An EPTAS for Budgeted Matching and Budgeted Matroid Intersection via Representative Sets

Ilan Doron-Arad
Computer Science Department, Technion, Haifa, Israel

Ariel Kulik
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Hadas Shachnai
Computer Science Department, Technion, Haifa, Israel

Abstract

We study the budgeted versions of the well known matching and matroid intersection problems. While both problems admit a *polynomial-time approximation scheme (PTAS)* [Berger et al. (Math. Programming, 2011), Chekuri, Vondrák and Zenklusen (SODA 2011)], it has been an intriguing open question whether these problems admit a *fully* PTAS (FPTAS), or even an *efficient* PTAS (EPTAS).

In this paper we answer the second part of this question affirmatively, by presenting an EPTAS for budgeted matching and budgeted matroid intersection. A main component of our scheme is a construction of *representative sets* for desired solutions, whose cardinality depends only on ε, the accuracy parameter. Thus, enumerating over solutions within a representative set leads to an EPTAS. This crucially distinguishes our algorithms from previous approaches, which rely on *exhaustive* enumeration over the solution set.

2012 ACM Subject Classification Theory of computation

Keywords and phrases budgeted matching, budgeted matroid intersection, efficient polynomial-time approximation scheme

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.49

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2209.04654 [5]

Funding *Ariel Kulik*: Research supported by the European Reseach Concil (ERC) consolidator grant no. 725978 SYSTEMATICGRAPH.

Acknowledgements We thank an anonymous reviewer for pointing us to the work of Huang and Ward [9], and for other helpful comments and suggestions.

1 Introduction

A wide range of NP-hard combinatorial optimization problems can be formulated as follows. We are given a ground set E and a family $\mathcal{M}$ of subsets of E called the *feasible sets*. The elements in the ground set are associated with a cost function $c : E \to \mathbb{R}_{\geq 0}$ and a profit function $p : E \to \mathbb{R}$, and we are also given a budget $\beta \in \mathbb{R}_{\geq 0}$. A *solution* is a feasible set $S \in \mathcal{M}$ of bounded cost $c(S) \leq \beta$.[1] Generally, the goal is to find a solution S of maximum profit, that is:

$$\max p(S) \text{ s.t. } S \in \mathcal{M}, c(S) \leq \beta. \tag{1}$$

[1] For a function $f : A \to \mathbb{R}$ and a subset of elements $C \subseteq A$, we define $f(C) = \sum_{e \in C} f(e)$.

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 49; pp. 49:1–49:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Notable examples include shortest weight-constrained path [7], constrained minimum spanning trees [16], and knapsack with a conflict graph [15]. In this work, we focus on two prominent problems which can be formulated as (1).

In the *budgeted matching (BM)* problem we are given an undirected graph $G = (V, E)$, profit and cost functions on the edges $p, c : E \to \mathbb{R}_{\geq 0}$, and a budget $\beta \in \mathbb{R}_{\geq 0}$. A *solution* is a *matching* $S \subseteq E$ in G such that $c(S) \leq \beta$. The goal is to find a solution S such that the total profit $p(S)$ is maximized. Observe that BM can be formulated using (1), by letting $\mathcal{M}$ be the set of matchings in G.

In the *budgeted matroid intersection (BI)* problem we are given two matroids $(E, \mathcal{I}_1)$ and $(E, \mathcal{I}_2)$ over a ground set E, profit and cost functions on the elements $p, c : E \to \mathbb{R}_{\geq 0}$, and a budget $\beta \in \mathbb{R}_{\geq 0}$. Each matroid is given by a membership oracle. A *solution* is a *common independent set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ such that $c(S) \leq \beta$; the goal is to find a solution S of maximum total profit $p(S)$. The formulation of BI as (1) follows by defining the feasible sets as all common independent sets $\mathcal{M} = \mathcal{I}_1 \cap \mathcal{I}_2$.

Let $\text{OPT}(I)$ be the value of an optimal solution for an instance I of a maximization problem Π. For $\alpha \in (0, 1]$, we say that $\mathcal{A}$ is an α-approximation algorithm for Π if, for any instance I of Π, $\mathcal{A}$ outputs a solution of value at least $\alpha \cdot \text{OPT}(I)$. A *polynomial-time approximation scheme* (PTAS) for Π is a family of algorithms $(A_\varepsilon)_{\varepsilon>0}$ such that, for any $\varepsilon > 0$, A_ε is a polynomial-time $(1-\varepsilon)$-approximation algorithm for Π. As ε gets smaller, a running time of the form $n^{\Theta\left(\frac{1}{\varepsilon}\right)}$ for a PTAS may become prohibitively large and thus impractical; therefore, it is natural to seek approximation schemes with better running times. Two families of such schemes have been extensively studied: an *efficient PTAS* (EPTAS) is a PTAS $(A_\varepsilon)_{\varepsilon>0}$ whose running time is of the form $f\left(\frac{1}{\varepsilon}\right) \cdot n^{O(1)}$, where f is an arbitrary computable function, and n is the bit-length encoding size of the input instance. In a *fully PTAS* (FPTAS) the running time of A_ε is of the form $\left(\frac{n}{\varepsilon}\right)^{O(1)}$. For comprehensive surveys on approximation schemes see, e.g., [18, 19].

The state of the art for BM and BI is a PTAS developed by Berger et al. [1]. Similar results for both problems follow from a later work of Chekuri et al. [3] for the multi-budgeted variants of BM and BI. The running times of the above schemes are dominated by exhaustive enumeration which finds a set of $\Theta\left(\frac{1}{\varepsilon}\right)$ elements of highest profits in the solution. In this paper we optimize the enumeration procedure by substantially reducing the size of the domain over which we seek an efficient solution. Our main results are the following.

▶ **Theorem 1.** *There is an* EPTAS *for the budgeted matching problem.*

▶ **Theorem 2.** *There is an* EPTAS *for the budgeted matroid intersection problem.*

1.1 Related Work

BM and BI are immediate generalizations of the classic 0/1-knapsack problem. While the knapsack problem is known to be NP-hard, it admits an FPTAS. This raises a natural question whether BM and BI admit an FPTAS as well. The papers [1, 3] along with our results can be viewed as first steps towards answering this question.

Berger et al. [1] developed the first PTAS for BM and BI. Their approach includes an elegant combinatorial algorithm for *patching* two solutions for the *Lagrangian relaxation* of the underlying problem (i.e., BM or BI); one solution is feasible but has small profit, while the other solution has high profit but is infeasible. The scheme of [1] enumerates over solutions containing only high profit elements and uses the combinatorial algorithm to add low profit elements. This process may result in losing (twice) the profit of a low profit element, leading to a PTAS.

Chekuri et al. [3] developed a PTAS for multi-budgeted matching and a randomized PTAS for multi-budgeted matroid intersection; these are variants of BM and BI, respectively, in which the costs are d-dimensional, for some constant $d \geq 2$. They incorporate a non-trivial martingale based analysis to derive the results, along with enumeration to facilitate the selection of profitable elements for the solution. The paper [3] generalizes a previous result of Grandoni and Zenklusen [8], who obtained a PTAS for multi-budgeted matching and multi-budgeted matroid intersection in *representable matroids*.[2] For $d \geq 2$, the multi-budgeted variants of BM and BI generalize the two-dimensional knapsack problem, and thus do not admit an EPTAS unless W[1] = FPT [11].

