Optimal Matroid Partitioning Problems*

Yasushi Kawase†[1], Kei Kimura‡[2], Kazuhisa Makino§[3], and Hanna Sumita¶[4]

1 **Tokyo Institute of Technology, Tokyo, Japan**
`kawase.y.ab@m.titech.ac.jp`
2 **Toyohashi University of Technology, Toyohashi, Japan**
`kimura@cs.tut.ac.jp`
3 **Kyoto University, Kyoto, Japan**
`makino@kurims.kyoto-u.ac.jp`
4 **National Institute of Informatics, Tokyo, Japan**
`sumita@nii.ac.jp`

Abstract

This paper studies optimal matroid partitioning problems for various objective functions. In the problem, we are given a finite set E and k weighted matroids $(E, \mathcal{I}_i, w_i)$, $i = 1, \ldots, k$, and our task is to find a minimum partition $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}_i$ for all i. For each objective function, we give a polynomial-time algorithm or prove NP-hardness. In particular, for the case when the given weighted matroids are identical and the objective function is the sum of the maximum weight in each set (i.e., $\sum_{i=1}^{k} \max_{e \in I_i} w_i(e)$), we show that the problem is strongly NP-hard but admits a PTAS.

1998 ACM Subject Classification G.2.1 Combinatorial algorithms

Keywords and phrases Matroids, Partitioning problem, PTAS, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.51

1 Introduction

The *matroid partitioning problem* is one of the most fundamental problems in combinatorial optimization. In this problem, we are given a finite set E and k matroids $(E, \mathcal{I}_i)$, $i = 1, \ldots, k$, and our task is to find a partition $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}_i$ for all i. We say that such a partition $(I_1, \ldots, I_k)$ of E is *feasible*. The matroid partitioning problem has been eagerly studied in a series of papers investigating structures of matroids. See, e.g., [7, 8, 9, 16, 23] for details. In this paper, we study weighted versions of the matroid partitioning problem. Namely, we assume that each matroid $(E, \mathcal{I}_i)$ has a weight function $w_i : E \to \mathbb{R}_+$. We consider several possible objective functions of the matroid partitioning problem.

Let $\mathrm{Op}^{(1)}$ and $\mathrm{Op}^{(2)}$ denote two mathematical operators taken from $\{\max, \min, \sum\}$. For any partition $P = (I_1, \ldots, I_k)$ of E, we call $\mathrm{Op}^{(1)}{}_{i=1,\ldots,k}\, \mathrm{Op}^{(2)}{}_{e \in I_i}\, w_i(e)$ the $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-*value* of P. For example, $(\sum, \min)$-value of P denotes $\sum_{i=1,\ldots,k} \min_{e \in I_i} w_i(e)$.

We define the *minimum* $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-*value matroid partitioning problem* as the one for finding a feasible partition with minimum $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-value. The maximum problems are

* A full version of the paper is available at [14], `https://arxiv.org/abs/1710.00950`.
† The first author is supported by JSPS KAKENHI Grant Number JP16K16005.
‡ The second author is supported by JSPS KAKENHI Grant Number JP15H06286.0
§ The third author is supported by JSPS KAKENHI Grant Number JP24106002, JP25280004, JP26280001, and JST CREST Grant Number JPMJCR1402, Japan.
¶ The forth author is supported by JST ERATO Grant Number JPMJER1201, Japan.

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 51; pp. 51:1–51:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

defined analogously. These matroid partitioning problems are natural to study, and have many applications in various areas such as scheduling and combinatorial optimization. We note that all the matroids and/or all the weights may be identical in case such as scheduling with identical machines.

The minimum $(\sum, \sum)$-value matroid partitioning problem is reducible to the *weighted matroid intersection problem*, and vice versa [8]. Here, the weighted matroid intersection problem is to find a maximum weight subset that is simultaneously independent in two given matroids. It is known that this problem is polynomially solvable, and many papers have worked on algorithmic aspects of this problem [16, 23]. Generalizations of the weighted matroid intersection problem have also been studied [17, 21, 18].

Special cases of the minimum $(\max, \sum)$-value matroid partitioning problem have been extensively addressed in the scheduling literature under the name of the minimum makespan scheduling. Since this problem is NP-hard, many papers have proposed polynomial-time approximation algorithms. We remark that most papers focused on subclasses of matroids as inputs: for example, free matroids [19, 24], partition matroids [25, 26, 20], uniform matroid [15, 1, 4], and general matroids [26]. Approximation algorithms for the maximum $(\min, \sum)$-value matroid partitioning problem are also well-studied, see, e.g., [3, 13, 25, 20].

The other matroid partitioning problems also have many applications, and yet they are not much studied especially for general matroids. We here describe some examples of applications.

Maximum total capacity spanning tree partition Assume that we are given an undirected weighted graph $G = (V, E; w)$, which can be partitioned into k edge-disjoint spanning trees. The maximum total capacity spanning tree partition problem is to compute a partition of the edges into k edge-disjoint spanning trees such that the total of the minimum weight in each spanning tree is maximized. Then, the problem can be written as the maximum $(\sum, \min)$-value matroid partitioning problem having k identical graphic matroids, where the $(\sum, \min)$-value is $\sum_{i=1}^{k} \min_{e \in I_i} w(e)$.

Minimum total memory of a scheduling In this problem we are also given n jobs E and k identical machines, and each job needs to be scheduled on exactly one machine. In addition, we are given size $s(e)$ of job $e \in E$. The set of feasible allocation for each machine i is represented by a family of independent sets $\mathcal{I}_i$ of a matroid. The goal of the problem is to minimize the total memory needed, i.e., $(\sum, \max)$-value $\sum_{i=1}^{k} \max_{e \in I_i} s(e)$.

Burkard and Yao [2] showed that the minimum $(\sum, \max)$-value matroid problem can be solved by a greedy algorithm for a subclass of matroids, which includes partition matroids. Dell'Olmo et al. [5] investigated optimal matroid partitioning problems where the input matroids are identical partition matroids.

The goal of our paper is to analyze the computational complexity of these matroid partitioning problems for general matroids.

Our results

We first show that the maximization problems can be reduced to the minimization problems. For example, the maximum $(\sum, \min)$-value matroid partitioning problem can be transformed to the minimum $(\sum, \max)$-value matroid partitioning problem. Hence, we focus only on the minimization problems.

