matroids with a ground set of cardinality one. First assume $N \neq \emptyset$ and let $e \in N$. Let M' be the matroid obtained from M by deleting e , and let F' be the projection of F onto the coordinates corresponding to $N \setminus \{e\}$. Since F' is a face of M' , the claim follows by induction. Henceforth, we assume $N = \emptyset$ which implies $k = m - d$. Let $A_0 = \emptyset$ and $B_i = A_i \setminus A_{i-1}$ for $i \in \{1, \dots, k\}$. In the following we show that we can assume

$$0 < r(A_i) - r(A_{i-1}) < |B_i| \quad \forall i \in \{1, \dots, k\}. \tag{3}$$

Notice that $0 \leq r(A_i) - r(A_{i-1}) \leq |B_i|$ clearly holds by standard properties of rank functions (see [38, p. 664] for more details). Assume that there is $i \in \{1, \dots, k\}$ with $r(A_i) = r(A_{i-1})$. Since all points $x \in F$ satisfy $x(A_i) = r(A_i)$ and $x(A_{i-1}) = r(A_{i-1})$, we have $x(B_i) = 0$. Hence for any $e \in B_i$, we have $x(e) = 0$ for $x \in F$. Again, we can delete e from the matroid, hence obtaining a smaller matroid for which the claim holds by the inductive hypothesis. Therefore, we can assume $r(A_i) > r(A_{i-1})$ which implies the left inequality in (3).

For the right inequality assume that there is $i \in \{1, \dots, k\}$ with $r(A_i) - r(A_{i-1}) = |B_i|$. Hence, every $x \in F$ satisfies $x(B_i) = |B_i|$, implying $x(e) = 1$ for all $e \in B_i$. Let $e \in B_i$, and let F' be the projection of the face F onto the components $N \setminus \{e\}$. Since F' is a face of the matroid M' obtained from M by contracting e , the result follows again by the inductive hypothesis.

Henceforth, we assume that (3) holds. This implies in particular that $|B_i| > 1$ for $i \in \{1, \dots, k\}$. Since $\sum_{i=1}^k |B_i| \leq m$, we have $k \leq m/2$, which together with $k = m - d$ implies $d \geq m/2$. The claim of the theorem that $x \in F$ has at most $2d$ non-integral components is thus trivial in this case.

To prove the second part of the theorem we show that if (3) holds then $x(E) \leq d$ for $x \in F$. For $x \in F$ we have

$$\begin{aligned} x(E) &= x(E \setminus A_k) + \sum_{i=1}^k x(B_i) \\ &\leq |E| - |A_k| + \sum_{i=1}^k (r(A_i) - r(A_{i-1})) \\ &\leq |E| - |A_k| + \sum_{i=1}^k (|A_i| - |A_{i-1}| - 1) \\ &= m - k = d, \end{aligned}$$

where the first inequality follows from $x(E \setminus A_k) \leq |E \setminus A_k|$ and $x(B_i) = r(A_i) - r(A_{i-1})$, and the second inequality follows from (3). □

Exploiting Theorem 4.3, we suggest in Fig. 3 a conceptually simple PTAS for multi-budgeted optimization over the independent sets of a matroid.

-
1. Guess the k/ε elements $E_H \subseteq E$ of largest weight in OPT and reduce the problem consequently, by removing from E the set E_H and all elements with weight strictly larger than the lowest weight among the elements in E_H . This leads to a new matroid $M' = (E', \mathcal{I}')$ for the reduced problem.
 2. Compute an optimal basic solution x^* to the LP which seeks to find a maximum weight point in $P_{T'}$ intersected with the k budget constraints.
 3. Let $E_L = \{e \in E \mid x^*(e) = 1\}$.
 4. Return $E_H \cup E_L$.
-

Fig. 3 A PTAS for k -BUDGETED MATROID INDEPENDENT SET

Corollary 4.4 *The algorithm presented in Fig. 3 is a PTAS for k -BUDGETED MATROID INDEPENDENT SET.*

Proof The guessing step guarantees that the maximum weight w_{\max} of an element in the reduced problem satisfies $kw_{\max} \leq \epsilon w(E_H)$. Since x^* is chosen to be a vertex solution, and only k linear constraints are added to the matroid polytope, x^* lies on a face of the matroid polytope $P_{\mathcal{I}'}$ of dimension at most k . By Theorem 4.3, the sum of all fractional components of x^* is at most k , i.e., $x^*(E' \setminus E_L) \leq k$. Hence,

$$\begin{aligned} w(E_L) &= \sum_{e \in E'} w(e)x^*(e) - \sum_{e \in E' \setminus E_L} w(e)x^*(e) \\ &\geq \sum_{e \in E'} w(e)x^*(e) - w_{\max} \sum_{e \in E' \setminus E_L} x^*(e) \\ &\geq \sum_{e \in E'} w(e)x^*(e) - kw_{\max}. \end{aligned}$$