An evidence for the difficulty of attaining an FPTAS for BM comes from the *exact* variant of the problem. In this setting, we are given a graph $G = (V, E)$, a cost function $c : E \to \mathbb{R}_{\geq 0}$, and a *target* $B \in \mathbb{R}_{\geq 0}$; the goal is to find a perfect matching $S \subseteq E$ with exact specified cost $c(S) = B$. There is a randomized pseudo-polynomial time algorithm for exact matching [13]. On the other hand, it is a long standing open question whether exact matching admits a deterministic pseudo-polynomial time algorithm [14]. Interestingly, as noted by Berger et al. [1], a deterministic FPTAS for BM would give an affirmative answer also for the latter question. A deterministic FPTAS for BI would have similar implications for the *exact* matroid intersection problem, which admits a randomized (but not a deterministic) pseudo-polynomial time algorithm for linear matroids [2]. While the above does not rule out the existence of an FPTAS for BM or BI, it indicates that improving our results from EPTAS to FPTAS might be a difficult task.

For the budgeted matroid independent set (i.e., the special case of BI with two identical matroids), the paper [6] gives an EPTAS using *representative sets* to enhance enumeration over elements of high profits. Their scheme exploits integrality properties of matroid polytopes under budget constraints (introduced in [8]) to efficiently combine elements of low profit into the solution.

1.2 Contribution and Techniques

Given an instance I of BM or BI, we say that an element e is *profitable* if $p(e) > \varepsilon \cdot \mathrm{OPT}(I)$; otherwise, e is *non-profitable*. The scheme for BM and BI of Berger et al. [1] distinguishes between profitable and non-profitable elements. In the main loop, the algorithm enumerates over all potential solutions containing only profitable elements.[3] Each solution is extended to include non-profitable elements using a combinatorial algorithm. The algorithm outputs a solution of highest profit. Overall, this process may lose at most twice the profit of a non-profitable element compared to the optimum, effectively preserving the approximation guarantee; however, an exhaustive enumeration over the profitable elements renders the running time $n^{\Omega\left(\frac{1}{\varepsilon}\right)}$. In stark contrast, in this paper we introduce a new approach which enhances the enumeration over profitable elements, leading to an EPTAS.

We restrict the enumeration to only a small subset of elements called *representative set*; that is, a subset of elements $R \subseteq E$ satisfying the following property: there is a solution S such that the profitable elements in S are a subset of R, and the profit of S is at least $(1-O(\varepsilon)) \cdot \mathrm{OPT}(I)$. If one finds efficiently a representative set R of cardinality $|R| \leq f\left(\frac{1}{\varepsilon}\right)$ for some computable function f, obtaining an EPTAS is straightforward based on the approach of [1].

[2] Representable matroids are also known as *linear matroids*.
[3] A similar technique is used also by Chekuri et al. [3].

Our scheme generalizes the *representative set* framework in [6], developed originally for budgeted matroid independent set. In [6], a representative set is a basis of minimum cost of a matroid, which can be found using a greedy algorithm. Alas, a greedy analogue for the setting of matching and matroid intersection fails; we give an example in Figure 1.[4] Hence, we take a different approach. Our main technical contribution is in the construction of representative sets for each of our problems.

For BM we design a surprisingly simple algorithm which finds a representative set using a union of multiple matchings. To this end, we partition the edges in G into *profit classes* such that each profit class contains edges of *similar* profits. We then use the greedy approach to repeatedly find in each profit class a union of disjoint matchings, where each matching has a bounded cardinality and is greedily selected to minimize cost. Intuitively, to show that the above yields a representative set, consider a profitable edge e in some optimal solution. Suppose that e is not chosen in our union of matchings, then we consider two cases. If each matching selected in the profit class of e contains an edge that is adjacent to (i.e., shares a vertex with) e, we show that at least one of these edges can be exchanged with e; otherwise, there exists a matching with no edge adjacent to e. In this case, we show that our greedy selection guarantees the existence of an edge in this matching which can be exchanged with e, implying the above is a representative set (see the details in Section 4).

For BI, we design a recursive algorithm that relies on an *asymmetric interpretation* of the two given matroids. We have learnt recently that a similar and more powerful construction was already proposed in [9]; we include the full details for completeness. In each recursive call of the algorithm, we are given an independent set $S \in \mathcal{I}_1$. The algorithm adds to the constructed representative set a minimum cost basis B_S of the second matroid $(E, \mathcal{I}_2)$, with the crucial restriction that any element $e \in B_S$ must satisfy $S \cup \{e\} \in \mathcal{I}_1$. Succeeding recursive calls will then use the set $S \cup \{e\}$, for every $e \in B_S$. Thus, we limit the search space to $\mathcal{I}_1$, while bases are constructed w.r.t. $\mathcal{I}_2$. To show that the algorithm yields a representative set, consider a profitable element f in an optimal solution. We construct a sequence of elements which are independent w.r.t. $\mathcal{I}_1$ and can be exchanged with f w.r.t. $\mathcal{I}_2$. Using matroid properties we show that one of these elements can be exchanged with f w.r.t. both matroids (see the details in Section 5).

Interestingly, our framework for solving BM and BI (presented in Section 3) can be extended to solve other problems formulated as (1) which possess similar *exchange properties*. We elaborate on that in Section 6.

Organization of the paper. In Section 2 we give some definitions and notation. Section 3 presents our framework that yields an EPTAS for each of the problems. In Sections 4 and 5 we describe the algorithms for constructing representative sets for BM and BI, respectively. We conclude in Section 6 with a summary and some directions for future work. Due to space constraints, some of the proofs are given in the full version of the paper [5].

2 Preliminaries

For simplicity of the notation, for any set A and an element e, we use $A + e$ and $A - e$ to denote $A \cup \{e\}$ and $A \setminus \{e\}$, respectively. Also, for any $k \in \mathbb{R}$ let $[k] = \{1, 2, \ldots, \lfloor k \rfloor\}$. For a function $f : A \to \mathbb{R}_{\geq 0}$ and a subset of elements $C \subseteq A$, let $f|_C : C \to \mathbb{R}_{\geq 0}$ be the *restriction* of f to C, such that $\forall e \in C : f|_C(e) = f(e)$.

[4] The example becomes clear once the reader is familiar with the definitions given in Section 3.

2.1 Matching and Matroids

Given an undirected graph $G = (V, E)$, a *matching* of G is a subset of edges $M \subseteq E$ such that each vertex appears as an endpoint in at most one edge in M, i.e., for all $v \in V$ it holds that $|\{\{u, v\} \in M \mid u \in V\}| \leq 1$. We denote by $V(M) = \{v \in V \mid \exists u \in V \text{ s.t. } \{u, v\} \in M\}$ the set of endpoints of a matching M of G.

Let E be a finite ground set and $\mathcal{I} \subseteq 2^E$ a non-empty set containing subsets of E called the *independent sets* of E. Then $\mathcal{M} = (E, \mathcal{I})$ is a *matroid* if it satisfies the following.

1. (Hereditary Property) For all $A \in \mathcal{I}$ and $B \subseteq A$, it holds that $B \in \mathcal{I}$.
2. (Exchange Property) For any $A, B \in \mathcal{I}$ where $|A| > |B|$, there is $e \in A \setminus B$ such that $B + e \in \mathcal{I}$.

A *basis* of a matroid $\mathcal{G} = (E, \mathcal{I})$ is an independent set $B \in \mathcal{I}$ such that for all $e \in E \setminus B$ it holds that $B + e \notin \mathcal{I}$. Given a cost function $c : E \to \mathbb{R}_{\geq 0}$, we say that a basis B of $\mathcal{G}$ is a *minimum* basis of $\mathcal{G}$ w.r.t. c if, for any basis A of $\mathcal{G}$ it holds that $c(B) \leq c(A)$. A minimum basis of $\mathcal{G}$ w.r.t. c can be easily constructed in polynomial-time using a greedy approach (see, e.g., [4]). In the following we define several matroid operations. Note that the structures resulting from the operations outlined in Definition 3 are matroids (see, e.g., [17]).