Our main result is to analyze the computational complexity of the minimum $(\sum, \max)$-value matroid partitioning problem. This problem contains the maximum total capacity spanning tree partitioning problem and the minimum total memory scheduling problem. We first show that the problem is strongly NP-hard even when the matroids and weights

Table 1 The time complexity of the optimal matroid partitioning problems (the results of the paper are in bold). Identical case means $\mathcal{I}_1 = \cdots = \mathcal{I}_k$ and $w_1 = \cdots = w_k$.

objective	identical case	general case	reference
(Σ, Σ)	P	P	[8, 11]
$(\max, \Sigma)$	SNP-hard	SNP-hard	[12]
$(\Sigma, \max)$	**PTAS**	**εk-approx.**	Section 3
	SNP-hard	**NP-hard even for $o(\log k)$-approx.**	
$(\min, \min)$	**P**	**P**	Section 4
$(\max, \max)$	**P**	**P**	Section 4
$(\min, \max)$	**P**	**P**	Section 4
$(\min, \Sigma)$	**P**	**P**	Section 4
$(\max, \min)$	**P**	**NP-hard even to approximate**	Section 4
$(\Sigma, \min)$	**P**	**NP-hard even to approximate**	Section 4

are respectively identical. However, for such instances, we also propose a *polynomial-time approximation scheme* (PTAS), i.e., a polynomial-time algorithm that outputs a $(1+\varepsilon)$-approximate solution for each fixed $\varepsilon > 0$. Our PTAS computes an approximate solution by two steps: guess the maximum weight in each I_i^* for an optimal solution $(I_1^*, \ldots, I_k^*)$, and check the existence of such a feasible partition. We remark that the number of possible combinations of maximum weights is $|E|^k$ and it may be too large. To reduce the possibility, we use rounding techniques in the design of the PTAS. First, we guess the maximum weight in I_i^* for only s indices. Furthermore, we round the weight of each element and reduce the number of different weights to a small number r. Then, now we have r^s possibilities. To obtain the approximation ratio $(1+\varepsilon)$, we need to set r and s to be $\Omega(\log k)$ respectively, and hence the number of possibilities r^s is still large. Our idea to tackle this is to enumerate *sequences* of maximum weights in the nonincreasing order. This enables us to reduce the number of possibilities to $\binom{r+s-1}{r}$ $(\leq 2^{r+s-1})$. This implies that our algorithm is a PTAS.

Moreover, for the $(\sum, \max)$ case with general inputs, we provide an εk-approximation algorithm for any $\varepsilon > 0$. The construction is similar to the identical case. We also prove the NP-hardness even to approximate the problem within a factor of $o(\log k)$.

For the $(\min, \min)$, $(\max, \max)$, $(\min, \max)$, and $(\min, \sum)$ cases, we provide polynomial-time algorithms. The main idea of these algorithms is a reduction to the feasibility problem of the matroid partitioning problem. For the $(\max, \min)$ and $(\sum, \min)$ cases, we give polynomial-time algorithms when the matroids and weights are respectively identical, and prove strong NP-hardness even to approximate for the general case. These results are summarized in Table 1 with their references.

Due to the space limitation, we omit proofs of some results, which are found in [14].

2 Preliminaries

A *matroid* is a set system $(E, \mathcal{I})$ with the following properties: (I1) $\emptyset \in \mathcal{I}$, (I2) $X \subseteq Y \in \mathcal{I}$ implies $X \in \mathcal{I}$, and (I3) $X, Y \in \mathcal{I}$, $|X| < |Y|$ implies the existence of $e \in Y \setminus X$ such that $X \cup \{e\} \in \mathcal{I}$. A set $I \subseteq \mathcal{I}$ is said to be *independent*, and an inclusion-wise maximal independent set is called a *base*. We denote the set of bases of $(E, \mathcal{I})$ by $B(\mathcal{I})$. All bases of a matroid have the same cardinality, which is called the *rank* of the matroid and is denoted by $\mathrm{rank}(\mathcal{I})$. For any $B_1, B_2 \in B(\mathcal{I})$ and $e_1 \in B_1 \setminus B_2$, there exists $e_2 \in B_2 \setminus B_1$ such that $B_1 - e_1 + e_2 \in B(\mathcal{I})$ and $B_2 - e_2 + e_1 \in B(\mathcal{I})$.

For a matroid $(E, \mathcal{I})$, a subset $A \subseteq E$, and a nonnegative integer $l \in \mathbb{Z}_+$, define $\mathcal{I}|A = \{X : A \supseteq X \in \mathcal{I}\}$, $\mathcal{I} \setminus A = \{X \setminus A : X \in \mathcal{I}\}$, $\mathcal{I}/A = \{X \subseteq E \setminus A : \mathrm{rank}(X \cup A) - \mathrm{rank}(A) = |X|\}$, and $\mathcal{I}^{(l)} = \{X \in \mathcal{I} : |X| \leq l\}$. We call $(A, \mathcal{I}|A)$, $(E \setminus A, \mathcal{I} \setminus A)$, $(E \setminus A, \mathcal{I}/A)$, and $(E, \mathcal{I}^{(l)})$, respectively, the *restriction*, *deletion*, *contraction*, and *truncation* of $(E, \mathcal{I})$. It is well known that $(A, \mathcal{I}|A)$, $(A, \mathcal{I} \setminus A)$, $(E \setminus A, \mathcal{I}/A)$, and $(E, \mathcal{I}^{(l)})$ are all matroids. Given matroids $\mathcal{M}_1 = (E_1, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E_2, \mathcal{I}_2)$, we define the *matroid union*, denoted by $\mathcal{M}_1 \vee \mathcal{M}_2$, to be $(E_1 \cup E_2, \mathcal{I}_1 \vee \mathcal{I}_2)$ where $\mathcal{I}_1 \vee \mathcal{I}_2 = \{I_1 \cup I_2 \,:\, I_1 \in \mathcal{I}_1,\ I_2 \in \mathcal{I}_2\}$. Any matroid union is also a matroid.

2.1 Model

Throughout the paper, we assume that every matroid is given by an independence oracle, which checks whether a given set is independent. Let k be a positive integer. We denote $[k] = \{1, \ldots, k\}$. Let $(E, \mathcal{I}_i)$ be a matroid and $w_i : E \to \mathbb{R}_+$ be a nonnegative weight function for $i \in [k]$. We denote $n = |E|$. For any k sets $I_1, \ldots, I_k \subseteq E$, we call $(I_1, \ldots, I_k)$ a *feasible partition* of E if it satisfies that $\bigcup_{i \in [k]} I_i = E$, $I_i \neq \emptyset$ $(\forall i \in [k])$[1], $I_i \cap I_j = \emptyset$ $(\forall i, j \in [k],\, i \neq j)$, and $I_i \in \mathcal{I}_i$ $(\forall i \in [k])$. In particular, $(I_1, \ldots, I_k)$ is said to be a *base partition* if it is a feasible partition and $I_i \in B(\mathcal{I}_i)$ for all $i \in [k]$. For two operators $\mathrm{Op}^{(1)} \in \{\max, \min, \sum\}$ and $\mathrm{Op}^{(2)} \in \{\max, \min, \sum\}$, we define the $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-value of a feasible partition $(I_1, \ldots, I_k)$ as ${\mathrm{Op}^{(1)}}_{i \in [k]} {\mathrm{Op}^{(2)}}_{e \in I_i} w_i(e)$. In this article, we study the following minimization problem:

$$\min_{(I_1, \ldots, I_k):\text{ feasible partition}} {\mathrm{Op}^{(1)}}_{i \in [k]} {\mathrm{Op}^{(2)}}_{e \in I_i} w_i(e).$$

We refer to the problem as the *minimum* $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$*-value matroid partitioning problem*. We write a problem instance as $(E, (\mathcal{I}_i, w_i)_{i \in [k]})$. If $(\mathcal{I}_i, w_i)$ are identical for all $i \in [k]$, we write $(E, (\mathcal{I}, w), k)$. For the identical case, we can consider the partitioning problem where k is also a variable. This problem can be solved by solving $(E, (\mathcal{I}, w), i)$ for $i = 1, \ldots, n$. Thus it suffices to focus on the problem where k is given.

