Multi-objective Matroid Optimization with Ordinal Weights

Kathrin Klamroth[a], Michael Stiglmayr[a], Julia Sudhoff[a,*]

[a]University of Wuppertal, Gaußstr. 20, 42119 Wuppertal, Germany

Declaration of Interest: none

Abstract

Bi-objective optimization problems on matroids are in general intractable and their corresponding decision problems are in general NP-hard. However, if one of the objective functions is restricted to binary cost coefficients the problem becomes efficiently solvable by an exhaustive swap algorithm. Binary cost coefficients often represent two categories and are thus a special case of ordinal coefficients that are in general non-additive.

In this paper we consider ordinal objective functions with more than two categories in the context of matroid optimization. We introduce several problem variants that can be distinguished w.r.t. their respective optimization goals, analyze their interrelations, and derive a polynomial time solution method that is based on the repeated solution of matroid intersection problems. Numerical tests on minimum spanning tree problems and on partition matroids confirm the efficiency of the approach.

Keywords: matroid intersection, multi-objective combinatorial optimization, ordinal weights, multi-objective minimum spanning tree

1. Introduction

Matroid optimization problems, especially the minimum spanning tree problem, are well investigated even for multi-objective optimization (cf. Ehrgott (1996) for matroids and Ehrgott and Klamroth (1997), Hamacher and Ruhe (1994), Chen et al. (2007), Arroyo et al. (2008), Davis-Moradkhan and Browne (2008) among others for spanning trees). In this paper, we consider multi-objective optimization problems where, in addition to one sum objective function, one or more ordinal objective functions are to be considered.

---

*Corresponding author

Email addresses: klamroth@math.uni-wuppertal.de (Kathrin Klamroth), stiglmayr@math.uni-wuppertal.de (Michael Stiglmayr), sudhoff@math.uni-wuppertal.de (Julia Sudhoff)

Ordinal coefficients occur whenever there is no numerical value to reflect the quality or cost of an element. Consider, for example, the minimum spanning tree problem which allows to find connected networks with small overall connection costs. In addition to the (non-negative integer-valued and additive) length of an edge that directly represents the cost of, e.g., a telephone cable, the construction may involve major work that affects road traffic or even public transportation systems like tramways or trains. Roads that have already a cable canal allow for a cheap and easy addition of another cable. But if it is necessary to build a new cable canal under a street, this could lead to limitations of the traffic or even affect public transportation systems. In this situation, it is useful to categorize each possible connection as "easy to build" (good), "leads to little problems with traffic jams" (medium), and "leads to major problems for public transportation" (bad), for the time of construction. It is then hard to compare, for example, a solution with two medium edges with a solution with one bad edge, since in general these categories can not be translated into monetary values.

Ordinal weights and ordinal objective functions have been first introduced in Schäfer et al. (2020) for shortest path problems. Motivated by applications in civil security, edges are categorized, for example, as "secure", "neutral", or "insecure". Schäfer et al. (2020) introduce an ordinal preorder based on ordinal weights, analyze the complexity of the problem, and suggest a polynomial time labeling algorithm for its solution. Knapsack problems with ordinal weights are analyzed in Schäfer et al. (2021). They consider a general vector dominance and two lexicographic dominance concepts and suggest a dynamic programming based solution strategy and efficient greedy methods, respectively. Moreover, an outlook to multi-objective versions of ordinal problems is provided.

In this paper we extend this concept by considering multi-objective problems that combine one "classical", sum objective function with possibly several additional ordinal objectives. We focus on multi-objective optimization problems with ordinal weights on matroids and relate the multi-objective formulation to a series of single-objective optimization problems on intersections of matroids. The latter can be efficiently solved by an algorithm from Edmonds (2003). For the special case of bi-objective problems and only two ordinal categories, we compare this approach to the *Exhaustive Swap Algorithm (ESA)* presented in Gorski et al. (2021) which is even more efficient due to the special problem structure. In Section 2 we review some basic concepts of matroid theory and multi-objective optimization.