▶ **Definition 3.** *Let $\mathcal{G} = (E, \mathcal{I})$ be a matroid.*

1. *(restriction) For any $F \subseteq E$ define $\mathcal{I}_{\cap F} = \{A \in \mathcal{I} \mid A \subseteq F\}$ and $\mathcal{G} \cap F = (F, \mathcal{I}_{\cap F})$.*
2. *(thinning) For any $F \in \mathcal{I}$ define $\mathcal{I}/F = \{A \subseteq E \setminus F \mid A \cup F \in \mathcal{I}\}$ and $\mathcal{G}/F = (E \setminus F, \mathcal{I}/F)$.*[5]
3. *(truncation) For any $q \in \mathbb{N}$ define $\mathcal{I}_{\leq q} = \{A \in \mathcal{I} \mid |A| \leq q\}$ and $[\mathcal{G}]_{\leq q} = (E, \mathcal{I}_{\leq q})$.*

2.2 Instance Definition

We give a unified definition for instances of budgeted matching and budgeted matroid intersection. Given a ground set E of elements, we say that $\mathcal{C}$ is a *constraint* of E if one of the following holds.

- $\mathcal{C} = (V, E)$ is a *matching constraint*, where $\mathcal{C}$ is an undirected graph. Let $\mathcal{M}(\mathcal{C}) = \{M \subseteq E \mid M \text{ is a matching in } \mathcal{C}\}$ be the *feasible sets* of $\mathcal{C}$. Given a subset of edges $F \subseteq E$, let $E/F = \{\{u, v\} \in E \mid u, v \notin V(F)\}$ be the *thinning* of F on E, and let $\mathcal{C}/F = (V, E/F)$ be the *thinning* of F on $\mathcal{C}$.
- $\mathcal{C} = (\mathcal{I}_1, \mathcal{I}_2)$ is a *matroid intersection constraint*, where $(E, \mathcal{I}_1)$ and $(E, \mathcal{I}_2)$ are matroids. Throughout this paper, we assume that each of the matroids is given by an independence oracle. That is, determining whether $F \subseteq E$ belongs to $\mathcal{I}_1$ or to $\mathcal{I}_2$ requires a single call to the corresponding oracle of $\mathcal{I}_1$ or $\mathcal{I}_2$, respectively. Let $\mathcal{M}(\mathcal{C}) = \mathcal{I}_1 \cap \mathcal{I}_2$ be the collection of *feasible sets* of $\mathcal{C}$. In addition, given some $F \subseteq E$, let $\mathcal{C}/F = (\mathcal{I}_1/F, \mathcal{I}_2/F)$ be the *thinning* of F on $\mathcal{C}$. We say that $\mathcal{C}$ is a *single matroid constraint* if $\mathcal{I}_1 = \mathcal{I}_2$

When understood from the context, we simply use $\mathcal{M} = \mathcal{M}(\mathcal{C})$. Define an instance of the *budgeted constrained (BC)* problem as a tuple $I = (E, \mathcal{C}, c, p, \beta)$, where E is a ground set of elements, $\mathcal{C}$ is a constraint of E, $c : E \to \mathbb{R}_{\geq 0}$ is a cost function, $p : E \to \mathbb{R}_{\geq 0}$ is a profit function, and $\beta \in \mathbb{R}_{\geq 0}$ is a budget. If $\mathcal{C}$ is a matching constraint then I is a BM instance; otherwise, I is a BI instance. A *solution* of I is a feasible set $S \in \mathcal{M}(\mathcal{C})$ such that $c(S) \leq \beta$. The objective is to find a solution S of I such that $p(S)$ is maximized. Let $|I|$ denote the encoding size of a BC instance I, and poly$(|I|)$ be a polynomial size in $|I|$.

[5] Thinning is generally known as contraction; we use the term thinning to avoid confusion with edge contraction in graphs.

3 The Algorithm

In this section we present an EPTAS for the BC problem. Our first step is to determine the set of *profitable* elements in the constructed solution.[6] To this end, we generalize the *representative set* notion of [6] to the setting of BC. Our scheme relies on initially finding a set of profitable elements of small cardinality, from which the most profitable elements are selected for the solution using enumeration. Then, *non-profitable* elements are added to the solution using techniques of [1].

For the remainder of this section, fix a BC instance $I = (E, \mathcal{C}, c, p, \beta)$ and an error parameter $0 < \varepsilon < \frac{1}{2}$. Let $H(I, \varepsilon) = \{e \in E \mid p(e) > \varepsilon \cdot \text{OPT}(I)\}$ be the set of *profitable* elements in I, and $E \setminus H(I, \varepsilon)$ the set of *non-profitable* elements; when understood from the context, we use $H = H(I, \varepsilon)$. Now, a representative set is a subset of elements which contains the profitable elements of an *almost* optimal solution. Formally,

▶ **Definition 4.** *Let $I = (E, \mathcal{C}, c, p, \beta)$ be a* BC *instance, $0 < \varepsilon < \frac{1}{2}$ and $R \subseteq E$. We say that R is a* representative set *of I and ε if there is a solution S of I such that the following holds.*
1. $S \cap H \subseteq R$.
2. $p(S) \geq (1 - 4\varepsilon) \cdot \text{OPT}(I)$.

The work of [6] laid the foundations for the following notions of *replacements* and *strict representative sets (SRS)*, for the special case of BC where $\mathcal{C}$ is a single matroid constraint. Below we generalize the definitions of replacements and SRS.

Intuitively, a replacement of a solution S for I of bounded cardinality is another solution for I which preserves the attributes of the profitable elements in S (i.e., $S \cap H$). In particular, the profit of the replacement is close to $p(S \cap H)$, whereas the cost and the number of profitable elements can only be smaller. An SRS is a subset of elements containing a replacement for any solution for I of bounded cardinality.

The formal definitions of replacement and SRS for general BC instances are given in Definitions 5 and 6, respectively. Let $q(\varepsilon) = \left\lceil \varepsilon^{-\varepsilon^{-1}} \right\rceil$, and $\mathcal{M}_{\leq q(\varepsilon)} = \{A \in \mathcal{M} \mid |A| \leq q(\varepsilon)\}$ be all *bounded feasible sets* of $\mathcal{C}$ and ε. Recall that we use $\mathcal{M} = \mathcal{M}(\mathcal{C})$ for the feasible sets of $\mathcal{C}$; similar simplification in notation is used also for bounded feasible sets.

▶ **Definition 5.** *Given a* BC *instance $I = (E, \mathcal{C}, c, p, \beta), 0 < \varepsilon < \frac{1}{2}$, $S \in \mathcal{M}_{\leq q(\varepsilon)}$, and $Z_S \subseteq E$, we say that Z_S is a* replacement *of S for I and ε if the following holds:*
1. $(S \setminus H) \cup Z_S \in \mathcal{M}_{\leq q(\varepsilon)}$.
2. $c(Z_S) \leq c(S \cap H)$.
3. $p\left((S \setminus H) \cup Z_S\right) \geq (1 - \varepsilon) \cdot p(S)$.
4. $|Z_S| \leq |S \cap H|$.

▶ **Definition 6.** *Given a* BC *instance $I = (E, \mathcal{C}, c, p, \beta), 0 < \varepsilon < \frac{1}{2}$, and $R \subseteq E$, we say that R is a* strict representative set (SRS) *of I and ε if, for any $S \in \mathcal{M}_{\leq q(\varepsilon)}$, there is a replacement $Z_S \subseteq R$ of S for I and ε.*

Observe that given any solution S of I such that $|S| \leq q(\varepsilon)$, it holds that $S \cap H$ is a replacement of S; also, E is an SRS. In the next result, we demonstrate the power of SRS in solving BC. Specifically, we show that any SRS $R \subseteq E$ is also a representative set. Hence, using enumeration on subsets of R we can find a subset of elements that can be extended by only non-profitable elements to an *almost* optimal solution (see Algorithm 2).

[6] A similar approach is used, e.g., in [8, 1, 6].