It is known to be easy to decide whether there exists a feasible partition or not. Moreover, the minimum $(\sum, \sum)$-value matroid partitioning problem can be solved in polynomial time. These facts are useful to show our results later.

▶ **Theorem 1** ([8, 11])**.** *There exists a polynomial-time algorithm that decides whether or not there exists a feasible partition for any given matroids $(E, \mathcal{I}_1), \ldots, (E, \mathcal{I}_k)$. Moreover, if it exists, we can find a feasible partition with minimum $(\sum, \sum)$-value in polynomial time.*

2.2 Basic properties

In this subsection, we prove basic properties of the partitioning problems. These properties imply that the minimization and maximization versions of matroid partitioning problems can be reduced to each other.

We first observe that we only need to consider base partitioning problems. Let $\mathcal{M}_i = (E, \mathcal{I}_i)$ be a matroid for $i \in [k]$. We add dummy elements so that any feasible partition is a base partition. To describe this precisely, we denote $r = \sum_{i \in [k]} \mathrm{rank}(\mathcal{I}_i) - |E|$. We remark that $r \geq 0$ if E has a feasible partition, since $|E| = \sum_{i \in [k]} |I_i| \leq \sum_{i \in [k]} \mathrm{rank}(\mathcal{I}_i)$ holds for any feasible partition $(I_1, \ldots, I_k)$. Then let $D = \{d_1, \ldots, d_r\}$ be a set of dummy elements. Note that $E \cap D = \emptyset$. We define two matroids $\mathcal{M}'_i = (D, \mathcal{I}'_i)$ and $\overline{\mathcal{M}}_i = (E \cup D, \overline{\mathcal{I}}_i)$ for each $i \in [k]$ by $\mathcal{I}'_i = \{D' \subseteq D \,:\, |D'| \leq \mathrm{rank}(\mathcal{I}_i) - 1\}$ and $\overline{\mathcal{I}}_i = \{I \cup D' \,:\, I \in \mathcal{I}_i,\ D' \in \mathcal{I}'_i,\ |I \cup D'| \leq \mathrm{rank}(\mathcal{I}_i)\}$.

[1] We remark that the condition $I_i \neq \emptyset$ $(\forall i \in [k])$ is imposed to make the objective function well-defined. Moreover, if we define $\max_{e \in \emptyset} w_i(e) = 0$, $\min_{e \in \emptyset} w_i(e) = \infty$, and $\sum_{e \in \emptyset} w_i(e) = 0$, then we can reduce the problem where empty sets are allowed to our problem by adding dummy elements.

Namely, $\mathcal{M}'_i$ is a uniform matroid of rank $(\mathrm{rank}(\mathcal{I}_i)-1)$, and $\overline{\mathcal{M}}_i$ is the $\mathrm{rank}(\mathcal{I}_i)$-truncation of the matroid union $\mathcal{M}_i \vee \mathcal{M}'_i$. Then, we have the following proposition.

▶ **Proposition 2.** *For any $(E, (\mathcal{I}_i, w_i)_{i\in[k]})$, its minimum $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-value is the same as the minimum $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-value for $(E \cup D, (\overline{\mathcal{I}}_i, \overline{w}_i)_{i\in[k]})$, where*

$$\overline{w}_i(e) = \begin{cases} w_i(e) & (e \in E), \\ \min_{e\in E} w_i(e) & (e \in D,\ \mathrm{Op}^{(2)} = \max), \\ \max_{e\in E} w_i(e) & (e \in D,\ \mathrm{Op}^{(2)} = \min), \\ 0 & (e \in D,\ \mathrm{Op}^{(2)} = \sum). \end{cases}$$

We remark that the same property holds for the maximization problem.

In the following, we assume $|E| = \sum_{i\in[k]} \mathrm{rank}(\mathcal{I}_i)$. We next show that the maximization problems are reducible to the minimization ones.

▶ **Proposition 3.** *For any feasible partition $(I_1, \ldots, I_k)$ for $(E, \mathcal{I}_i)_{i\in[k]}$, it is an optimal solution for the minimum $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)})$-value matroid partitioning problem instance $(E, (\mathcal{I}_i, w_i)_{i\in[k]})$ if and only if it is optimal for the maximum $(\widetilde{\mathrm{Op}^{(1)}}, \widetilde{\mathrm{Op}^{(2)}})$-value matroid partitioning problem instance $(E, (\mathcal{I}_i, w'_i)_{i\in[k]})$, where $w^{\max} = \max_{i\in[k]} \max_{e\in E} w_i(e)$,*

$$\begin{cases} \widetilde{\min} = \max, \\ \widetilde{\max} = \min, \\ \widetilde{\sum} = \sum \end{cases} \quad \text{and} \quad w'_i(e) = \begin{cases} \frac{|E|\cdot w^{\max}}{\mathrm{rank}(\mathcal{I}_i)} - w_i(e) & (\mathrm{Op}^{(1)} \in \{\min, \max\}, \mathrm{Op}^{(2)} = \sum), \\ w^{\max} - w_i(e) & (\textit{otherwise}). \end{cases}$$

We note that these reductions above are not approximation factor preserving. Hence, the (in)approximability of the maximization problems are not deduced from that of the minimization problems.

3 The minimum $(\sum, \max)$-value matroid partitioning problem

In this section, we study the minimum $(\sum, \max)$-value matroid partitioning problem. We first deal with the case where the matroids and weights are respectively identical and then go to the general case.

3.1 Strong NP-hardness of the identical case

We first prove that the minimum $(\sum, \max)$-value matroid partitioning problem is strongly NP-hard even if the matroids and weights are respectively identical.

To prove this, we use the *densest l-subgraph* problem, which is known to be strongly NP-hard [10]. The densest l-subgraph problem is, given a graph G and an integer l, to find a subgraph of G induced on l vertices that contains the largest number of edges.