The paper is organized as follows. In Section 2 we review the basic concepts from matroid optimization and from multi-objective optimization. We particularly focus on (partial) ordering relations for ordinal objective functions in the light of state-of-the-art references. Ordinal matroid optimization problems with only one ordinal objective function are discussed in Section 3. We show that ordinal matroid optimization problems can be solved by a greedy strategy, due to their special structure. Multi-objective matroid optimization problems with one sum objective and one ordinal objective are introduced in Section 4. Their relation to matroid intersection problems is analyzed in Section 5, yielding efficient polynomial time solution strategies for all considered problem variants.

The algorithms are numerically tested and compared at randomly generated instances of graphic matroids and of partition matroids in Section 6, and the paper is concluded with a short outlook on future research topics in Section 7.

2. Preliminaries

Since we combine matroid theory and multi-objective combinatorial optimization in this paper, this preliminaries section is divided into two parts. In the first subsection we summarize some basic definitions and results of matroid theory (for a self-contained introduction to matroid theory see, for example, Oxley (2011); Schrijver (2002); Edmonds (1971); Schrijver (2017)). In the second subsection we review basic concepts of multi-objective optimization with a particular focus on dominance relations. For a general introduction into multi-objective optimization see, e.g., Ehrgott (2005). A survey on multi-objective combinatorial optimization is given in Ehrgott and Gandibleux (2000).

2.1. Matroid Theory

Let E be a finite set and let $\mathcal{I} \subset 2^E$ be a subset of the power-set of E. The tuple $(E, \mathcal{I})$ is called a *matroid* if and only if the following three properties hold:

$$\emptyset \in \mathcal{I} \tag{1}$$

$$I \in \mathcal{I} \text{ and } J \subseteq I \implies J \in \mathcal{I} \tag{2}$$

$$I, J \in \mathcal{I} \colon |I| < |J| \implies \exists j \in J \setminus I \colon I \cup \{j\} \in \mathcal{I}. \tag{3}$$

The subsets $I \in \mathcal{I}$ are called *independent sets* while the subsets $D \in 2^E \setminus \mathcal{I}$ are called *dependent sets*. Moreover, if the tuple $(E, \mathcal{I})$ satisfies at least the conditions (1) and (2), then it is called an *independence system*.

Furthermore, all inclusion-wise maximal independent sets are called *bases*, and all inclusion-wise minimal dependent sets are called *circuits*. In the following, we write $\mathcal{X} := \{B \in \mathcal{I} : \nexists I \in \mathcal{I} : I \supsetneq B\}$ for the set of all bases of a matroid. All bases of a matroid have the same cardinality which is referred to as the *rank* of the matroid, see, for example, Oxley (2011). An important characteristic of matroids is the so-called *basis exchange property*:

$$\forall B_1, B_2 \in \mathcal{X} \ \ \forall b_1 \in B_1 \setminus B_2 \ \ \exists b_2 \in B_2 \setminus B_1 \ : \ (B_1 \cup \{b_2\}) \setminus \{b_1\} \in \mathcal{X}. \tag{4}$$

The *restriction* of a matroid $\mathcal{M} = (E, \mathcal{I})$ to a subset $S \subseteq E$ is defined as $\mathcal{M} - S := (E \setminus S, \mathcal{I}_S)$ with $\mathcal{I}_S = \{I \in \mathcal{I} : I \subseteq E \setminus S\}$. Another way to manipulate a given matroid $\mathcal{M}$ is the *contraction* of $\mathcal{M}$ by an independent set $I \in \mathcal{I}$ that is defined as $\mathcal{M}/I := (E \setminus I, \mathcal{K})$ with $\mathcal{K} = \{K \subseteq E \setminus I : K \cup I \in \mathcal{I}\}$.

To simplify the notation we define the set operations $S + e := S \cup \{e\}$ and $S - f := S \setminus \{f\}$ for $S \subseteq E$ and $e, f \in E$. Furthermore, let $S^c := E \setminus S$ denote the complement of S in E.