▶ **Lemma 7.** *Let $I = (E, \mathcal{C}, c, p, \beta)$ be a* BC *instance, let $0 < \varepsilon < \frac{1}{2}$, and let R be an* SRS *of I and ε. Then R is a representative set of I and ε.*

The proof of Lemma 7 is given in [5]. We proceed to construct an SRS whose cardinality depends only on ε. First, we partition the profitable elements (and possibly some more elements) into a small number of *profit classes*, where elements from the same profit class have *similar* profits. The profit classes are derived from a 2-approximation α for $\text{OPT}(I)$, which can be easily computed in polynomial time. Specifically, for all $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$ define the *r-profit class* as

$$\mathcal{K}_r(\alpha) = \left\{ e \in E \;\middle|\; \frac{p(e)}{2 \cdot \alpha} \in \left((1-\varepsilon)^r, (1-\varepsilon)^{r-1}\right] \right\}. \tag{2}$$

In the following we give a definition of an *exchange set* for each profit class. This facilitates the construction of an SRS. In words, a subset of elements X is an exchange set for some profit class $\mathcal{K}_r(\alpha)$ if any feasible set Δ and element $a \in (\Delta \cap \mathcal{K}_r(\alpha)) \setminus X$ can be replaced (while maintaining feasibility) by some element $b \in (X \cap \mathcal{K}_r(\alpha)) \setminus \Delta$, such that the cost of b is no larger than the cost of a. Formally,

▶ **Definition 8.** *Let $I = (E, \mathcal{C}, c, p, \beta)$ be a* BC *instance, $0 < \varepsilon < \frac{1}{2}$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$, and $X \subseteq \mathcal{K}_r(\alpha)$. We say that X is an* exchange set *for I, ε, α, and r if:*

- *For all $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in (\Delta \cap \mathcal{K}_r(\alpha)) \setminus X$ there is $b \in (\mathcal{K}_r(\alpha) \cap X) \setminus \Delta$ satisfying*
  - *$c(b) \leq c(a)$.*
  - *$\Delta - a + b \in \mathcal{M}_{\leq q(\varepsilon)}$.*

The similarity between SRS and exchange sets is not coincidental. We show that if a set $R \subseteq E$ satisfies that $R \cap \mathcal{K}_r(\alpha)$ is an exchange set for any $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$, then R is an SRS, and thus also a representative set by Lemma 7. This allows us to construct an SRS using a union of disjoint exchange sets, one for each profit class.

▶ **Lemma 9.** *Let $I = (E, \mathcal{C}, c, p, \beta)$ be a* BC *instance, $0 < \varepsilon < \frac{1}{2}$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$ and $R \subseteq E$. If for all $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$ it holds that $R \cap \mathcal{K}_r(\alpha)$ is an exchange set for I, ε, α, and r, then R is a representative set of I and ε.*

We give the formal proof in [5]. We now present a unified algorithm for finding a representative set for both types of constraints, namely, matching or matroid intersection constraints. This is achieved by taking the union of exchange sets of all profit classes. Nevertheless, for the construction of exchange sets we distinguish between the two types of constraints. This results also in different sizes for the obtained representative sets. Our algorithms for finding the exchange sets are the core technical contribution of this paper.

For matching constraints, we design an algorithm which constructs an exchange set for any profit class by finding multiple matchings of $\mathcal{C}$ from the given profit class. Each matching has a bounded cardinality, and the edges are chosen using a greedy approach to minimize the cost. We give the full details and a formal proof of Lemma 10 in Section 4.

▶ **Lemma 10.** *There is an algorithm* ExSet-Matching *that given a* BM *instance I, $0 < \varepsilon < \frac{1}{2}$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, and $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$, returns in time $q(\varepsilon) \cdot \text{poly}(|I|)$ an exchange set X for I, ε, α, and r, such that $|X| \leq 18 \cdot q(\varepsilon)^2$.*

Our algorithm for matroid intersection constraints is more involved and generates an exchange set by an *asymmetric interpretation* of the two given matroids. As the technique was introduced by Huang and Ward [9], the proof of the next lemma follows immediately from Theorem 3.6 in [9]. For completeness, we give the full details in Section 5.

▶ **Lemma 11.** *There is an algorithm* ExSet-MatroidIntersection *that given a* BI *instance* I, $0 < \varepsilon < \frac{1}{2}$, $\frac{\mathrm{OPT}(I)}{2} \leq \alpha \leq \mathrm{OPT}(I)$, *and* $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$, *returns in time* $q(\varepsilon)^{O(q(\varepsilon))} \cdot \mathrm{poly}(|I|)$ *an exchange set* X *for* I, ε, α, *and* r, *such that* $|X| \leq q(\varepsilon)^{O(q(\varepsilon))}$.

Using the above, we design an algorithm that returns a representative set for both types of constraints. This is done by computing a 2-approximation α for $\mathrm{OPT}(I)$, and then finding exchange sets for all profit classes, for the corresponding type of constraint. Finally, we return the union of the above exchange sets. The pseudocode of our algorithm, RepSet, is given in Algorithm 1.

Algorithm 1 RepSet$(I = (E, \mathcal{C}, c, p, \beta), \varepsilon)$.

```
input  : A BC instance I and error parameter 0 < ε < 1/2.
output : A representative set R of I and ε.
Compute a 2-approximation S* for I using a PTAS for BC with parameter ε' = 1/2.
Set α ← p(S*).
Initialize R ← ∅.
for r ∈ [log_{1−ε}(ε/2) + 1] do
    if I is a BM instance then
        R ← R ∪ ExSet-Matching(I, ε, α, r).
    else
        R ← R ∪ ExSet-MatroidIntersection(I, ε, α, r).
Return R.
```

▶ **Lemma 12.** *Given a* BC *instance* $I = (E, \mathcal{C}, c, p, \beta)$ *and* $0 < \varepsilon < \frac{1}{2}$, *Algorithm 1 returns a representative set* R *of* I *and* ε, *such that one of the following holds.*

- *If* $\mathcal{C}$ *is a matching constraint the running time is* $q(\varepsilon)^2 \cdot \mathrm{poly}(|I|)$, *and* $|R| \leq 54 \cdot q(\varepsilon)^3$.
- *If* $\mathcal{C}$ *is a matroid intersection constraint the running time is* $q(\varepsilon)^{O(q(\varepsilon))} \cdot \mathrm{poly}(|I|)$, *and* $|R| \leq q(\varepsilon)^{O(q(\varepsilon))}$.

The proof of the lemma is given in [5]. Next, we use a result of [1] for adding elements of smaller profits to the solution. The techniques of [1] are based on a non-trivial patching of two solutions of the Lagrangian relaxation of BC (for both matching and matroid intersection constraints). This approach yields a feasible set of almost optimal profit; in the worst case, the difference from the optimum is twice the maximal profit of an element in the instance. Since we use the latter approach only for non-profitable elements, this effectively does not harm our approximation guarantee. The following is a compact statement of the above result of [1].

▶ **Lemma 13.** *There is a polynomial-time algorithm* NonProfitableSolver *that given a* BC *instance* $I = (E, \mathcal{C}, c, p, \beta)$ *computes a solution* S *for* I *of profit* $p(S) \geq \mathrm{OPT}(I) - 2 \cdot \max_{e \in E} p(e)$.