In our reduction, we use the following property on a partition matroid. Let $(E, \mathcal{I})$ be a partition matroid defined by $\mathcal{I} = \{I : |I \cap S_i| \leq \eta_i \ (i \in [p])\}$, where $(S_1, \ldots, S_p)$ is a partition of E, and $\eta_1, \ldots, \eta_p$ are positive integers. In addition, we assume that $|S_i| = \eta_i \cdot k$ for each $i \in [p]$ so that E can be partitioned into k bases of $\mathcal{I}$. Then, for any weight w, we can construct greedily an optimal partition to the instance $(E, (\mathcal{I}, w), k)$ of the minimum $(\sum, \max)$-value matroid partitioning problem.

▶ **Lemma 4** ([2])**.** *Let $(E, \mathcal{I})$ be any partition matroid with $|S_i| = \eta_i \cdot k$ ($\forall i \in [p]$), and let w be any weight. Let $I_{i,j}$ consist of η_i elements with the η_i largest weights in $S_i \setminus (\bigcup_{h=1}^{j-1} I_{i,h})$. Then $(\bigcup_{i\in[p]} I_{i,1}, \ldots, \bigcup_{i\in[p]} I_{i,k})$ is an optimal solution to $(E, (\mathcal{I}, w), k)$.*

Table 2 The weight of each element e_{ij}, where each row corresponds to i and each column corresponds to j.

$i \backslash j$	1	$\cdots$	$l-1$	l	$\cdots$	$l+2t-2$	$l+2t-1$	$l+2t$	$\cdots$	$l+2m-1$	$l+2m$	$\cdots$	$n+2m-1$
1	0	$\cdots$	0	0		$t-1$	$t-1$	t			m	$\cdots$	m
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$t-1$	$t-1$	t			$\vdots$		$\vdots$
u_t	$\vdots$		$\vdots$	$\vdots$		$t-1$	t	t			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$t-1$	$t-1$	t			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$t-1$	$t-1$	t			$\vdots$		$\vdots$
v_t	$\vdots$		$\vdots$	$\vdots$		$t-1$	t	t			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$t-1$	$t-1$	t			$\vdots$		$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$			$\vdots$		$\vdots$
n	0	$\cdots$	0	0		$t-1$	$t-1$	t			m	$\cdots$	m
$n+1$	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	0	$2m^2$	$\cdots$	$2m^2$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
$n+2m$	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	0	$2m^2$	$\cdots$	$2m^2$

▶ **Theorem 5.** *The minimum* $(\sum, \max)$*-value matroid partitioning problem is strongly NP-hard even if the matroids and weights are identical.*

Proof. Let $G = (V, F)$ be an instance of the densest l-subgraph problem. We denote $V = \{1, \dots, n\}$, $F = \{f_1, \dots, f_m\}$, and $f_i = \{u_i, v_i\}$. For any vertex set $T \subseteq V$, we denote $F[T] = \{\{u, v\} \in F : \{u, v\} \subseteq T\}$.

To solve the densest l-subgraph problem, it suffices to find a set of $n - l$ vertices such that the set of the other l vertices attain $\max_{T \subseteq V} |F[T]|$. We construct a matroid so that every feasible partition of the ground set corresponds to some set of $n - l$ vertices in V, and the $(\sum, \max)$-value is the number of edges in the induced subgraph by the other l vertices.

Let $V' = \{n+1, \dots, n+2m\}$ be a set of dummy vertices. For each $i \in V \cup V'$, we define a set E_i of $n + 2m - 1$ elements as $E_i = \{e_{ij} : j \in \{1, \dots, n + 2m - 1\}\}$. Let

$$E = \bigcup_{i=1}^{n+2m} E_i \quad \text{and} \quad \mathcal{I} = \{I \subseteq E : |I| \leq n + 2m - 1,\ |I \cap E_i| \leq 1\ (\forall i \in [n + 2m])\}.$$

The resulting matroid is denoted by $(E, \mathcal{I})$, which is a $(n + 2m - 1)$-truncation of a partition matroid. We set $k = n + 2m$. The weights of elements are defined as follows:

- for each $j = 1, \dots, l - 1$, set $w(e_{ij}) = 0 \quad (\forall i \in [n + 2m])$;
- for each $j = l+2m, \dots, n+2m-1$, set $w(e_{ij}) = m$ if $i \leq n$, and $w(e_{ij}) = 2m^2$ if $i \geq n+1$;
- set $w(e_{ij})$ $(j = l, l + 1, \dots, l + 2m - 1)$ as follows: for each $f_t = \{u_t, v_t\}$ $(t = 1, \dots, m)$,

$$w(e_{i,l+2t-2}) = \begin{cases} t-1 & (i \in [n]), \\ 0 & (i \geq n+1), \end{cases} \quad \text{and}$$

$$w(e_{i,l+2t-1}) = \begin{cases} t & (i \in \{u_t, v_t\}), \\ t-1 & (i \in [n] \setminus \{u_t, v_t\}), \\ 0 & (i \geq n+1). \end{cases}$$

The weight is illustrated in Table 2.

We remark that $|E| = (n + 2m)(n + 2m - 1)$. By the definition of the matroid, for every $i \in [n + 2m]$, all elements in E_i belong to different independent sets from each other. Thus, for any feasible partition of E, each independent set has $n + 2m - 1$ elements which consist of one element from each E_i except one set.

It remains to show that the resulting instance is equivalent to the densest l-subgraph problem instance $(G = (V, F), l)$.

▶ **Claim 6.** *Let $\alpha \in \{0, \ldots, m\}$. The graph G has a vertex set T^* with $|T^*| = l$ and $|F[T^*]| \geq \alpha$ if and only if there exists a feasible partition $(I_1, \ldots, I_k)$ of E with $(\sum, \max)$-value at most $2m^2(n-l) + m^2 + m - \alpha$.*

First, we assume that there exists $T^* \subseteq V$ such that $|T^*| = l$ and $|F[T^*]| \geq \alpha$. Without loss of generality, we assume that $T^* = \{1, \ldots, l\}$ and $V \setminus T^* = \{l+1, \ldots, n\}$. We show that there exists a partition such that its $(\sum, \max)$-value is at most $2m^2(n-l) + m^2 + m - \alpha$. We denote $E^j[p, q] = \{e_{p,j}, \ldots, e_{q,j}\}$. Let $J_1 = \{1, \ldots, l\}$, $J_2 = \{l+1, \ldots, l+2m\}$, and $J_3 = \{l+2m+1, \ldots, n+2m\}$. We construct a partition $(I_1^*, \ldots, I_{n+2m}^*)$ of E as follows:

$$I_j^* = \begin{cases} E^{j-1}[1, j-1] \cup E^j[j+1, n+2m] & (j \in J_1), \\ E^{j-1}[1, l] \cup E^j[l+1, n+2m+l-j] \cup E^{j-1}[n+2m+l-j+2, n+2m] & (j \in J_2), \\ E^{j-1}[1, j-2m-1] \cup E^j[j-2m+1, n] \cup E^{j-1}[n+1, n+2m] & (j \in J_3). \end{cases}$$

Then, the maximum weight of each independent set is

$$\max_{e \in I_j^*} w(e) = \begin{cases} 0 & (j \in J_1), \\ t-1 & (j = l+2t-1 \in J_2,\ t = 1, \ldots, m,\ \{u_t, v_t\} \in F[T^*]), \\ t & \left(\begin{array}{l} j = l+2t-1 \in J_2,\ t = 1, \ldots, m,\ \{u_t, v_t\} \notin F[T^*] \\ j = l+2t \in J_2,\ t = 1, \ldots, m \end{array} \right), \\ 2m^2 & (j \in J_3). \end{cases}$$

Thus, the $(\sum, \max)$-value is at most $0 \cdot l + \sum_{t=1}^{m}(2t) - |F[T^*]| + 2m^2 \cdot (n-l) \leq 2m^2(n-l) + m^2 + m - \alpha$.