For an extensive list of examples of matroids we refer to Oxley (2011). The following three matroids are frequently considered and will be used for illustrations and numerical tests in this paper.

- **graphic matroid**: Let $G = (V, E)$ be an undirected graph, then $\mathcal{M} = (E, \mathcal{I})$ with $\mathcal{I} := \{I \subseteq E \colon (V, I) \text{ contains no cycle}\}$ is a matroid.

- **uniform matroid**: Let E be a finite set and $k \in \mathbb{N}_0$, then $\mathcal{M} = (E, \mathcal{I})$ with $\mathcal{I} := \{I \subseteq E \colon |I| \leq k\}$ is a matroid.

- **partition matroid**: Let $E = E_1 \cup E_2 \cup \ldots \cup E_k$ be the disjoint union of k finite sets and let $u_1, \ldots, u_k \geq 0$ be non-negative integers. Then $\mathcal{M} = (E, \mathcal{I})$ with $\mathcal{I} := \{I \subseteq E \colon |I \cap E_i| \leq u_i \ \forall 1 \leq i \leq k\}$ is a matroid.

An important concept that will prove very useful in this paper is the intersection of two matroids. Consider two matroids $\mathcal{M}_1 = (E, \mathcal{I}_1)$ and $\mathcal{M}_2 = (E, \mathcal{I}_2)$ over the same ground set E. Then the *matroid intersection* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is defined as the independence system $\mathcal{M}_1 \cap \mathcal{M}_2 := (E, \mathcal{I}_1 \cap \mathcal{I}_2)$.

It is important to note that a matroid intersection is not necessarily a matroid itself. A counter example is given in Figure 1. Let $G = (V, E_1 \cup E_2)$ be a graph with an edge set $E = E_1 \cup E_2$ that is partitioned into two subsets E_1, E_2 (e.g., green and red edges, respectively). Moreover, let $\mathcal{M}_1 = (E, \mathcal{I}_1)$ be the graphic matroid on G and let $\mathcal{M}_2 = (E, \mathcal{I}_2)$ be a partition matroid with $E_1 := \{[1, 2], [4, 5], [4, 6]\}$ the set of all green edges, $E_2 := \{[1, 3], [2, 3], [3, 4], [5, 6]\}$ the set of all red edges and $\mathcal{I}_2 := \{I \subseteq E \colon |I \cap E_1| \leq 3, |I \cap E_2| \leq 2\}$. Then the set $I := \{[1, 3], [2, 3], [4, 5], [4, 6]\}$ is an inclusion-wise maximal independent set of $\mathcal{M}_1 \cap \mathcal{M}_2$ since every additional edge $e \in E \setminus I$ makes $I \cup \{e\}$ dependent w.r.t. either $\mathcal{M}_1$ or $\mathcal{M}_2$. However, the set $J := \{[1, 2], [1, 3], [3, 4], [4, 5], [4, 6]\}$ is also a maximal independent set of $\mathcal{M}_1 \cap \mathcal{M}_2$ that has larger cardinality, i.e., $|J| > |I|$. Thus, $\mathcal{M}_1 \cap \mathcal{M}_2$ is not a matroid because all maximal independent sets of a matroid must have the same cardinality. However, if at least one of the intersected matroids is a uniform matroid then the intersection is again a matroid, see Oxley (2011).
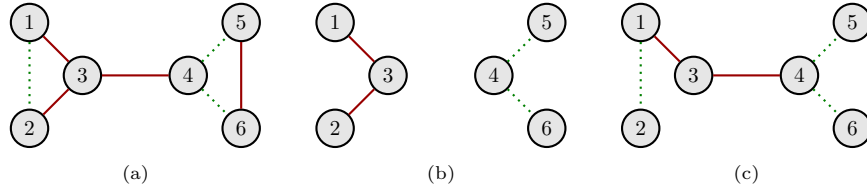

Figure 1: The edges $E = E_1 \cup E_2$ of the graph $G = (V, E)$ in (a) are the ground-set of a graphic matroid $\mathcal{M}_1$ and of a partition matroid $\mathcal{M}_2$ (where E_1 is the set of all green-dotted edges and E_2 is the set of all red-solid edges). The independent set of edges I illustrated in (b) is inclusion-wise maximal for $\mathcal{M}_1 \cap \mathcal{M}_2$, however, the alternative independent set J shown in (c) has larger cardinality.