Using the algorithm above and our algorithm for computing a representative set, we obtain an EPTAS for BC. Let R be the representative set returned by RepSet(I, ε). Our scheme enumerates over subsets of R to select profitable elements for the solution. Using algorithm NonProfitableSolver of [1], the solution is extended to include also non-profitable elements. Specifically, let $\frac{\mathrm{OPT}(I)}{2} \leq \alpha \leq \mathrm{OPT}(I)$ be a 2-approximation for the optimal profit for I. In addition, let $E(\alpha) = \{e \in E \mid p(e) \leq 2\varepsilon \cdot \alpha\}$ be the set including the non-profitable elements, and possibly also profitable elements $e \in E$ such that $p(e) \leq 2\varepsilon \cdot \mathrm{OPT}(I)$. Given a feasible set $F \in \mathcal{M}$, we define a residual BC instance containing elements which can *extend* F by adding elements from $E(\alpha)$. More formally,

▶ **Definition 14.** *Given a* BC *instance* $I = (E, \mathcal{C}, c, p, \beta)$, $\frac{\mathrm{OPT}(I)}{2} \leq \alpha \leq \mathrm{OPT}(I)$, *and* $F \in \mathcal{M}(\mathcal{C})$, *the* residual instance of F and α *for* I *is the* BC *instance* $I_F(\alpha) = (E_F, \mathcal{C}_F, c_F, p_F, \beta_F)$ *defined as follows.*

- $E_F = E(\alpha) \setminus F$.
- $\mathcal{C}_F = \mathcal{C}/F$.
- $p_F = p|_F$ *(i.e., the restriction of* p *to* F*).*
- $c_F = c|_F$.
- $\beta_F = \beta - c(F)$.

▶ **Observation 15.** *Let* $I = (E, \mathcal{C}, c, p, \beta)$ *be a* BC *instance,* $\frac{\mathrm{OPT}(I)}{2} \leq \alpha \leq \mathrm{OPT}(I)$, $F \in \mathcal{M}(\mathcal{C})$, *and let* T *be a solution for* $I_F(\alpha)$*. Then,* $T \cup F$ *is a solution for* I.

For all solutions $F \subseteq R$ for I with $|F| \leq \varepsilon^{-1}$, we find a solution T_F for the residual instance $I_F(\alpha)$ using Algorithm NonProfitableSolver and define $K_F = T_F \cup F$ as the *extended solution* of F. Our scheme iterates over the extended solutions K_F, for all such solutions F, and chooses an extended solution K_{F^*} of maximal total profit. The pseudocode of the scheme is given in Algorithm 2.

Algorithm 2 EPTAS($I = (E, \mathcal{C}, c, p, \beta), \varepsilon$).

```
input  : A BC instance I and an error parameter 0 < ε < 1/2.
output : A solution for I.
Construct the representative set R ← RepSet(I, ε).
Compute a 2-approximation S* for I using a PTAS for BC with parameter ε' = 1/2.
Set α ← p(S*).
Initialize an empty solution A ← ∅.
for F ⊆ R s.t. |F| ≤ ε^{-1} and F is a solution of I do
    Find a solution for I_F(α) by T_F ← NonProfitableSolver(I_F(α)).
    Let K_F ← T_F ∪ F.
    if p(K_F) > p(A) then
        Update A ← K_F
Return A.
```

The running time of Algorithm 2 crucially depends on the cardinality of the representative set. Roughly speaking, the running time is the number of subsets of the representative set containing at most ε^{-1} elements, multiplied by a computation time that is polynomial in the encoding size of the instance. Moreover, since $R = \mathsf{RepSet}(I, \varepsilon)$ is a representative set (by Lemma 12), there is an almost optimal solution S of I such that the profitable elements in S are a subset of R. Thus, there is an iteration of the **for** loop in Algorithm 2 such that $F = S \cap H$. In the proof of Lemma 16 we focus on this iteration and show that it yields a solution K_F of I with an almost optimal profit.

▶ **Lemma 16.** *Given a* BC *instance* $I = (E, \mathcal{C}, c, p, \beta)$ *and* $0 < \varepsilon < \frac{1}{2}$, *Algorithm 2 returns a solution for* I *of profit at least* $(1 - 8\varepsilon) \cdot \mathrm{OPT}(I)$ *such that one of the following holds.*

- *If* I *is a* BM *instance the running time is* $2^{O\left(\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)} \cdot \mathrm{poly}(|I|)$.
- *If* I *is a* BI *instance the running time is* $q(\varepsilon)^{O(\varepsilon^{-1} \cdot q(\varepsilon))} \cdot \mathrm{poly}(|I|)$, *where* $q(\varepsilon) = \left\lceil \varepsilon^{-\varepsilon^{-1}} \right\rceil$.

The proof of Lemma 16 is given in [5]. We are ready to prove our main results.

Proofs of Theorem 1 and Theorem 2. Given a BC instance I and $0 < \varepsilon < \frac{1}{2}$, using Algorithm 2 for I with parameter $\frac{\varepsilon}{8}$ we have by Lemma 16 the desired approximation guarantee. Furthermore, the running time is $2^{O\left(\varepsilon^{-2} \log \frac{1}{\varepsilon}\right)} \cdot \mathrm{poly}(|I|)$ or $q(\varepsilon)^{O(\varepsilon^{-1} \cdot q(\varepsilon))} \cdot \mathrm{poly}(|I|)$, depending on whether I is a BM instance or a BI instance, respectively. ◀

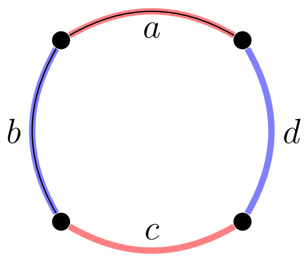

Figure 1 An example showing that bipartite matching may not yield an exchange set. Consider the two matchings $\Delta_1 = \{a, c\}, \Delta_2 = \{b, d\}$ marked in red and blue, and suppose that $\mathcal{K}_r(\alpha) = \{a, b\}$ is a profit class. The only exchange set for $\mathcal{K}_r(\alpha)$ is $\{a, b\}$, which is not a matching. Note that a bipartite matching can be cast as matroid intersection. For a bipartite graph $G = (L \cup R, E)$, define the matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$, where $\mathcal{I}_1 = \{F \subseteq E |\ \forall v \in L : |F \cap N(v)| \leq 1\}$, and $\mathcal{I}_2 = \{F \subseteq E |\ \forall v \in R : |F \cap N(v)| \leq 1\}$, where $N(v)$ is the set of neighbors of v. Thus, bipartite matching is a special case of both matching and matroid intersection.

4 Exchange Set for Matching Constraints

In this section we design an algorithm for finding an exchange set for a BM instance and a profit class, leading to the proof of Lemma 10. For the remainder of this section, fix a BM instance $I = (E, \mathcal{C}, c, p, \beta)$, an error parameter $0 < \varepsilon < \frac{1}{2}$, a 2-approximation for $\text{OPT}(I)$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, and an index $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$ of the profit class $\mathcal{K}_r(\alpha)$.

We note that for a single matroid constraint an exchange set can be constructed by finding a minimum cost basis in the matroid [6]. More specifically, given a matroid $\mathcal{G} = (E, \mathcal{I})$, it is shown in [6] that a minimum cost basis in the matroid $[\mathcal{G} \cap \mathcal{K}_r(\alpha)]_{\leq q(\varepsilon)}$ is an exchange set for $\mathcal{K}_r(\alpha)$. Such exchange set can be easily computed using a greedy approach. An analogue for the setting of matching constraints is to find a matching of cardinality $\Omega(q(\varepsilon))$ and minimum total cost in $\mathcal{K}_r(\alpha)$. However, as shown in Figure 1, this idea fails. Thus, we turn to use a completely different approach.

A key observation is that even if a greedy matching algorithm may not suffice for the construction of an exchange set, applying such an algorithm multiple times can be the solution. Thus, as a subroutine our algorithm finds a matching using a greedy approach. The algorithm iteratively selects an edge of minimal cost while ensuring that the selected set of edges is a matching. This is done until the algorithm reaches a given cardinality bound, or no more edges can be added. The pseudocode of GreedyMatching is given in Algorithm 3.[7]

Algorithm 3 GreedyMatching($G = (V, E), N, c$).

input : A graph G, an integer $N \in \mathbb{N} \setminus \{0\}$, and a cost function $c : E \to \mathbb{R}_{\geq 0}$.
output : A matching M of G.