Conversely, we assume that there exists a feasible partition $(I_1, \ldots, I_k)$ of E such that $\max_{e \in I_1} w(e) \leq \cdots \leq \max_{e \in I_k} w(e)$, and $\sum_{j \in [k]} \max_{e \in I_j} w(e) \leq 2m^2(n-l) + m^2 + m - \alpha$. All elements in E_k must be contained in different I_j's from each other by definition of $(E, \mathcal{I})$. Hence at least $n-l$ sets contain elements e with $w(e) = 2m^2$. If $\max_{e \in I_j} w(e) \geq 2m^2$ holds for some $j \leq l+2m$, then the objective value is at least $2m^2(n-l+1) > 2m^2(n-l) + m^2 + m - \alpha$. Thus, each of $I_{l+2m+1}, \ldots, I_k$ contains $2m$ elements with weight $2m^2$, and none of $I_1, \ldots, I_{l+2m}$ contains such elements. Let $U = \{i : |E_i \cap I_j| = 0\ (\exists j \in \{l+2m+1, \ldots, k\})\}$. Note that $|U| = n-l$ and $U \subseteq \{1, \ldots, n\}$. Here, we have $2m^2(n-l) + m^2 + m - \alpha \geq \sum_{j \in [k]} \max_{e \in I_j} w(e) = 2m^2(n-l) + \sum_{j \in [l+2m]} \max_{e \in I_j} w(e)$. In order to obtain a lower bound of $\sum_{j \in [l+2m]} \max_{e \in I_j} w(e)$, we define $E' = \{e_{ij} : i \in U,\ j = 1, \ldots, l+2m\}$. Let $(E', \mathcal{I}')$ be a partition matroid where $\mathcal{I}' = \{I' : |I' \cap E_i| \leq 1\ (\forall i \in U)\}$. We observe that $\sum_{j \in [l+2m]} \max_{e \in I_j} w(e) \geq \sum_{j \in [l+2m]} \max_{e \in I_j \cap E'} w(e)$, and $(I_1 \cap E', \ldots, I_{l+2m} \cap E')$ is a feasible partition to the $(\sum, \max)$ problem instance $(E', (\mathcal{I}', w), l+2m)$. By Lemma 4, an optimal solution to $(E', (\mathcal{I}', w), l+2m)$ can be obtained by a greedy algorithm. Let $(I_1', \ldots, I_{l+2m}')$ be an output solution of the greedy algorithm. Then we have

$$\begin{aligned} \sum_{j \in [l+2m]} \max_{e \in I_j} w(e) \geq \sum_{j \in [l+2m]} \max_{e \in I_j'} w(e) &= m + \sum_{l=1}^{m} 2(l-1) + |\{\{u, v\} : |\{u, v\} \cap U| \geq 1\}| \\ &\geq m^2 + m - |F[V \setminus U]|. \end{aligned}$$

This implies $|F[V \setminus U]| \geq \alpha$. Therefore, $T = V \setminus U$ is a vertex set with $|T| = l$ and $|F[T]| \geq \alpha$. This proves the theorem. ◀

Note that the matroid $(E, \mathcal{I})$ in the above proof is graphic because it can be seen as a matroid corresponding to a cycle with $n+2m$ vertices and each adjacent vertices is connected by $n+2m-1$ multiple edges. Thus, the maximum total capacity spanning tree partition problem is NP-hard.

3.2 PTAS for the identical case

In this subsection, we provide a PTAS for the minimum $(\sum, \max)$-value matroid partitioning problem with identical matroids and weights. This is the best possible result (unless P=NP) because the problem is strongly NP-hard as we proved in the previous subsection.

We start with the following observation, which will be also useful in Section 3.4.

▶ **Proposition 7.** *Let $(E, (\mathcal{I}_i, w_i)_{i\in[k]})$ be any instance of the minimum $(\sum, \max)$-value matroid partitioning problem, and let $(I_1^*, \dots, I_k^*)$ be an optimal solution. When we know $\max_{e\in I_i^*} w_i(e)$ for all $i \in [k]$, we can easily compute a feasible partition $(I_1, \dots, I_k)$ such that $\sum_{i\in[k]} \max_{e\in I_i} w_i(e) \le \sum_{i\in[k]} \max_{e\in I_i^*} w_i(e)$.*

Proof. The feasible partitions for matroids $(E, \mathcal{I}_i|\{e : w_i(e) \le \max_{e^*\in I_i^*} w_i(e^*)\})_{i\in[k]}$ satisfy the condition. Thus, we can find one of them in polynomial time by Theorem 1. ◀

Let $(E, (\mathcal{I}, w), k)$ be a problem instance, and let $\varepsilon < 1/2$ be a positive number. We write $w^{\max} = \max_{e\in E} w(e)$. Let $(I_1^*, \dots, I_k^*)$ be an optimal solution.

The idea of the algorithm is to guess the maximum weights. Since the number of possibilities of the maximum weights is at most n^k, we can solve the problem by solving the feasibility of matroid partitioning problems n^k times. Thus, we can solve the problem efficiently when k is small, but not in polynomial time. In order to reduce the possibilities, we guess $\max_{e\in I_i^*} w(e)$ only for some i's. Without loss of generality, we assume that $\max_{e\in I_1^*} w(e) \ge \dots \ge \max_{e\in I_k^*} w(e)$. We define a set $J = \{i_1, \dots, i_s\}$ of indices by

$$i_j = \begin{cases} j & (j = 1, \dots, \lfloor 1/\varepsilon^2 \rfloor), \\ \lfloor (1+\varepsilon)^t/\varepsilon^2 \rfloor & (j = \lfloor 1/\varepsilon^2 \rfloor + t,\ t = 1, \dots, \lfloor \log_{1+\varepsilon}(k\varepsilon^2) \rfloor). \end{cases}$$

By definition, it holds that $1 = i_1 < i_2 < \dots < i_s \le k$, and $s = \lfloor 1/\varepsilon^2 \rfloor + \lfloor \log_{1+\varepsilon}(k\varepsilon^2) \rfloor$. Note that for any $j = \lfloor 1/\varepsilon^2 \rfloor + t$ and $t \ge 1$, we have

$$i_j - i_{j-1} \ge ((1+\varepsilon)^t/\varepsilon^2 - 1) - ((1+\varepsilon)^{t-1}/\varepsilon^2) = (1+\varepsilon)^{t-1}/\varepsilon - 1 \ge 1/\varepsilon - 1 > 1$$

as $\varepsilon < 1/2$. For notational convenience, we denote $i_0 = 0$ and $i_{s+1} = k+1$.