Furthermore, since we solved a relaxation of the original problem, we have

$$w(OPT) \leq w(E_H) + \sum_{e \in E'} w(e)x^*(e).$$

Combining the above inequalities, and using $kw_{\max} \leq \epsilon w(E_H)$, we obtain

$$\begin{aligned} w(E_H \cup E_L) &\geq w(E_H) + \sum_{e \in E'} w(e)x^*(e) - kw_{\max} \\ &\geq w(OPT) - kw_{\max} \\ &\geq w(OPT) - \delta w(E_H) \\ &\geq (1 - \delta)w(OPT). \end{aligned}$$

□

4.3 A PTAS for k -BUDGETED MATCHING

In this section we present our PTAS for k -BUDGETED MATCHING. We denote by \mathcal{M} the set of incidence vectors of matchings. With a slight abuse of terminology we call the elements in \mathcal{M} matchings. Let $P_{\mathcal{M}}$ be the matching polytope. To simplify the exposition, it is convenient to consider weights w and lengths ℓ^i for $i \in \{0, \dots, k\}$ sometimes as vectors in \mathbb{Q}_+^E . We denote by $\ell = (\ell^1, \dots, \ell^k)$ the matrix whose i th column is ℓ^i , and let $L = (L_1, \dots, L_k)^T$ be the vector corresponding to the budgets. Using this terminology, a feasible solution to the k -BUDGETED MATCHING problem is a matching $x \in \mathcal{M}$ such that $\ell^T x \leq L$.

We will first describe a procedure that returns a feasible matching of weight at least $w(OPT) - \frac{(k+3)k^2}{k+1}w_{\max}$, where $w_{\max} = \max\{w(e) \mid e \in E\}$. Similar to the algorithms seen in previous sections, it then suffices to perform a preprocessing step to guess the

1. Compute a basic solution x^* to the LP $\max\{w^T x \mid x \in P_{\mathcal{M}}, \ell^T x \leq L\}$.
2. Express x^* as a convex combination $x^* = \sum_{j=1}^{k+1} \alpha_j x_j$ of matchings $x_1, \dots, x_{k+1} \in P_{\mathcal{M}}$. The numbering is chosen such that $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_{k+1}$.
3. Let $y_1 = x_1$, and let $\beta_j = \sum_{r=1}^j \alpha_r$ for $j \in \{1, \dots, k+1\}$.
4. **For** $j = 2, \dots, k+1$:
 Compute a matching $y_j = \text{MERGE}(\frac{\beta_{j-1}}{\beta_j}, y_{j-1}, \frac{\alpha_j}{\beta_j}, x_j)$ such that the following holds, where $z_j = \frac{\beta_{j-1}}{\beta_j} y_{j-1} + \frac{\alpha_j}{\beta_j} x_j$:
 - (a) $(\ell^i)^T y_j \leq (\ell^i)^T z_j \quad \forall i \in \{1, \dots, k\}$,
 - (b) $w^T y_j \geq w^T z_j - 2kw_{\max}$.
5. Return $y = y_{k+1}$.

Fig. 4 Obtaining a feasible matching $y \in \mathcal{M}$ for k -BUDGETED MATCHING with $w^T y \leq w(OPT) - \frac{(k+3)k^2}{k+1} w_{\max}$

$\lceil \frac{(k+3)k^2}{\epsilon(k+1)} \rceil$ heaviest edges of the optimal solution. Our algorithm starts with an optimal basic solution x^* to the natural LP relaxation $\max\{w^T x \mid x \in \mathcal{P}_{\mathcal{M}}, \ell^T x \leq L\}$, rewrites x^* as a convex combination of at most $k+1$ matchings that are then successively merged to obtain a feasible matching. The procedure, modulo the sub procedure MERGE, is described in Fig. 4.

Notice that since x^* is a vertex of the polytope $P_{\mathcal{M}}$ with k additional linear constraints, it lies on a face of $P_{\mathcal{M}}$ of dimension at most k . Hence, by Carathéodory’s Theorem, x^* can indeed be expressed as a convex combination of at most $k+1$ matchings as done in step 2 of the algorithm. Such a decomposition of x^* can be obtained efficiently by standard techniques (see for example [37]). Notice that if fewer than $k+1$ matchings are needed in the decomposition of x^* , then this corresponds to having some of the α_j equal to zero.

The procedure $\text{MERGE}(\lambda, y, \mu, x)$ takes four arguments, where $\lambda, \mu \geq 0, \lambda + \mu = 1$ and $x, y \in \mathcal{M}$ are matchings, and returns in polynomial time a matching \bar{y} that satisfies conditions (a) and (b) highlighted in the algorithm, i.e, $(\ell^i)^T \bar{y} \leq (\ell^i)^T (\lambda y + \mu x)$ for $i \in \{1, \dots, k\}$ and $w^T \bar{y} \geq w^T (\lambda y + \mu x) - 2kw_{\max}$. Intuitively, the MERGE procedure takes a point $z = \lambda y + (1 - \lambda)x$ on an edge of the matching polytope $\mathcal{P}_{\mathcal{M}}$ and returns a matching \bar{y} with weight and lengths similar to z .

We give the details of MERGE later, and first show the following, assuming that MERGE returns a matching with the properties described in the algorithm.