Initialize $M \leftarrow \emptyset$.
while $|M| < N$ and $E/M \neq \emptyset$ **do**
  Find $e \in E/M$ of minimal cost w.r.t. c.
  Update $M \leftarrow M + e$.
Return M.

[7] Given a graph $G = (V, E)$ and a matching M of G, the definition of thinning E/M is given in Section 2.

Given a graph $G = (V, E)$ and two edges $a, b \in E$, we say that a, b are *adjacent* if there are $x, y, z \in V$ such that $a = \{x, y\}$ and $b = \{y, z\}$; for all $e \in E$, let $\mathsf{Adj}_G(e)$ be the set of edges adjacent to e in G. In the next result we show that if an edge a is not selected for the solution by GreedyMatching, then either the algorithm selects an adjacent edge of cost at most $c(a)$, or all of the selected edges have costs at most $c(a)$.

▶ **Lemma 17.** *Given a graph $G = (V, E)$, $N \in \mathbb{N} \setminus \{0\}$, and $c : E \to \mathbb{R}_{\geq 0}$, Algorithm 3 returns in polynomial time a matching M of G such that for all $a \in E \setminus M$ one of the following holds.*

1. *$|M| \leq N$ and there is $b \in \mathsf{Adj}_G(a) \cap M$ such that $c(b) \leq c(a)$.*
2. *$|M| = N$, for all $b \in M$ it holds that $c(b) \leq c(a)$, and $M + a$ is a matching of G.*

Proof. Clearly, Algorithm 3 returns in polynomial time a matching M of G. Observe that $|M| \leq N$ by Step 2. To prove that either 1. or 2. hold, we distinguish between two cases.

- $a \notin E/M$. Then $\mathsf{Adj}_G(a) \cap M \neq \emptyset$. Let e be the first edge in $\mathsf{Adj}_G(a) \cap M$ that is added to M in Step 4; also, let L be the set of edges added to M before e. Then $a \in E/L$, since L does not contain edges adjacent to a. By Step 3, it holds that $c(e) = \min_{e' \in E/L} c(e') \leq c(a)$.
- $a \in E/M$. Thus, $|M| = N$; otherwise, a would be added to M. Also, $M + a$ is a matching of G. Now, let $b \in M$, and let K be the set of edges added to M before b. Since $M + a$ is a matching of G, by the hereditary property of $(E, \mathcal{M}(G))$ it holds that $K + a$ is a matching of G; thus, $a \in E/K$ and by Step 3 it follows that $c(b) = \min_{e' \in E/K} c(e') \leq c(a)$. ◀

By Lemma 17, we argue that an exchange set can be found by using Algorithm GreedyMatching iteratively. Specifically, let $k(\varepsilon) = 6 \cdot q(\varepsilon)$ and $N(\varepsilon) = 3 \cdot q(\varepsilon)$. We run Algorithm GreedyMatching for $k(\varepsilon)$ iterations, each iteration with a bound $N(\varepsilon)$ on the cardinality of the matching. In iteration i, we choose a matching M_i from the edges of the profit class $\mathcal{K}_r(\alpha)$ and remove the chosen edges from the graph. Therefore, in the following iterations, edges adjacent to previously chosen edges can be chosen as well. A small-scale illustration of the algorithm is presented in Figure 2. The pseudocode of Algorithm ExSet-Matching, which computes an exchange set for the given profit class, is presented in Algorithm 4.

Algorithm 4 ExSet-Matching$(I = (E, \mathcal{C}, c, p, \beta), \varepsilon, \alpha, r)$.

input : a matching-BC instance I, $0 < \varepsilon < \frac{1}{2}$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$.
output: An exchange set for I, ε, α, and r.
Initialize $X \leftarrow \emptyset$ and $\mathcal{E}_0 \leftarrow \mathcal{K}_r(\alpha)$.
for $i \in \{1, \ldots, k(\varepsilon)\}$ **do**
  Define $G_i = (V, \mathcal{E}_{i-1})$ where V is the vertex set of $\mathcal{C}$.
  Compute $M_i \leftarrow$ GreedyMatching $\left(G_i, N(\varepsilon), c|_{\mathcal{E}_{i-1}}\right)$.
  Update $X \leftarrow X \cup M_i$ and define $\mathcal{E}_i \leftarrow \mathcal{E}_{i-1} \setminus M_i$.
Return X.

Algorithm ExSet-Matching outputs a union X of disjoint matchings $M_1, \ldots, M_{k(\varepsilon)}$ taken from the edges of the profit class $\mathcal{K}_r(\alpha)$. For some $\Delta \in \mathcal{M}(\mathcal{C})$ and $a \in (\Delta \cap \mathcal{K}_r(\alpha)) \setminus X$, by Lemma 17, there are two options summarizing the main idea in the proof of Lemma 10.

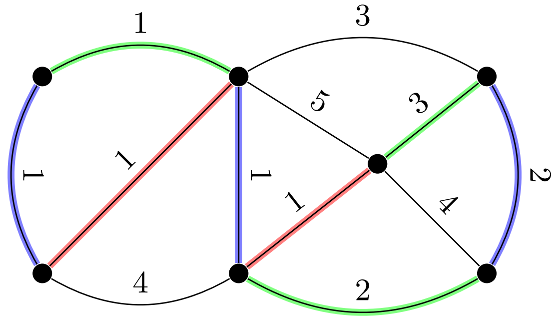

Figure 2 An execution of Algorithm ExSet-Matching with the (illegally small) parameters $N(\varepsilon) = k(\varepsilon) = 3$. The numbers by the edges are the costs. The edges chosen in iterations $i = 1, 2, 3$ are marked in blue, red, and green, respectively.

- all matchings M_i contain some b_i adjacent to a such that $c(b_i) \leq c(a)$. Then, as $k(\varepsilon)$ is sufficiently large, one such b_i is not adjacent to any edge in $\Delta - a$. Hence, $\Delta - a + b_i$ is a matching.
- One such M_i contains only edges of costs at most $c(a)$; as $N(\varepsilon)$ is sufficiently large, there is $b \in M_i$ such that $\Delta - a + b$ is a matching.

Proof of Lemma 10. For all $i \in \{1, \ldots, k(\varepsilon)\}$, let G_i and M_i be the outputs of Steps 3 and 4 in iteration i of the **for** loop in ExSet-Matching$(I, \varepsilon, \alpha, r)$, respectively. Also, let X be the output of the algorithm; observe that $X = \bigcup_{i \in [k(\varepsilon)]} M_i$. We show that X is an exchange set for I, ε, α and r (see Definition 8). Let $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in (\Delta \cap \mathcal{K}_r(\alpha)) \setminus X$. We use the next inequality in the claim below.

$$\frac{k(\varepsilon)}{2} = N(\varepsilon) = 3 \cdot q(\varepsilon) > 2 \cdot |\Delta| = |V(\Delta)|. \tag{3}$$

The inequality holds since $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$. The last equality holds since each vertex appears as an endpoint in a matching at most once.

▷ Claim 18. There is $b \in (X \cap \mathcal{K}_r(\alpha)) \setminus \Delta$ such that $\Delta - a + b \in \mathcal{M}_{\leq q(\varepsilon)}$, and $c(b) \leq c(a)$.

Proof. Let $a = \{x, y\}$, $I = (E, \mathcal{C}, c, p, \beta)$, and $\mathcal{C} = (V, E)$. Since $a \notin X$, for all $i \in \{1, \ldots, k(\varepsilon)\}$ it holds that $a \notin M_i$; thus, $a \in \mathcal{E}_i = \mathcal{E}_{i-1} \setminus M_i$. Hence, by Lemma 17, one of the following holds.