2.2. Multi-objective Optimization

In this article we consider three different concepts of efficiency for multi-objective optimization. The first two are used for general, usually sum objective functions, while the last one is particularly defined for ordinal objective

functions. Throughout this section we consider minimization problems with a discrete feasible set $X \subset 2^E$ that is defined over a finite ground set E. However, all of the presented concepts can be defined analogously for maximization problems.

Pareto Optimality. The Pareto concept of optimality (see again, e.g., Ehrgott (2005)) is based on the component-wise order. Let $y^1, y^2 \in \mathbb{R}^p$ and define

$$y^1 \leqq y^2 :\Longleftrightarrow y_i^1 \leq y_i^2, \quad i = 1, \ldots, p,$$
$$y^1 \leqslant y^2 :\Longleftrightarrow y_i^1 \leq y_i^2, \quad i = 1, \ldots, p \text{ and } y^1 \neq y^2,$$
$$y^1 < y^2 :\Longleftrightarrow y_i^1 < y_i^2, \quad i = 1, \ldots, p.$$

We say that y^1 *dominates* y^2 whenever $y^1 \leqslant y^2$. Note that the binary relation $\leqq$ is reflexive, transitive and antisymmetric and hence a partial order. The binary relation $\leqslant$ is irreflexive, transitive and asymmetric and hence defines a strict partial order. Similarly, $<$ defines a strict partial order.

This concept of dominance is typically used when comparing outcome vectors of a *multi-objective problem* (MOP) that aims at minimizing a vector-valued objective function $w : X \to \mathbb{R}^p$:

$$\begin{aligned} \min \quad & w(x) \\ \text{s.t.} \quad & x \in X. \end{aligned} \tag{MOP}$$

A feasible solution $\bar{x} \in X$ is called *efficient* or *Pareto-optimal* for (MOP) if there exists no $x \in X$ with $w(x) \leqslant w(\bar{x})$. Moreover, a feasible solution $\hat{x} \in X$ is called *weakly efficient* or *weakly Pareto-optimal* for (MOP) if there exists no $x \in X$ with $w(x) < w(\hat{x})$. The set of all efficient solutions of a problem (MOP) is called *efficient set* and denoted by X_{eff}. The image $w(\bar{x})$ of an efficient solution $\bar{x}$ is called *non-dominated* outcome vector or non-dominated point. If $w(x') \leqslant w(x'')$ for two feasible solutions $x', x'' \in X$, then we say that x' *dominates* x'', in accordance with the above concept of dominance in the objective space. The image of the efficient set is called *non-dominated set* and denoted by $Y_{\text{nd}} := w(X_{\text{eff}})$.