Theorem 4.5 *Given an efficient MERGE procedure fulfilling the requirements described in the algorithm in Fig. 4, the algorithm in Fig. 4 is an efficient procedure that returns a feasible matching y for k -BUDGETED MATCHING with $w^T y \leq w(OPT) - \frac{(k+3)k^2}{k+1} w_{\max}$.*

Proof The algorithm is clearly efficient.

To prove feasibility of $y = y_{k+1}$, we fix $i \in \{1, \dots, k\}$ and show by induction on $j \in \{1, \dots, k + 1\}$ that

$$(\ell^i)^T y_j \leq (\ell^i)^T \left(\frac{1}{\beta_j} \sum_{r=1}^j \alpha_r x_r \right). \tag{4}$$

Feasibility then follows from feasibility of x^* since

$$(\ell^i)^T y_{k+1} \stackrel{(4)}{\leq} (\ell^i)^T \left(\frac{1}{\beta_{k+1}} \sum_{r=1}^{k+1} \alpha_r x_r \right) = (\ell^i)^T x^* \leq L_i.$$

Since $y_1 = x_1$, (4) trivially holds for $j = 1$. Furthermore, let $j \in \{1, \dots, k\}$ and assume that (4) holds for any value less or equal to j . Then using property (a) of MERGE, we obtain

$$\begin{aligned} (\ell^i)^T y_{j+1} &\stackrel{(a)}{\leq} (\ell^i)^T z_{j+1} \\ &= (\ell^i)^T \left(\frac{\beta_j}{\beta_{j+1}} y_j + \frac{\alpha_{j+1}}{\beta_{j+1}} x_{j+1} \right) \stackrel{ind. hyp. (4)}{\leq} (\ell^i)^T \left(\frac{1}{\beta_{j+1}} \sum_{r=1}^{j+1} \alpha_r x_r \right), \end{aligned}$$

thus proving (4) and implying feasibility of y .

Similarly, to prove that y has large weight, we show by induction on $j \in \{1, \dots, k + 1\}$ that

$$w^T y_j \geq \frac{1}{\beta_j} \left(w^T \left(\sum_{r=1}^j \alpha_r x_r \right) - \left(\sum_{r=2}^j \beta_r \right) 2k w_{\max} \right). \tag{5}$$

The desired result on the weight of $y = y_{k+1}$ then follows by observing that $\beta_r \leq \frac{r}{k+1}$, since $\alpha_1 \leq \dots \leq \alpha_{k+1}$, and

$$\begin{aligned} w^T y_{k+1} &\stackrel{(5)}{\geq} w^T \left(\sum_{r=1}^{k+1} \alpha_r x_r \right) - \left(\sum_{r=2}^{k+1} \beta_r \right) 2k w_{\max} \\ &\geq w^T x^* - \left(\sum_{r=2}^{k+1} \frac{r}{k+1} \right) 2k w_{\max} \\ &= w(OPT) - \frac{(k+3)k^2}{k+1} w_{\max}. \end{aligned}$$

Again, (5) is trivially true for $j = 1$ since $y_1 = x_1$. Furthermore, for $j \in \{1, \dots, k\}$ we obtain the following by using property (b) of MERGE:

$$\begin{aligned}
 w^T y_{j+1} &\stackrel{(b)}{\geq} w^T z_{j+1} - 2kw_{\max} = \frac{1}{\beta_{j+1}} w^T (\beta_j y_j + \alpha_{j+1} x_{j+1}) - 2kw_{\max} \\
 &\stackrel{\text{ind. hyp. (5)}}{\geq} \frac{1}{\beta_{j+1}} \left(w^T \left(\sum_{r=1}^j \alpha_r x_r \right) - \left(\sum_{r=2}^j \beta_r \right) 2kw_{\max} + w^T (\alpha_{j+1} x_{j+1}) \right) \\
 -2kw_{\max} &= \frac{1}{\beta_{j+1}} \left(w^T \left(\sum_{r=1}^{j+1} \alpha_r x_r \right) - \left(\sum_{r=2}^{j+1} \beta_r \right) 2kw_{\max} \right).
 \end{aligned}$$

□

Hence, it remains to present an efficient MERGE procedure fulfilling property (a) and (b).

4.3.1 Merging procedure

Consider fixed input parameters $x', x'' \in \mathcal{M}$ and $\alpha \in [0, 1]$ of $\text{MERGE}(\alpha, x', 1-\alpha, x'')$. MERGE works in two steps. First, it constructs a fractional point $y \in [0, 1]^E$ that is structurally close to being a matching in a well-defined sense, and satisfies

$$w^T y = w^T (\alpha x' + (1 - \alpha)x''), \text{ and} \tag{6}$$

$$(\ell^i)^T y = (\ell^i)^T (\alpha x' + (1 - \alpha)x'') \quad \forall i \in \{1, \dots, k\}. \tag{7}$$

In a second step y is transformed to a matching with the desired properties.

More precisely, we want y to be a $2k$ -almost matching, defined as follows.

Definition 4.6 (*r*-almost matching) For $r \in \mathbb{N}$, a vector $y \in [0, 1]^E$ is an *r*-almost matching in $G = (V, E)$ if it is possible to set at most r components of y to zero to obtain a matching.