1. For all $i \in [k(\varepsilon)]$ there is $b_i \in \mathsf{Adj}_{G_i}(a) \cap M_i$ such that $c(b_i) \leq c(a)$. For $z \in \{x, y\}$ let

   $$J_z = \{i \in [k(\varepsilon)] \mid \exists u \in V : \; b_i = \{z, u\}\}$$

   be the set of indices of edges b_i neighboring to z. Since $b_i \in \mathsf{Adj}_{G_i}(a)$ it holds that $J_x \cup J_y = [k(\varepsilon)]$. Thus, there is $z \in \{x, y\}$ such that $|J_z| \geq \frac{k(\varepsilon)}{2} > |V(\Delta)|$, where the last inequality follows from (3). For any $i \in J_z$ let $v_i \in V$ be the vertex connected to z in b_i, that is $b_i = \{z, v_i\}$. Since the matchings $M_1, \ldots M_{k(\varepsilon)}$ are disjoint and $b_i \in M_i$ it follows that the vertices v_i for $i \in J_z$ are all distinct. As $|J_z| > |V(\Delta)|$ there is $i^* \in J_z$ such that $v_{i^*} \notin V(\Delta)$. Therefore, $\Delta - a + b_{i^*} \in \mathcal{M}_{\leq q(\varepsilon)}$ and $c(b_{i^*}) \leq c(a)$.
2. There is $i \in \{1, \ldots, k(\varepsilon)\}$ such that $|M_i| = N(\varepsilon)$, and for all $b \in M_i$ it holds that $c(b) \leq c(a)$. Then,

   $$|M_i| = N(\varepsilon) > |V(\Delta)|. \tag{4}$$

The equality follows by the definition of M_i in Case 2. The inequality follows from (3). Since each vertex appears as an endpoint in a matching at most once, by (4) there is $b \in M_i$ such that both endpoints of b are not in $V(\Delta)$. Thus, $\Delta + b \in \mathcal{M}$; by the hereditary property and since $a \in \Delta$, it holds that $\Delta - a + b \in \mathcal{M}_{\leq q(\varepsilon)}$. ◁

By Claim 18 and Definition 8, we have that X is an exchange set for I, ε, α, and r as required. To complete the proof of the lemma we show (in [5]) the following.

▷ Claim 19. $|X| \leq 18 \cdot q(\varepsilon)^2$, and the running time of Algorithm 4 is $q(\varepsilon) \cdot \text{poly}(|I|)$. ◀

5 Exchange Set for Matroid Intersection Constraints

In this section we design an algorithm for finding an exchange set for a profit class in a BI instance. For the remainder of this section, fix a BI instance $I = (E, \mathcal{C}, c, p, \beta)$, an error parameter $0 < \varepsilon < \frac{1}{2}$, a 2-approximation for $\text{OPT}(I)$, $\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, and an index $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right)+1]$ of the profit class $\mathcal{K}_r(\alpha)$. Also, let $\mathcal{C} = (\mathcal{I}_1, \mathcal{I}_2)$ be the matroid intersection constraint $\mathcal{C}$ of I. For simplicity, when understood from the context, some of the lemmas in this section consider the given parameters (e.g., I) without explicit mention. The proofs of the lemmas in this section are given in the full version of the paper [5].

As shown in Figure 1, a simple greedy approach which finds a feasible set of minimum cost (within $\mathcal{K}_r(\alpha)$) in the intersection of the matroids may not output an exchange set for $\mathcal{K}_r(\alpha)$. Instead, our approach builds on some interesting properties of matroid intersection. The next definition presents a *shifting property* for a feasible set $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and an element $a \in \Delta \cap \mathcal{K}_r(\alpha)$ w.r.t. the two matroids. We use this property to show that our algorithm constructs an exchange set.

▶ **Definition 20.** *Let $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$, $a \in \Delta \cap \mathcal{K}_r(\alpha)$ and $b \in \mathcal{K}_r(\alpha) \setminus \Delta$. We say that b is a* shift *to a for Δ if $c(b) \leq c(a)$ and $\Delta - a + b \in \mathcal{M}_{\leq q(\varepsilon)}$. Furthermore, b is a* semi-shift *to a for Δ if $c(b) \leq c(a)$ and $\Delta - a + b \in \mathcal{I}_2$ but $\Delta - a + b \notin \mathcal{I}_1$.*

As a starting point for our exchange set algorithm, we show how to obtain small cardinality sets which contain either a shift or a semi-shift for every pair $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in \Delta \cap \mathcal{K}_r(\alpha)$.

▶ **Lemma 21.** *Let $U \subseteq \mathcal{K}_r(\alpha)$, $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$, and B be a minimum basis of $[(E, \mathcal{I}_2) \cap U]_{\leq q(\varepsilon)}$ w.r.t. c. Also, let $a \in (U \cap \Delta) \setminus B$. Then, there is $b \in B \setminus \Delta$ such that b is a semi-shift to a for Δ, or b is a shift to a for Δ.*

Observe that to obtain an exchange set, our goal is to find a subset of $\mathcal{K}_r(\alpha)$ which contains a shift for every pair $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in \Delta \cap \mathcal{K}_r(\alpha)$. Thus, using Lemma 21 we design the following recursive algorithm ExtendChain, which finds a union of minimum bases of matroids w.r.t $\mathcal{I}_2$, of increasingly restricted ground sets w.r.t. $\mathcal{I}_1$. The pseudocode of Algorithm ExtendChain is given in Algorithm 5.

We can view the execution of ExtendChain as a tree, where each node (called below a *branch*) corresponds to the subset $S \subseteq \mathcal{K}_r(\alpha)$ in a specific recursive call. We now describe the role of S in Algorithm ExtendChain. If $|S| \geq q(\varepsilon) + 1$, we simply return $\emptyset$; such a branch is called a *leaf*, and does not contribute elements to the constructed exchange set. Otherwise, define the *universe* of the branch S as $U_S = \{e \in \mathcal{K}_r(\alpha) \setminus S \mid S + e \in \mathcal{I}_1\}$; that is, elements in the universe of S which can be added to S to form an independent set w.r.t. $\mathcal{I}_1$. Next, we construct a minimum basis B_S w.r.t. c of the matroid $[(E, \mathcal{I}_2) \cap U_S]_{\leq q(\varepsilon)}$. Observe that B_S contains up to $q(\varepsilon)$ elements, selected from the universe of S, and that B_S is independent w.r.t. $\mathcal{I}_2$. Note that the definition of the universe relates to $\mathcal{I}_1$ while the construction of the bases to $\mathcal{I}_2$; thus, the two matroids play completely different roles in the algorithm.

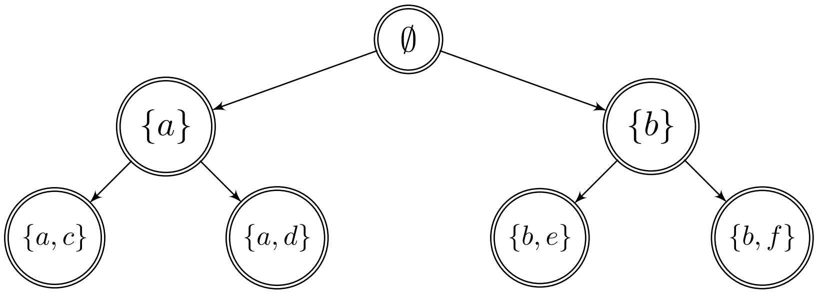

Figure 3 An illustration of the branches in Algorithm 5 for $S = \emptyset$. Note that $B_\emptyset = \{a, b\}$, $B_{\{a\}} = \{c, d\}$ and $B_{\{b\}} = \{e, f\}$. Also, $\{a, c\}$ and $\{a, d\}$ are the child branches of $\{a\}$.