Lexicographic Optimality. The concept of lexicographic optimality assumes a specific ordering among the components of the given (outcome) vectors, i.e., the first component is more important than the second, and so on. We refer again to Ehrgott (2005) for a more general introduction. Let $y^1, y^2 \in \mathbb{R}^p$. Then $y^1 <_{\text{lex}} y^2$, i.e., y^1 *lexicographically dominates* y^2, if there is an index $k \in \{1, \ldots, p\}$ such that $y_k^1 < y_k^2$ and $y_i^1 = y_i^2$ for all $i \in \{1, \ldots, k-1\}$. Furthermore, we write $y^1 \leqq_{\text{lex}} y^2$ if $y^1 <_{\text{lex}} y^2$ or $y^1 = y^2$. Note that the lexicographic order $\leqq_{\text{lex}}$ is a total order, i.e., it is reflexive, transitive and antisymmetric, and for all $y^1, y^2 \in \mathbb{R}^p$ with $y^1 \neq y^2$ either $y^1 \leqq_{\text{lex}} y^2$ or $y^2 \leqq_{\text{lex}} y^1$ holds. Consequently, we call a solution $\bar{x} \in X$ of (MOP) *lexicographically optimal* if $w(\bar{x}) \leqq_{\text{lex}} w(x)$ for all $x \in X$. To distinguish lexicographic optimization from multi-objective optimization in the Pareto sense we write lexmin (and lexmax in the case of maximization problems, respectively).
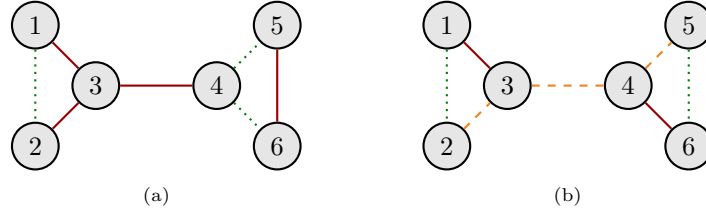
Figure 2: The edges of the graph in (a) are categorized w.r.t. two categories(green-dotted and red-solid), c. f. Figure 1(a) above, while the edges of the same graph are categorized w.r.t. three categories in (b) (green-dotted, orange-dashed, and red-solid).

Ordinal Optimality. Now assume that all solutions in the feasible set X have the same cardinality, i. e. $|x| = r$ for all $x \in X$. This is, for example, satisfied in the case of a matroid optimization problem. Then an *ordinal weight optimization problem* (OWOP) can be formulated as

$$\begin{aligned} \min \quad & o(x) \\ \text{s. t.} \quad & x \in X. \end{aligned} \tag{OWOP}$$

Intuitively, the ordinal objective function o assigns one out of K ordered categories to each element of the ground set E, and hence the ordinal objective of x is given by an r-dimensional ordinal vector. For example, in the case $K = 3$ we may think of good (green), medium (orange) and bad (red) elements, where we prefer good over medium and medium over bad. Figure 2 shows two examples of the ground set E of a graphic matroid. While the edges in Figure 2(a) are assigned to only two categories (where green-dotted is better than red-solid), Figure 2(b) shows an example with three categories (where green-dotted is better than orange-dashed which is again better than red-solid).

Throughout this paper we assume that the components of the ordinal vectors of feasible solutions are sorted in non-decreasing order w.r.t. the quality of the respective categories, see Figure 3 for an illustration. This sorting will be useful when comparing different solutions in the following.

More formally, let $\mathcal{C} = \{\eta_1, \ldots, \eta_K\}$ be an ordinal space consisting of K ordered categories, and let $o : E \to \mathcal{C}$ assign one ordinal category to each element of the ground set E. Moreover, (by slightly abusing the notation) let $o : X \to \mathcal{C}^r$ be a function mapping each feasible solution to an r-dimensional ordinal vector. We assume that category η_i with $i \in \{1, \ldots, K - 1\}$ is strictly preferred over all categories η_j with $i < j$, which is denoted by $\eta_i \prec \eta_j$. Similarly, we write $\eta_i \preceq \eta_j$ whenever $i \leq j$. Moreover, the components of the objective vector $o(x)$ of a feasible solution $x \in X$ are sorted in non-decreasing order, which is denoted by $o(x) := \text{sort}(o(x_1), \ldots, o(x_r))$. If we specify the vector o for an explicit example we often write i instead of η_i for better readability.

In order to define meaningful optimality concepts for problem (OWOP), we need to compare ordinal vectors in $\mathcal{C}^r$. The following definition is based on the concept first introduced in Schäfer et al. (2020) and Schäfer et al. (2021). Let

6

$y^1, y^2 \in \mathcal{C}^r$ be two ordinal vectors. Then we write

$$y^1 \preceq y^2 :\Longleftrightarrow y_i^1 \preceq y_i^2, \ i = 1, \ldots, r,$$
$$y^1 \preccurlyeq y^2 :\Longleftrightarrow y_i^1 \preceq y_i^2, \ i = 1, \ldots, r \text{ and } y^1 \neq y^2,$$
$$y^1 \prec y^2 :\Longleftrightarrow y_i^1 \prec y_i^2, \ i = 1, \ldots, r.$$