We denote by \mathcal{M}_r the set of all *r*-almost matchings.

Given an *r*-almost matching y , we say that z is a *corresponding matching* (to y), if it is a matching obtained by setting at most r components of y to zero. Such a corresponding matching z can easily be found by first setting all fractional components of y to zero, and then computing a maximum cardinality matching in the resulting set of edges. Clearly, a corresponding matching z satisfies $w^T z \geq w^T y - rw_{\max}$. Hence, to obtain a MERGE procedure with the desired properties, it indeed suffices to present an efficient procedure to construct a $2k$ -almost matching y that satisfies (6) and (7): a corresponding matching z clearly has lower lengths, since $z \leq y$, and

$$w^T z \geq w^T y - 2kw_{\max} = w^T (\alpha x' + (1 - \alpha)x'') - 2kw_{\max},$$

by the above observation and (7). Hence, we now focus on efficiently getting a $2k$ -almost matching y satisfying (6) and (7).

To construct y , we start with x' and define a way to fractionally swap parts of x' with x'' . For this we define a generalized symmetric difference. For two vectors $z', z'' \in [0, 1]^E$, we define their *symmetric difference* $z' \Delta z'' \in [0, 1]^E$ by $(z' \Delta z'')(e) = |z'(e) - z''(e)|$ for all $e \in E$. In particular, if z' and z'' are incidence vectors, then their symmetric difference as defined above corresponds indeed to the symmetric difference in the usual sense. Recall that, when z' and z'' are matchings, $z' \Delta z''$ consists of a set of node-disjoint paths and cycles.

Consider the paths and cycles in $s = x' \Delta x''$. We number the edges $\{e_0, \dots, e_{\tau-1}\}$ in s such that two consecutively numbered edges are either consecutive in some path/cycle or belong to different paths/cycles. This can easily be achieved by cutting each cycle, appending the resulting set of paths one to the other, gluing together the endpoints of the obtained path, and then number consecutively the edges in the obtained path. For $t \in [0, \tau]$, we define $s(t) \in [0, 1]^E$ as

$$(s(t))(e) = \begin{cases} 1 & \text{if } e = e_i, i \in \{0, \dots, \lfloor t \rfloor\}; \\ t - \lfloor t \rfloor & \text{if } e = e_i, i = \lfloor t \rfloor \text{ and } i \leq \tau - 1; \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, for $a, b \in [0, \tau]$ we define $s(a, b) \in [0, 1]^E$ as follows

$$s(a, b) = \begin{cases} s(b) - s(a) & \text{if } a \leq b, \\ s(a) + s(\tau) - s(b) & \text{if } a > b. \end{cases}$$

Hence, $s(a, b)$ can be interpreted as a fractional version of a cyclic interval of edges in s . We denote by $[a, b]_{\odot}$, the cyclic interval contained in $[0, \tau]$, which is defined by

$$[a, b]_{\odot} = \begin{cases} [a, b] & \text{if } a \leq b, \\ [0, a] \cup [b, \tau] & \text{if } a > b. \end{cases}$$

Notice that for $a, b \in [0, \tau]$, the vector $y = x' \Delta s(a, b)$ is a 2-almost matching since to obtain a matching, it suffices to set the coordinates of y that correspond to $e_{\lfloor a \rfloor}$ and $e_{\lfloor b \rfloor}$ to zero. Analogously, for any family of k disjoint cyclic intervals $[a_1, b_1]_{\odot} \dots [a_k, b_k]_{\odot} \subseteq [0, \tau]$, the vector

$$y = x' \Delta s(a_1, b_1) \Delta \dots \Delta s(a_k, b_k) \tag{8}$$

is a $2k$ -almost matching. We will find k disjoint cyclic intervals such that y as defined in (8) satisfies (6) and (7).

We first prove non-constructively the existence of intervals $[a_1, b_1]_{\odot}, \dots, [a_k, b_k]_{\odot}$ that lead to a desired y , using the following slight generalization of a Theorem of Stromquist and Woodall [40]. In the theorem below, a *signed measure* denotes the difference between two finite measures. Furthermore, we call a measure *non-atomic* if every singleton has measure zero.

Theorem 4.7 *Let $d \geq 2$, and let μ_1, \dots, μ_d be non-atomic signed measures on $[0, 1]$. For each $\lambda \in [0, 1]$ there is a set $K_\lambda \subseteq [0, 1]$ that is the union of at most $d - 1$ circular intervals of $[0, 1]$ and satisfies $\mu_i(K_\lambda) = \lambda\mu_i([0, 1]) \forall i \in \{1, \dots, d\}$.*

The difference between Theorem 4.7 and [40, Theorem 1] is that we allow for signed measures instead of usual measure. Even though Theorem 4.7 is a direct generalization of Stromquist and Woodall’s Theorem, for completeness, we provide a proof in Sect. 4.3.2. Notice that Theorem 4.7 could be equivalently stated for the interval $[0, \tau]$ instead of $[0, 1]$. Based on Theorem 4.7 we obtain the following.