To reduce the number of possibilities more, we round the weights $w(e)$. For all $e \in E$, define

$$w'(e) = \begin{cases} \frac{(1+\varepsilon)^t w^{\max}}{k}\varepsilon & \left(\frac{(1+\varepsilon)^t w^{\max}}{k}\varepsilon \le w(e) < \frac{(1+\varepsilon)^{t+1} w^{\max}}{k}\varepsilon,\ t \in \{0, 1, \dots, \lfloor \log_{1+\varepsilon}(\frac{k}{\varepsilon}) \rfloor\}\right), \\ 0 & \left(w(e) < \frac{w^{\max}}{k}\varepsilon\right). \end{cases}$$

Our algorithm guesses $\max_{e\in I_{i_j}^*} w'(e)$ for each $i_j \in J$. We write u_j^* for the value. Then, it finds a feasible partition $(I_1, \dots, I_k)$ that satisfies $\max_{e\in I_1} w(e) \ge \dots \ge \max_{e\in I_k} w(e)$ and $\max_{e\in I_{i_j}} w'(e) \le u_j^*$ for all $i_j \in J$. The algorithm is summarized in Algorithm 1.

▶ **Theorem 8.** *Algorithm 1 is a PTAS algorithm for the minimum $(\sum, \max)$-value matroid partitioning problem with identical matroids and weights.*

Algorithm 1: PTAS for the $(\sum, \max)$ problem with identical matroids and weights

1 **foreach** $u_1, \ldots, u_s \in \{0\} \cup \left\{ \frac{(1+\varepsilon)^t w^{\max}}{k} \varepsilon \,:\, t = 0, \ldots, \lfloor \log_{1+\varepsilon}(k/\varepsilon) \rfloor \right\}$ *such that* $u_1 \geq \cdots \geq u_s$ **do**

2 find a partition $(I_1, \ldots, I_k)$ such that $I_i \in (\mathcal{I} | \{e \,:\, w'(e) \leq u_j\})$ for each $i_j \leq i < i_{j+1},\ j = 1, \ldots, s$ if such a partition exists;

3 **return** the best solution $(I_1, \ldots, I_k)$ among the obtained partitions;

Proof. Let $(I_1^*, \ldots, I_k^*)$ be an optimal solution to the problem and $(I_1, \ldots, I_k)$ be the output of Algorithm 1. Without loss of generality, we assume that $\max_{e \in I_1^*} w(e) \geq \cdots \geq \max_{e \in I_k^*} w(e)$. Let $u_j^* = \max_{e \in I_{i_j}^*} w'(e)$ for each $i_j \in J$.

We first analyze the running time of Algorithm 1.

▶ **Claim 9.** *Algorithm 1 runs in polynomial time with respect to k for fixed ε.*

Proof of Claim 9. Let $r = \lfloor \log_{1+\varepsilon}(k/\varepsilon) \rfloor + 2$. We observe that any choice of a possible combination of values $u_1, \ldots, u_s$ corresponds a multisubset of size s from the set of r values. Thus the number of possible combinations is $\binom{r+s-1}{s}$. Furthermore, we have

$$\binom{r+s-1}{s} \leq \sum_{l=0}^{r+s-1} \binom{r+s-1}{l} = 2^{r+s-1} \leq 2^{(\log_{1+\varepsilon}(k/\varepsilon)+2)+(1/\varepsilon^2+\log_{1+\varepsilon}(k\varepsilon^2))}$$
$$\leq 2^{2\log_{1+\varepsilon} k + 2 + 1/\varepsilon^2} = 2^{2+1/\varepsilon^2} \cdot k^{\log_{1+\varepsilon} 4}.$$

This is a polynomial with respect to k for fixed ε. Thus, the algorithm runs in polynomial time. ◀

Note that, without the restriction $u_1 \geq \cdots \geq u_s$, the number of possible combinations of values $u_1, \ldots, u_s$ is $r^s = k^{\Theta(\log \log k)}$, which is not polynomial with respect to k.

In the remainder, we show the approximation ratio of the algorithm.

▶ **Claim 10.** *Let* OPT *denote the optimal value and let* ALG *denote the* $(\sum, \max)$*-value of* $(I_1, \ldots, I_k)$*. Then it holds that* $\mathrm{ALG} \leq (1 + 15.5\varepsilon)\mathrm{OPT}$.

Proof of Claim 10. First, OPT is at least

$$\mathrm{OPT} = \sum_{i \in [k]} \max_{e \in I_i^*} w(e) \geq \sum_{i \in [k]} \max_{e \in I_i^*} w'(e) \geq \sum_{j=1}^{s} (i_j - i_{j-1}) u_j^* \ .$$

Let $(I_1', \ldots, I_k')$ be a feasible partition of E obtained at line 2 in Algorithm 1 using $u_1^*, \ldots, u_s^*$. Then ALG is at most

$$\begin{aligned} \mathrm{ALG} &= \sum_{i \in [k]} \max_{e \in I_i} w(e) \leq \sum_{i \in [k]} \max_{e \in I_i'} w(e) \\ &\leq \sum_{j=1}^{s} (i_{j+1} - i_j) \max_{e \in I_{i_j}'} w(e) \leq \sum_{j=1}^{s} (i_{j+1} - i_j) \left((1+\varepsilon) u_j^* + \frac{w^{\max}}{k} \varepsilon \right) \\ &\leq \sum_{j=1}^{s} (i_{j+1} - i_j)(1+\varepsilon) u_j^* + k \cdot \frac{w^{\max}}{k} \varepsilon \leq (1+\varepsilon) \sum_{j=1}^{s} (i_{j+1} - i_j) u_j^* + \varepsilon \cdot \mathrm{OPT}. \end{aligned} \quad (1)$$

Here, the third inequality holds by the definition of w' and $\max_{e \in I_{i_j}'} w'(e) \leq u_j^*$.