For every element $e \in B_S$ we apply Algorithm ExtendChain recursively with $S' = S + e$ to find the corresponding basis B_{S+e}. The algorithm returns (using recursion) the union of the constructed bases over all branches. Finally, algorithm ExSet-MatroidIntersection constructs an exchange set for I, ε, α, and r by calling Algorithm ExtendChain with the initial branch (i.e., *root*) $S = \emptyset$:

$$\text{ExSet-MatroidIntersection}(I, \varepsilon, \alpha, r) = \text{ExtendChain}(I, \varepsilon, \alpha, r, \emptyset). \tag{5}$$

For an illustration of the algorithm, see Figure 3.

Algorithm 5 ExtendChain$(I = (E, \mathcal{C}, c, p, \beta), \varepsilon, \alpha, r, S)$.

input : a matroid-BC instance I , where $\mathcal{C} = (\mathcal{I}_1, \mathcal{I}_2)$, $0 < \varepsilon < \frac{1}{2}$,
$\frac{\text{OPT}(I)}{2} \leq \alpha \leq \text{OPT}(I)$, $r \in [\log_{1-\varepsilon}\left(\frac{\varepsilon}{2}\right) + 1]$, and $S \subseteq E$.
output : (for $S = \emptyset$) An exchange set X for I, ε, α, and r.

if $|S| \geq q(\varepsilon) + 1$ **then**
  Return $\emptyset$
Define $U_S = \{e \in \mathcal{K}_r(\alpha) \setminus S \mid S + e \in \mathcal{I}_1\}$.
Compute a minimum basis B_S w.r.t. c of the matroid $[(E, \mathcal{I}_2) \cap U_S]_{\leq q(\varepsilon)}$.
Return $B_S \cup \left(\bigcup_{e \in B_S} \text{ExtendChain}(I, \varepsilon, \alpha, r, S + e)\right)$.

In the analysis of the algorithm, we consider branches with useful attributes, called *chains*; these are essentially sequences of semi-shifts to some $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in \Delta \cap \mathcal{K}_r(\alpha)$. Let $X = \text{ExSet-MatroidIntersection}(I, \varepsilon, \alpha, r)$, and let $\mathcal{S}$ be the set of all branches $S \subseteq \mathcal{K}_r(\alpha)$ such that ExtendChain$(I, \varepsilon, \alpha, r, S)$ is computed during the construction of X.

▶ **Definition 22.** *Let $S \in \mathcal{S}$, $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$, and $a \in (\mathcal{K}_r(\alpha) \cap \Delta) \setminus X$. We say that S is a* chain *of a and Δ if $a \in U_S$, and for all $e \in S$ it holds that e is a semi-shift to a for Δ.*

Note that there must be a chain for a and Δ since the empty set satisfies the conditions of Definition 22. Moreover, we can bound the cardinality of a chain by $q(\varepsilon)$ using the exchange property of the matroid $(E, \mathcal{I}_1)$. The above arguments are formalized in the next lemmas.

▶ **Lemma 23.** *For all $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in (\mathcal{K}_r(\alpha) \cap \Delta) \setminus X$ there is $S \subseteq X$ such that S is a chain of a and Δ.*

▶ **Lemma 24.** *For all $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$, $a \in (\mathcal{K}_r(\alpha) \cap \Delta) \setminus X$, and a chain S of a and Δ, it holds that $|S| \leq q(\varepsilon)$.*

For a chain S of a and Δ, let B_S be the result of the first computation of Step 4 (i.e., not within a recursive call) in $\mathsf{ExtendChain}(I, \varepsilon, \alpha, r, S)$. The key argument in the proof of Lemma 11 is that for a chain S^* of maximal cardinality, B_{S^*} contains a shift to a and Δ, using the maximality of S^* and Lemma 21.

▶ **Lemma 25.** *For all $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$, $a \in (\mathcal{K}_r(\alpha) \cap \Delta) \setminus X$, and a chain S^* of a and Δ of maximum cardinality, there is a shift $b^* \in B_{S^*}$ to a for Δ.*

In the proof of Lemma 11, for every $\Delta \in \mathcal{M}_{\leq q(\varepsilon)}$ and $a \in (\mathcal{K}_r(\alpha) \cap \Delta) \setminus X$, we take a chain S^* of a and Δ of maximum cardinality (which exists by Lemma 23 and Lemma 24). Then, by Lemma 25, there is a shift b^* to a for Δ, and it follows that X is an exchange set for I, ε, α, and r. The formal proof is given in [5].

6 Discussion

In this paper we present the first EPTAS for budgeted matching and budgeted matroid intersection, thus improving upon the existing PTAS for both problems. We derive our results via a generalization of the representative set framework in [6]; this ameliorates the exhaustive enumeration applied in similar settings [1, 3].

We note that the framework based on representative sets may be useful for solving other problems formulated as (1). Indeed, the proofs of Lemma 7 and Lemma 9, which establish the representative set framework, are oblivious to the exact type of constraints and only require having a *k-exchange system* for some constant k.[8]

Furthermore, our exchange sets algorithms can be applied with slight modifications to other variants of our problems and are thus of independent interest. In particular, we can use a generalization of Algorithm 4 to construct an exchange set for the *budgeted b-matching* problem. Also, using the techniques of [9], Algorithm 5 can be generalized to construct exchange sets for budgeted *multi-matroid intersection* for any constant number of matroids; this includes the *budgeted multi-dimensional matching* problem. While this problem does not admit a PTAS unless P=NP [10], our initial study shows that by constructing a representative set we may obtain an FPT-*approximation scheme* by parameterizing on the number of elements in the solution.[9]

Finally, to resolve the complexity status of BM and BI, the gripping question of whether the problems admit an FPTAS needs to be answered. Unfortunately, this may be a very difficult task. Even for special cases of a single matroid, such as graphic matroid, the existence of an FPTAS is still open. Moreover, a deterministic FPTAS for budgeted matching would solve deterministically the exact matching problem, which has been open for over four decades [14].

References

1 André Berger, Vincenzo Bonifaci, Fabrizio Grandoni, and Guido Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 128(1):355–372, 2011.

2 Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.

[8] A set system $(E, \mathcal{I})$ satisfies the k-exchange property if for all $A \in \mathcal{I}$ and $e \in E$ there is $B \subseteq A, |B| \leq k$, such that $(A \setminus B) \cup \{e\} \in \mathcal{I}$.

[9] We refer the reader, e.g., to [12] for the definition of parameterized approximation algorithms running in fixed-parameter tractable (FPT)-time.

3 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1080–1097. SIAM, 2011.

4 Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

5 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for budgeted matching and budgeted matroid intersection. *arXiv preprint*, 2023. arXiv:2302.05681.

6 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for budgeted matroid independent set. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 69–83, 2023.

7 Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

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

9 Chien-Chung Huang and Justin Ward. FPT-algorithms for the ℓ-matchoid problem with a coverage objective. *arXiv preprint*, 2020. arXiv:2011.06268.

10 Viggo Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, 1991.

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

12 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

13 Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 345–354, 1987.

14 Christos H Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM (JACM)*, 29(2):285–309, 1982.

15 Ulrich Pferschy and Joachim Schauer. The knapsack problem with conflict graphs. *J. Graph Algorithms Appl.*, 13(2):233–249, 2009.

16 Ram Ravi and Michel X Goemans. The constrained minimum spanning tree problem. In *Scandinavian Workshop on Algorithm Theory*, pages 66–75. Springer, 1996.

17 Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

18 Petra Schuurman and Gerhard J Woeginger. Approximation schemes – A tutorial. *Lectures on scheduling*, 2001.

19 Hadas Shachnai and Tami Tamir. Polynomial time approximation schemes. In *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 1: Methologies and Traditional Applications*, pages 125–156. CRC Press, 2018.