Theorem 4.8 *There exist k disjoint cyclic intervals $[a_1, b_1]_{\odot}, \dots, [a_k, b_k]_{\odot} \subseteq [0, \tau]$, some of which may be empty, such that the vector*

$$y = x' \Delta s(a_1, b_1) \Delta \dots \Delta s(a_k, b_k)$$

satisfies (6) and (7).

Proof We define the following $k + 1$ signed measures μ_1, \dots, μ_{k+1} on $[0, \tau]$ by providing their values on subintervals of $[0, \tau]$. For $[a, b] \subseteq [0, \tau]$, let

$$\begin{aligned} \mu_i([a, b]) &= (\ell^i)^T (x' \Delta s(a, b) - x') \quad \forall i \in \{1, \dots, k\}, \\ \mu_{k+1}([a, b]) &= w^T (x' \Delta s(a, b) - x'). \end{aligned}$$

In particular, for any cyclic interval $[a, b]_{\odot} \subseteq [0, \tau]$, $\mu_i([a, b]_{\odot})$ and $\mu_{k+1}([a, b]_{\odot})$ measure the change in the i th length and weight, respectively, when replacing x' by $x' \Delta s(a, b)$. More generally, for k disjoint cyclic intervals $[a_1, b_1]_{\odot}, \dots, [a_k, b_k]_{\odot}$ and the vector

$$y = x \Delta s(a_1, b_1) \Delta \dots \Delta s(a_k, b_k), \tag{9}$$

we have

$$\begin{aligned} (\ell^i)^T y &= (\ell^i)^T x' + \mu_i(\cup_{j=1}^k [a_j, b_j]_{\odot}) \quad i \in \{1, \dots, k\}, \\ w^T y &= w^T x' + \mu_{k+1}(\cup_{j=1}^k [a_j, b_j]_{\odot}). \end{aligned} \tag{10}$$

Furthermore, it is easy to check that μ_i for $i \in \{1, \dots, k + 1\}$ are non-atomic signed measures. Hence, using Theorem 4.7 with $\lambda = 1 - \alpha$, there exist disjoint circular intervals $[a_1, b_1]_{\odot}, \dots, [a_k, b_k]_{\odot} \subseteq [0, \tau]$ such that

$$\mu_i(\cup_{j=1}^k [a_j, b_j]_{\odot}) = (1 - \alpha)\mu_i([0, \tau]) \quad \forall i \in \{1, \dots, k + 1\}.$$

Consider y being defined as in (9) for those intervals. Then combining the above equality with (10) we obtain

$$\begin{aligned} (\ell^i)^T y &= (\ell^i)^T x' + (1 - \alpha)\mu_i([0, \tau]) \\ &= (\ell^i)^T x' + (1 - \alpha)(\ell^i)^T (x'' - x') \\ &= (\ell^i)^T (\alpha x' + (1 - \alpha)x''), \quad \forall i \in \{1, \dots, k\}, \end{aligned}$$

and similaly,

$$\begin{aligned} w^T y &= w^T x' + (1 - \alpha)w([0, \tau]) \\ &= w^T x' + (1 - \alpha)w^T (x'' - x') \\ &= w^T (\alpha x' + (1 - \alpha)x''), \end{aligned}$$

as desired. □

Theorem 4.9 *A set of k cyclic intervals as described in Theorem 4.8 can be found efficiently.*

Proof Notice that we can efficiently guess $\lfloor a_j \rfloor$ and $\lfloor b_j \rfloor$ for $j \in \{1, \dots, k\}$ for a set of cyclic intervals $[a_1, b_1]_{\circlearrowleft}, \dots, [a_k, b_k]_{\circlearrowleft} \subseteq [0, \tau]$ guaranteed to exist by Theorem 4.8. More precisely, there are only $O(\tau^{2k})$ possibilities. Since τ is bounded by the number of vertices and k is a constant, this is polynomial. Futhermore, when the integer part of a_i, b_i for $j \in \{1, \dots, k\}$ is fixed, then the vector

$$y = x' \Delta s(a_1, b_1) \Delta \dots \Delta s(a_k, b_k)$$

is linear in the a_j 's and b_j 's. Therefore $(\ell^i)^T y$ for $i \in \{1, \dots, k\}$ and $w^T y$ are linear in the a_j 's and b_j 's. Hence, finding values of the a_j 's and b_j 's, with predetermined integer parts, such that the resulting y satisfies (6) and (7), reduces to solving a constant-size linear problem, which can be done in constant time. □

Summarizing the above discussion, we get the following.

Corollary 4.10 *There is an efficient procedure $\text{MERGE}(\alpha, x', 1 - \alpha, x'')$, that for $\alpha \in [0, 1]$, and $x', x'' \in \mathcal{M}$, outputs a matching y satisfying (6) and (7).*

This finishes the description of the algorithm in Fig. 4 and implies together with Theorem 4.5 the following.