When $y^1 = o(x^1)$ and $y^2 = o(x^2)$ are outcome vectors of problem (OWOP), then their components are sorted in non-decreasing order. We use the same notation in this case, and we say that y^1 *ordinally dominates* y^2 whenever $y^1 \preccurlyeq y^2$. Note that what we consider here is a special case of the concept of ordinal dominance introduced in Schäfer et al. (2020) who considered the more general case when feasible solutions – and hence their outcome vectors – may differ w.r.t. their number of elements. Schäfer et al. (2020) showed that in this more general case, the binary relation $\preceq$ is a partial preorder on the set of sorted outcome vectors of an ordinal optimization problem, i.e., it is reflexive and transitive. In the special case when all feasible solutions have the same number of elements, as considered in this paper, the binary relation $\preceq$ is also antisymmetric and thus a partial order. See also Schäfer et al. (2021) for yet another perspective on ordinal efficiency. Moreover, the binary relation $\preccurlyeq$ is a strict partial order in our case, i.e., it is irreflexive, transitive and asymmetric. Thus, the concepts of (weak) ordinal efficiency and (weak) ordinal dominance can be defined in a similar way as for the case of Pareto optimality by replacing $\leqslant$ with $\preccurlyeq$ and $<$ with $\prec$, respectively.

Combined Orderings. If an optimization problem has an objective function that maps feasible solutions to outcome vectors on which several of the above orderings are combined – this may be the case when, for example, the first p components of an outcome vector represent sum objective functions that are ordered w.r.t. Pareto dominance, while the following r objective values represent ordinal values to which ordinal dominance is applied – then we say that a solution x' *dominates* a solution x'', if all objective values of x' are "at least as good" w.r.t. all components with at least one strict inequality in the respective ordering concept.

3. Ordinal Matroid Optimization

As a first step towards multi-objective ordinal optimization problems we investigate matroid optimization problems with only one ordinal objective function and show that such problems can be solved using a greedy algorithm. In slight abuse of the standard notation, we will refer to the resulting problems as "single-objective optimization problems", even though their objective functions are vector-valued. Similar results were obtained by Schäfer et al. (2020) and Schäfer et al. (2021) for shortest path and knapsack problems, respectively, however, for the case that a lexicographic optimization is employed on the ordinal outcome vectors. We show in the following that in the case of matroids, ordinal optimality actually coincides with lexicographic optimality, and hence a greedy algorithm always yields the ordinally non-dominated set in this case.

3.1. Ordinal and Lexicographic Optimality and their Interrelation

Let a matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$ with rank r be given and denote the set of its bases by $\mathcal{X}_1$. As a first special case, consider the situation of only two categories, i.e., $K = 2$. Wlog we set $\eta_1 = 0$ (green) and $\eta_2 = 1$ (red). It is easy to see that in this case problem (OWOP) is equivalent to a matroid optimization problem with a "classical" sum objective function with binary coefficients $b : E \to \{0, 1\}$ where the cost of a basis B is the aggregated cost of all of its elements, i.e. $b(B) := \sum_{e \in B} b(e)$. Indeed, a basis $B_1 \in \mathcal{X}_1$ ordinally dominates a basis $B_2 \in \mathcal{X}_1$ whenever the number of one-entries in $o(B_1)$ (i.e., red elements in B_1) is smaller than that in $o(B_2)$. This leads to a *matroid problem with a binary objective function* (BMP)

$$\begin{aligned} \min \quad & b(B) \\ \text{s.t.} \quad & B \in \mathcal{X}_1 \end{aligned} \tag{BMP}$$

as a particularly simple special case of problem (OWOP). When $K > 2$, i.e., when more than two ordinal categories have to be considered, a simple aggregation of all categories into one single aggregated objective value is no longer meaningful. However, we will discuss two related optimization problems that are based on partial aggregation in the following. Towards this end, let problem (OWOP) for the special case of matroid optimization be defined as a *matroid problem with ordinal costs*, given by

$$\begin{aligned} \min \quad & o(B) \\ \text{s.t.} \quad & B \in \mathcal{X}_1. \end{aligned} \tag{MPO}$$