We derive an upper bound on $\sum_{j=1}^{s}(i_{j+1}-i_j)u_j^*$. To simplify notation, let $q=\lfloor 1/\varepsilon^2 \rfloor$. First, since $i_{j+1}-i_j = i_j - i_{j-1} = 1$ holds for any $j=1,\dots,q-1$, we have

$$\sum_{j=1}^{q-1}(i_{j+1}-i_j)u_j^* = \sum_{j=1}^{q-1}(i_j-i_{j-1})u_j^*. \tag{2}$$

Second, we evaluate $(i_{q+1}-i_q)u_q^*$. Note that $i_q = q = \lfloor 1/\varepsilon^2 \rfloor$ and $i_{q+1}=\lfloor (1+\varepsilon)/\varepsilon^2 \rfloor$. Thus $i_{q+1}-i_q \le (1+\varepsilon)/\varepsilon^2 - (1/\varepsilon^2 - 1) = (1+\varepsilon)/\varepsilon$. Moreover, $u_q^* = \max_{e\in I_q^*} w'(e) \le \max_{e\in I_q^*} w(e) \le \mathrm{OPT}/q$, because $\mathrm{OPT} = \sum_{i\in[k]} \max_{e\in I_i^*} w(e) \ge \sum_{i\in[q]} \max_{e\in I_i^*} w(e) \ge q\cdot \max_{e\in I_q^*} w(e)$. We remark that $1/q = 1/\lfloor 1/\varepsilon^2 \rfloor \le 1/(1/\varepsilon^2 - 1) = \varepsilon^2/(1-\varepsilon^2) < \frac{4}{3}\varepsilon^2 < 2\varepsilon^2$ as $\varepsilon < 1/2$. Therefore, it follows that

$$(i_{q+1}-i_q)u_q^* \le 2\varepsilon(1+\varepsilon)\mathrm{OPT}. \tag{3}$$

Lastly, let $j\in\{q+1,\dots,s\}$, and let t (≥ 1) be the integer such that $i_j = \lfloor (1+\varepsilon)^t/\varepsilon^2 \rfloor$ (i.e., $t=j-q$). We observe that $i_j - i_{j-1} \ge (1+\varepsilon)^{t-1}/\varepsilon - 1$. In addition, we have

$$\begin{aligned} i_{j+1}-i_j &\le \left(\frac{(1+\varepsilon)^{t+1}}{\varepsilon^2}\right) - \left(\frac{(1+\varepsilon)^t}{\varepsilon^2} - 1\right) = \frac{(1+\varepsilon)^t}{\varepsilon} + 1 \\ &\le \frac{(1+\varepsilon)/\varepsilon + 1}{(1+\varepsilon)^0/\varepsilon - 1}\left(\frac{(1+\varepsilon)^{t-1}}{\varepsilon} - 1\right) \le \frac{1+2\varepsilon}{1-\varepsilon}(i_j - i_{j-1}) < (1+6\varepsilon)(i_j-i_{j-1}), \end{aligned}$$

where the second inequality holds since $\frac{(1+\varepsilon)^x/\varepsilon+1}{(1+\varepsilon)^{x-1}/\varepsilon-1}$ is monotone decreasing for $x\ge 1$ and the last inequality holds since $\varepsilon < 1/2$. Therefore, it follows that

$$\sum_{j=q+1}^{s}(i_{j+1}-i_j)u_j^* = \sum_{j=q+1}^{s}(1+6\varepsilon)(i_j-i_{j-1})u_j^*. \tag{4}$$

By combining (1), (2), (3), (4), together with $\varepsilon < 1/2$, we have

$$\begin{aligned} \mathrm{ALG} &\le (1+\varepsilon)\left((1+6\varepsilon)\sum_{j=1}^{s}(i_j-i_{j-1})u_j^* + 2\varepsilon(1+\varepsilon)\mathrm{OPT}\right) + \varepsilon\cdot\mathrm{OPT} \\ &\le (1+\varepsilon)\left((1+6\varepsilon)+2\varepsilon(1+\varepsilon)\right)\cdot\mathrm{OPT} + \varepsilon\cdot\mathrm{OPT} = (1+10\varepsilon+10\varepsilon^2+2\varepsilon^3)\mathrm{OPT} \\ &< (1+10\varepsilon+5\varepsilon+0.5\varepsilon)\mathrm{OPT} = (1+15.5\varepsilon)\mathrm{OPT}. \end{aligned}$$ ◀

3.3 Hardness of the general case

We show a stronger result than the NP-hardness of the minimum $(\sum,\max)$-value matroid partitioning problem by reducing the *set cover* problem. Given a set $V=[n]$ and a collection $\mathcal{S}=\{S_i\subseteq V : i\in[k]\}$, the set cover problem is to find a subset $\mathcal{S}'$ $(\subseteq \mathcal{S})$ of minimum cardinality such that $\mathcal{S}'$ covers V, i.e., $\bigcup_{S\in\mathcal{S}'} S = V$. It is known that the set cover problem cannot be approximated in polynomial time to within a factor of $o(\log k)$ unless P=NP [6, 22].

▶ **Theorem 11.** *Even if either matroids or weights, but not both, are identical, the minimum $(\sum,\max)$-value matroid partitioning problem cannot be approximated in polynomial time within a factor of $o(\log k)$, unless P=NP.*

3.4 Algorithm for the general case

In this subsection, we provide an εk-approximation algorithm for any $\varepsilon > 0$. Let $(E,(\mathcal{I}_i,w_i)_{i\in[k]})$ be an instance of the minimum $(\sum,\max)$-value matroid partitioning problem, and let $(I_1^*,\dots,I_k^*)$ be any optimal partition.

Similarly to the PTAS described in Section 3.2, our algorithm guesses $\max_{e\in I_i^*} w_i(e)$ for each $i\in[k]$. In order to reduce the number of possibilities, we only guess top-$\lceil 1/\varepsilon \rceil$

weights of $\max_{e \in I_i^*} w_i(e)$. For simplicity, let $r = \lceil 1/\varepsilon \rceil$. Let $J^* = \{i_1, \ldots, i_r\}$ be the indices of top-r weights, i.e., $\max_{e \in I_i^*} w_i(e) \geq \max_{e \in I_j^*} w_i(e)$ for any $i \in J^*$ and $j \in [k] \setminus J^*$. Let $u_i^* = \max_{e \in I_i^*} w_i(e)$ for each $i \in J^*$. Then it finds a feasible partition $(I_1, \ldots, I_k)$ that satisfies $\max_{e \in I_i} w_i(e) \leq u_i^*$ for $i \in J^*$ and $\max_{e \in I_i} w_i(e) \leq \min_{j \in J^*} u_j^*$ for $i \in [k] \setminus J^*$.

▶ **Theorem 12.** *For any positive fixed number $\varepsilon > 0$, there exists a polynomial-time εk-approximation algorithm for the minimum $(\sum, \max)$-value matroid partitioning problem.*

4 Complexity of other optimal matroid partitioning problems

In this section, we prove the other results in Table 1. We first deal with the cases (1) $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)}) = (\min, \min)$, $(\max, \max)$, $(\min, \max)$ or $(\min, \sum)$; (2) $(\mathrm{Op}^{(1)}, \mathrm{Op}^{(2)}) = (\max, \min)$ or $(\sum, \min)$ with identical matroids. For the $(\min, \min)$, $(\max, \max)$, $(\min, \max)$ and $(\min, \sum)$ problems, we show polynomial-time reductions to the matroid partitioning problem. Then we can see that these are polynomially solvable by Theorem 1.