Corollary 4.11 *The algorithm described in Fig. 4 is an efficient procedure that returns a feasible matching y for k -BUDGETED MATCHING with $w^T y \geq w(\text{OPT}) - \frac{(k+3)k^2}{k+1} w_{\max}$.*

To obtain a PTAS for k -BUDGETED MATCHING, it suffices to guess the $\lceil \frac{(k+3)k^2}{(k+1)\epsilon} \rceil$ heaviest edges of an optimal solution before applying the algorithm in Fig. 4.

Theorem 4.12 *Let $\epsilon > 0$. The following procedure is an efficient $(1 - \epsilon)$ -approximation for k -BUDGETED MATCHING.*

1. *Guess the $\lceil \frac{(k+3)k^2}{(k+1)\epsilon} \rceil$ heaviest edges E_H of an optimal solution, and reduce the problem correspondingly.*
2. *Apply the algorithm in Fig. 4 to the reduced problem to get a matching E_L .*
3. *Return $E_H \cup E_L$.*

Proof Efficiency and feasibility is obvious. We have to show $w(E_H \cup E_L) \geq (1 - \epsilon)w(OPT)$. Let w'_{\max} be the maximum weight of an edge in the reduced problem. Since, to build the reduced problem, we remove all edges of weight strictly larger than the smallest weight in E_H , we have

$$w'_{\max} \leq w(e) \quad \forall e \in E_H. \tag{11}$$

Furthermore, let OPT' be an optimal solution in the reduced problem. Clearly

$$w(OPT) = w(E_H) + w(OPT'). \tag{12}$$

Hence,

$$\begin{aligned} w(E_H \cup E_L) &= w(E_H) + w(E_L) \stackrel{Cor. (4.11)}{\geq} w(E_H) + w(OPT') - \frac{(k+3)k^2}{k+1}w'_{\max} \\ &\stackrel{(11)}{\geq} w(E_H) + w(OPT') - \epsilon w(E_H) \geq (1 - \epsilon)(w(E_H) + w(OPT')) \\ &\stackrel{(12)}{=} (1 - \epsilon)w(OPT). \end{aligned}$$

□

For completeness, the following section shows how Theorem 4.7 is obtained by following the proof of Stromquist and Woodall [40, Theorem 1].

4.3.2 A generalization of a theorem by Stromquist and Woodall

To prove Theorem 4.7 we use the same proof technique as [40] with the only difference that we replace the classical version of the Ham Sandwich Theorem by the following generalization.

Theorem 4.13 (Generalized Ham Sandwich Theorem) *Suppose we are given d signed measures μ_1, \dots, μ_d on \mathbb{R}^d that vanish on any hyperplane, then there exists a (possible degenerate) halfspace H in \mathbb{R}^d such that*

$$\mu_i(H) = \frac{1}{2}\mu_i(\mathbb{R}^d) \quad \forall i \in \{1, \dots, d\},$$

where a degenerate halfspace is either \emptyset or \mathbb{R}^d .

Theorem 4.13 with the additional requirement of the μ_i being absolutely continuous was proven in [2]. Their proof follows a standard approach used in [41] to show the classic Ham Sandwich Theorem for (unsigned) measures, which is based on the Borsuk-Ulam Theorem [3]. However, the proof presented in [2] actually shows Theorem 4.13, and was only stated in a slightly weaker form requiring absolute continuity of the measures.

For completeness, we replicate the proof of Stromquist and Woodall, in a slightly generalized version that uses Theorem 4.13, to show Theorem 4.7.

Proof (of Theorem 4.7, following [40]) We denote by $A \subseteq [0, 1]$ the values of $\lambda \in [0, 1]$ for which the statement is true. We have to show that $A = [0, 1]$. This will follow from the following four statements:

- (a) $1 \in A$,
- (b) $\lambda \in A \Rightarrow (1 - \lambda) \in A$,
- (c) $\lambda \in A \Rightarrow \frac{1}{2}\lambda \in A$,
- (d) A is closed.

Statement (a) follows by choosing $K_1 = [0, 1]$, and (b) by setting $K_{1-\lambda} = [0, 1] \setminus K_\lambda$. Furthermore, (d) follows by observing that the space of the unions of $d - 1$ intervals is compact in a suitable topology, hence, for any sequence of λ 's in A , the corresponding K_λ 's converge. It remains to prove (c).

Fix $\lambda \in A$ and let $K_\lambda \subseteq [0, 1]$ be a union of $d - 1$ cyclic intervals of $[0, 1]$ such that $\mu^i(K_\lambda) = \lambda\mu^i(\mathbb{R}^d)$ for $i \in \{1, \dots, d\}$. If $K_\lambda \neq [0, 1]$, we can assume that the origin is not contained in K_λ , for otherwise, we can simply reparameterize the circular interval $[0, 1]$. Let $f : [0, 1) \rightarrow \mathbb{R}^d$ be defined by

$$f(t) = (t, t^2, \dots, t^d).$$

We will apply the generalized Ham Sandwich Theorem 4.13 to a family of measures with support included in the image of f . More precisely, we define d measures ν_1, \dots, ν_d on \mathbb{R}^d by

$$\nu_i(B) = \mu_i(f^{-1}(B) \cap K_\lambda) \quad \forall i \in \{1, \dots, d\}, B \subseteq \mathbb{R}^d \text{ Borel set.}$$