Now consider a feasible basis $B \in \mathcal{X}_1$. Then the information contained in the objective vector $o(B) \in \mathcal{C}^r$ can equivalently be stored in an aggregated vector $c(B) \in \mathbb{Z}_{\geq}^K$ with components $c_j(B) := |\{e \in B : o(e) = \eta_j\}|$ for $j = 1, \ldots, K$ that count the number of elements in each category in B. Indeed, there is a simple one-to-one correspondence between $o(B)$ and $c(B)$. We will refer to c as a *counting objective function* in the following. This representation is often advantageous since in general K, i.e., the number of categories, is constant and much smaller than the dimension r, i.e., the number of elements in a basis. Note that since all bases have the same number of elements we have that $\sum_{j=1}^{K} c_j(B) = r$, and hence all outcome vectors $c(B)$, $B \in \mathcal{X}_1$, lie on the same hyperplane in $\mathbb{R}^K$. Moreover, one of the components of c can be omitted without loosing any information.

This reformulation suggests two related lexicographic optimization problems: On the one hand, we may aim at lexicographically maximizing the number of elements in the "good" categories, and on the other hand, we may want to lexicographically minimize the number of elements in the "bad" categories. In order to clearly distinguish between these two optimization goals, we introduce two separate variants of the counting objective c denoted as $c^{\max}$ and $c^{\min}$, respectively.

Maximizing the Number of Good Elements. When aiming at the maximization of the number of elements in *good* categories, we can apply a lexicographic

maximization to the counting objective c. Thus, in this case we set $c^{\max}(B) :=$ $c(B)$ for $B \in \mathcal{X}_1$ with $c_j^{\max}(B) := |\{e \in B : o(e) = \eta_j\}|$ for $j = 1, \dots, K$ as defined above, and formulate problem (MPCmax) as

$$\begin{aligned} \text{lexmax} \quad & c^{\max}(B) \\ \text{s.\,t.} \quad & B \in \mathcal{X}_1. \end{aligned} \qquad \text{(MPCmax)}$$

Minimizing the Number of Bad Elements. In order to lexicographically minimize the number of elements in *bad* categories, we first have to bring the corresponding entries of the counting objective c that represent the bad categories into the leading positions (which are always considered first in lexicographic optimization). We hence define $c_j^{\min}(B) := c_{K-j+1}(B)$ for $j = 1, \dots, K$ and for $B \in \mathcal{X}_1$, i.e., $c_j^{\min}(B) := |\{e \in B : o(e) = \eta_{K-j+1}\}|$, and consider problem (MPCmin) given by

$$\begin{aligned} \text{lexmin} \quad & c^{\min}(B) \\ \text{s.\,t.} \quad & B \in \mathcal{X}_1. \end{aligned} \qquad \text{(MPCmin)}$$

Figure 3 shows an example of a graphic matroid with all of its feasible bases and their respective objective vectors o, $c^{\max}$ and $c^{\min}$, see also Example 6 below.

3.2. Interrelation Between (MPO), (MPCmin) *and* (MPCmax)

In general, the ordinally non-dominated set of problem (MPO) is different from the sets of lexicographically optimal outcome vectors of the associated formulations (MPCmin) and (MPCmax), respectively. This can be seen, for example, at the cases of ordinal shortest path problems (see Schäfer et al. (2020)) and ordinal knapsack problems (see Schäfer et al. (2021)). In the special case of matroids, however, these three concepts are closely related and their respective efficient and non-dominated sets coincide.