▶ **Theorem 13.** *The minimum $(\min, \min)$-value matroid partitioning problem is solvable in polynomial time.*

▶ **Theorem 14.** *The minimum $(\max, \max)$ and $(\min, \max)$-value matroid partitioning problems $(E, (\mathcal{I}_i, w_i)_{i \in [k]})$ are solvable in polynomial time.*

▶ **Theorem 15.** *The minimum $(\min, \sum)$-value matroid partitioning problem $(E, (\mathcal{I}_i, w_i)_{i \in [k]})$ is solvable in polynomial time.*

Next we consider the $(\max, \min)$ case and the $(\sum, \min)$ case. As we will see later, the optimal matroid partitioning problems for these cases are (strongly) NP-hard even to approximate. We provide polynomial-time algorithms for instances where matroids are identical (weights may differ). The following lemma plays the crucial role for this purpose.

▶ **Lemma 16.** *Let $(E, \mathcal{I})$ be a matroid. If there is a partition $(I_1, \ldots, I_k)$ of E such that $I_i \in \mathcal{I}$ for all $i \in [k]$, then for any k elements $e_1, \ldots, e_k \in E$, there is a partition $(I'_1, \ldots, I'_k)$ of E such that $e_i \in I'_i \in \mathcal{I}$ for all $i \in [k]$,*

We will reduce the problem of finding an optimal partition to the minimum weight perfect bipartite matching problem. It is well-known that this problem is solvable in polynomial time (see, e.g., [16, 23] for basic algorithms). Now we are ready to prove the theorem.

▶ **Theorem 17.** *The minimum $(\max, \min)$ and $(\sum, \min)$-value matroid partitioning problems with identical matroids $(E, (\mathcal{I}, w_i)_{i \in [k]})$ are solvable in polynomial time.*

Proof. Let $(E, \mathcal{I})$ be any matroid. Recall that the existence of a feasible partition is checkable in polynomial time by Theorem 1. Hence, in what follows, we assume that $(E, (\mathcal{I}, w), k)$ has a feasible partition.

We first consider the $(\max, \min)$ problem. By Lemma 16, the minimum $(\max, \min)$-value is at most w if and only if the bipartite graph $(E, [k], \{(e, i) : w_i(e) \leq w\})$ has a right-perfect matching. Thus, we can get the optimal value in polynomial time by setting w for all $\{w_i(e) : i \in [k],\ e \in E\}$ and checking the existence of a right-perfect matching.

Next, we consider the $(\sum, \min)$ problem. By Lemma 16, the minimum $(\sum, \min)$-value is the minimum weight of right-perfect matchings in the weighted bipartite graph $(E, [k], E \times [k]; w)$, where weight w is defined as $w(e, i) = w_i(e)$ for each $(e, i) \in E \times [k]$. Thus, we can find the optimal value in polynomial time. ◀

In addition, for the $(\max, \min)$ case and the $(\sum, \min)$ case, we prove the following hardness result by a reduction from *SAT*, which is an NP-complete problem [12].

▶ **Theorem 18.** *The minimum* $(\max, \min)$ *and* $(\sum, \min)$*-value matroid partitioning problems are both strongly NP-hard. Moreover, there exists no approximation algorithm for the problems unless P=NP.*

References

1 L. Babel, H. Kellerer, and V. Kotov. The k-partitioning problem. *Mathematical Methods of Operations Research*, 47:59–82, 1998.

2 R. E. Burkard and E. Yao. Constrained partitioning problems. *Discrete Applied Mathematics*, 28:21–34, 1990.

3 S.-P. Chen, Y. He, and G. Lin. 3-partitioning for maximizing the minimum load. *Journal of Combinatorial Optimization*, 6:67–80, 2002.

4 M. Dell'Amico and S. Martello. Bounds for the cardinality constrained $p||c_{\max}$ problem. *Journal of Scheduling*, 4:123–138, 2001.

5 P. Dell'Olmo, P. Hansen, S. Pallottino, and G. Storchi. On uniform k-partition problems. *Discrete Applied Mathematics*, 150:121–139, 2005.

6 I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, pages 624–633. ACM, 2014.

7 J. Edmonds. Minimum partition of a matroid into independent subsets. *JOURNAL OF RESEARCH of the National Bureau of Standards—B. Mathematicsnd Mathematical Physics*, 69B:67–72, 1965.

8 J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications*, pages 69–87. Gordon and Breach, New York, 1970.

9 J. Edmonds and D. R. Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards*, 69B:147–153, 1965.

10 U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

11 A. Frank. A weighted matroid intersection algorithm. *Journal of Algorithms*, 2(4):328–336, 1981.

12 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman New York, 1979.

13 Y. He, Z. Tan, J. Zhu, and E. Yao. k-partitioning problems for maximizing the minimum load. *Computers and Mathematics with Applications*, 46:1671–1681, 2003.

14 Y. Kawase, K. Kimura, K. Makino, and H. Sumita. Optimal Matroid Partitioning Problems. *arXiv preprints cs.DS/1710.00950*, October 2017. `arXiv:1710.00950`.

15 H. Kellerer and G. Woeginger. A tight bound for 3-partitioning. *Discrete Applied Mathematics*, 45:249–259, 1993.

16 B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2002.

17 E. L. Lawler. Matroids with parity conditions: A new class of combinatorial optimization problems. Memo erl-m334, Electronics Research Laboratory, College of Engineering, UC Berkeley, Berkeley, CA, 1971.

18 J. Lee, M. Sviridenko, and J. Vondrák. Matroid matching: The power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013.

19 J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990.

20 W. Li and J. Li. Approximation algorithms for k-partitioning problems with partition matroid constraint. *Optimization Letters*, 8:1093–1099, 2014.

21 L. Lovász. Matroid matching and some applications. *Journal of Combinatorial Theory, Series B*, 28(2):208–236, 1980.

22 D. Moshkovitz. The projection games conjecture and the NP-hardness of ln n-approximating set-cover. *Theory of Computing*, 11(7):221–235, 2015.

23 A. Schrijver. *Combinatorial Optimization*. Springer, 2003.

24 J. Verschae and A. Wiese. On the configuration-LP for scheduling on unrelated machines. *Journal of Scheduling*, 17:371–383, 2014.

25 B. Wu and E. Yao. k-partitioning problems with partition matroid constraint. *Theoretical Computer Science*, 374:41–48, 2007.

26 B. Wu and E. Yao. Lower bounds and modified LPT algorithm for k-partitioning problems with partition matroid constraint. *Applied Mathematics-A Journal of Chinese Universities*, 23:1–8, 2008.