Notice that $\nu_i(\mathbb{R}^d) = \mu_i(K_\lambda) = \lambda\mu_i(\mathbb{R}^d)$, and ν_i are signed measures. Furthermore, for any hyperplane X in \mathcal{R}^d , $f^{-1}(X)$ is a set of size at most d , and since the measures μ_i are non-atomic, this implies $\nu_i(X) = 0$. Hence, we can apply the Generalized Ham Sandwich Theorem 4.13 to obtain that there exists a halfspace $H \subset \mathbb{R}^d$ such that $\nu_i(H) = \frac{\lambda}{2}\mu_i(\mathbb{R}^d)$ for $i \in \{1, \dots, d\}$. Let X be the hyperplane that is the boundary of H . Consider the complementary halfspace \bar{H} of H , i.e. $\bar{H} \cap H = X$.

Since $\nu_i(\mathbb{R}^d) = \lambda\mu_i(\mathbb{R}^d)$ for $i \in \{1, \dots, d\}$, we also have $\nu_i(\bar{H}) = \frac{\lambda}{2}\mu_i(\mathbb{R}^d)$ for $i \in \{1, \dots, d\}$. We will show that one can choose $K_{\frac{\lambda}{2}}$ to be either $K^H := f^{-1}(H) \cap K_\lambda$ or $K^{\bar{H}} := f^{-1}(\bar{H}) \cap K_\lambda$. Notice that $\nu_i(H) = \nu_i(\bar{H}) = \frac{\lambda}{2}\mu_i(\mathbb{R}^d)$ for $i \in \{1, \dots, d\}$ can be rephrased as

$$\mu_i(K^H) = \mu_i(K^{\bar{H}}) = \frac{\lambda}{2}\mu_i(\mathbb{R}^d) \quad \forall i \in \{1, \dots, d\}.$$

Hence, both K^H and $K^{\bar{H}}$ have the desired mass with respect to the measures μ_1, \dots, μ_d . It remains to show that one of them is a union of at most d intervals.

Since $|f^{-1}(X) \cap K_\lambda| \leq d$, the $\leq d$ intervals of K_λ are hit in at most d points by $f^{-1}(X)$, thus subdividing K_λ into at most $2d + 1$ intervals. These intervals are partitioned by K^H and $K^{\bar{H}}$. Hence, indeed, either K^H or $K^{\bar{H}}$ is the union of at most d intervals. □

4.4 Conclusions

We presented approximation algorithm for a variety of multi-budgeted problems. A main drawback of the procedures presented here is the quite high dependence of the running time on the (constant) number of budgets. It is an interesting question whether FPTASs can be designed for problems presented in this paper, as for example k -BUDGETED MATCHING. The main reason why the presented procedures do not lead to FPTASs is the use of guessing steps. This includes the guessing of a number of element that depends on k , which is a typical preprocessing step in our algorithms. Furthermore, for the k -BUDGETED MATCHING problem, we use a further guessing step in the proof of Theorem 4.9 to find a good set of cyclic intervals. It could be of independent interest to find a more efficient algorithm that returns the intervals claimed by Theorem 4.8. This may be approached by seeking a constructive proof of the Generalized Ham Sandwich Theorem (Theorem 4.13) for the special setting that we need in the proof of Theorem 4.7.

Acknowledgments We are grateful to Christian Reitwießner and Maximilian Witek for pointing us to the paper of Stromquist and Woodall [42]. Furthermore, we are thankful to the two anonymous referees, whose comments and suggestions considerably improved the presentation of this paper. F. Grandoni: Partially supported by the ERC Starting Grant NEWNET 279352. R. Ravi: Supported in part by NSF grant CCF-0728841. R. Zenklusen: Supported by Swiss National Science Foundation Grant PBEZP2-129524, by NSF Grants CCF-1115849 and CCF-0829878, and by ONR Grants N00014-11-1-0053 and N00014-09-1-0326.

References

1. Aggarwal, V., Aneja, Y.P., Nair, K.P.K.: Minimal spanning tree subject to a side constraint. *Comput. Oper. Res.* **9**, 287–296 (1982)
2. Akopyan, A., Karasev, R.: Cutting the same fraction of several measures. *Discret. Comput. Geom.* **49**(2), 402–410 (2013)
3. Borsuk, U.: Drei Sätze über die n -dimensionale euklidische Sphäre. *Fundam. Math.* **20**, 170–190 (1933)
4. Bansal, N., Khandekar, R., Nagarajan, V.: Additive Guarantees for Degree Bounded Directed Network Design. In: *STOC*, pages 769–778 (2008)
5. Barahona, F., Pulleyblank, W.R.: Exact arborescences, matchings and cycles. *Discret. Appl. Math.* **16**(2), 91–99 (1987)
6. Berger, A., Bonifaci, V., Grandoni, F., Schäfer, G.: Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Math. Program.* **128**(1–2), 355–372 (2011)
7. Bilu, V., Goyal, V., Ravi, R., Singh M.: On the crossing spanning tree problem. In *APPROX-RANDOM*, pages 51–64 (2004)
8. Byrka, J., Grandoni, F., Rothvoss, T., Sanità L.: An Improved LP-Based Approximation for Steiner Tree. In: *STOC*, pages 583–592 (2010)
9. Camerini, P., Galbiate, G., Maffioli, F.: Random pseudo-polynomial algorithms for exact matroid problems. *J. Algorithms* **13**, 258–273 (1992)
10. Chekuri, C., Vondrák, J., Zenklusen, R.: Dependent randomized rounding for matroid polytopes and applications. In: *FOCS*, pages 575–584 (2010)
11. Chekuri, C., Vondrák, J., Zenklusen, R.: Multi-budgeted matchings and matroid intersection via dependent rounding. In: *SODA*, pages 1080–1097 (2011)
12. Correa, J.R., Levin, A.: Monotone covering problems with an additional covering constraint. *Math. Oper. Res.* **34**(1), 238–248 (2009)
13. Cunningham, W.H.: Testing membership in matroid polyhedra. *J. Comb.Theory B* **36**(2), 161–188 (1984)
14. Goemans, M.X.: Minimum Bounded Degree Spanning Trees. In: *FOCS*, pages 273–282 (2006)

15. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H Freeman, San Francisco, CA (1979)
16. Goemans, M.X., Olver, N., Rothvoss, T., Zenklusen, R.: Matroids and Integrality Gaps for Hypergraphic Steiner Tree Relaxations. In: *STOC*, pages 1161–1175 (2012)
17. Grandoni, F., Ravi, R., Singh, M.: Iterative rounding for multi-objective optimization problems. In: *ESA*, pages 95–106 (2009)
18. Grandoni, F., Zenklusen R., Approximation Schemes for Multi-Budgeted Independence Systems. In: *ESA*, pages 536–548 (2010)
19. Hurkens, C.A., Lovász, L., Schrijver, A., Tardos, E.: *How to Tidy Up Your Set System*. Combinatorics, North-Holland (1988)
20. Hassin, R.: Approximation schemes for the restricted shortest path problem. *Math. Oper. Res.* **17**(1), 36–42 (1992)
21. Hassin, R., Levin, A.: An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM J. Comput.* **33**(2), 261–268 (2004)
22. Jain, K.: A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica* **21**, 39–60 (2001)
23. Korte, B., Vygen, J.: *Combinatorial optimization*. Springer, Berlin (2008)
24. Lau, L.C., Naor, S., Salavatipour, M., Singh, M.: Survivable network design with degree or order constraints. In: *STOC*, pages 651–660 (2007)
25. Lau, L.C., Singh, M.: Additive approximation for bounded degree survivable network design. In: *STOC*, pages 759–768 (2008)
26. de Longueville, M., Zivaljevic, R.T.: Splitting multidimensional necklaces. *Adv. Math.* **218**(3), 926–939 (2008)
27. Lorenz, D., Raz, D.: A simple efficient approximation scheme for the restricted shortest paths problem. *Oper. Res. Lett.* **28**, 213–219 (2001)
28. Marathe, M.V., Ravi, R., Sundaram, R., Ravi, S.S., Rosenkrantz, D.J., Hunt, H.B.: III. Bicriteria network design problems. In: *ICALP*, pages 487–498 (1995)
29. Motwani, R., Raghavan, P.: *Random Algorithms*. Cambridge University Press, Cambridge (1995)
30. Mulmuley, K., Vazirani, U., Vazirani, V.: Matching is as easy as matrix inversion. *Combinatorica* **7**(1), 101–104 (1987)
31. Munkres, J.R.: *Topology*, second edition, Prentice Hall, UK (2000)
32. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: *FOCS*, pages 86–92 (2000)
33. Ravi, R.: Rapid rumor ramification: approximating the minimum broadcast time. In: *FOCS*, pages 202–213 (1994)
34. Ravi, R.: Matching based augmentations for approximating connectivity problems. Invited Lecture. In: *LATIN*, pages 13–24 (2006)
35. Ravi, R., Goemans, M.X.: The constrained minimum spanning tree problem (extended abstract). In: *SWAT*, pages 66–75 (1996)
36. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Hunt, H.B.: Many birds with one stone: Multi-objective approximation algorithms. In: *STOC*, pages 438–447 (1993)
37. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley, London (1998)
38. Schrijver, A.: *Combinatorial Optimization. Polyhedra and Efficiency*. Springer, Berlin (2003)
39. Singh, M., Lau, L.C.: Approximating minimum bounded degree spanning trees to within one of optimal. In: *STOC*, pages 661–670 (2007)
40. Stromquist, W., Woodall, D.R.: Sets on which several measures agree. *J. Math. Anal. Appl.* **108**, 241–248 (1985)
41. Stone, A.H., Tukey, J.W.: Generalized “sandwich” theorems. *Duke Mathematical Journal* **9**, 356–359 (1942)
42. Warburton, A.: Approximation of Pareto optima in multiple-objective, shortest path problems. *Oper. Res.* **35**, 70–79 (1987)
43. Zenklusen, R.: Matroidal Degree-Bounded Minimum Spanning Trees. In: *SODA*, pages 1512–1521 (2012)