Theorem 1. *Let $\mathcal{M}_1 = (E, \mathcal{I}_1)$ be a matroid, let $\mathcal{X}_1$ denote the set of bases of $\mathcal{M}_1$, and let the functions o, $c^{\min}$ and $c^{\max}$ be given and defined as above. Moreover, let $B_1, B_2 \in \mathcal{X}_1$ be two bases of $\mathcal{M}_1$. Then*

$$\left(o(B_1) \preccurlyeq o(B_2) \right) \Rightarrow \left(c^{\min}(B_1) <_{\text{lex}} c^{\min}(B_2) \text{ and } c^{\max}(B_1) >_{\text{lex}} c^{\max}(B_2) \right),$$

i.e., if $o(B_1)$ ordinally dominates $o(B_2)$, then $c^{\min}(B_1)$ lexicographically dominates $c^{\min}(B_2)$ and $c^{\max}(B_1)$ lexicographically dominates $c^{\max}(B_2)$.

Proof. We prove the result for $c^{\min}$. The corresponding result for $c^{\max}$ follows analogously, noting that (MPCmin) involves lexicographic minimization while (MPCmax) involves lexicographic maximization.

Now let $o(B_1) \preccurlyeq o(B_2)$ and assume that $c^{\min}(B_1)$ does not lexicographically dominate $c^{\min}(B_2)$. First note that $o(B_1) \preccurlyeq o(B_2)$ implies $o(B_1) \neq o(B_2)$ and hence $c^{\min}(B_1) \neq c^{\min}(B_2)$. Let $\tau := \min\{i : c_i^{\min}(B_1) \neq c_i^{\min}(B_2)\}$ be the smallest index where $c^{\min}(B_1)$ and $c^{\min}(B_2)$ differ. Since we assumed that $c^{\min}(B_1)$ does not lexicographically dominate $c^{\min}(B_2)$, it follows that $c_\tau^{\min}(B_1) > c_\tau^{\min}(B_2)$. Thus, the vectors $o(B_1)$ and $o(B_2)$ are equal in the last

$\ell := \sum_{i=1}^{\tau-1} c_i^{\min}(B_1) = \sum_{i=1}^{\tau-1} c_i^{\min}(B_2)$ components, i.e., $o_j(B_1) = o_j(B_2)$ for all $j = K - \ell + 1, \ldots, K$. Furthermore, it holds that $o_{K-\ell}(B_1) \succ o_{K-\ell}(B_2)$, which contradicts the assumption that $o(B_1)$ ordinally dominates $o(B_2)$. □

Note that while the proof of Theorem 1 relies on the fact that all feasible solutions have the same number of elements (and hence all outcome vectors have the same length), the matroid property is not used. Hence, Theorem 1 generalizes to all ordinal optimization problems with fixed length solutions. The following Corollary 2, that also follows from the results in Schäfer et al. (2020), is an immediate consequence of Theorem 1.

Corollary 2. *The set of efficient bases of* (MPO) *is a superset of the set of efficient bases of* (MPCmin) *and of* (MPCmax).

Proof. Theorem 1 implies that the efficient set of (MPCmin) can not contain any bases that are ordinally dominated w.r.t. o since this would imply that they are also lexicographically dominated w.r.t. $c^{\min}$. The same argument applies to (MPCmax). □

Remark 3. *The reverse implication of Theorem 1 does not hold in general, neither for $c^{\min}$ nor for $c^{\max}$. As a counter example consider the bases B_4 and B_6 from Figure 3. We have that $c^{\min}(B_6) = c^{\max}(B_6) = (2,1,2)$, $c^{\min}(B_4) = c^{\max}(B_4) = (1,3,1)$, $o(B_6) = (1,1,2,3,3)$ and $o(B_4) = (1,2,2,2,3)$. Hence, $c^{\min}(B_4)$ lexicographically dominates $c^{\min}(B_6)$ and $c^{\max}(B_6)$ lexicographically dominates $c^{\max}(B_4)$, while $o(B_4)$ and $o(B_6)$ are ordinally incomparable.*

We show in the following that in the case of matroids Corollary 2 can be strengthened. Indeed, the following result shows that the respective ordinal and lexicographic non-dominated sets are always equal and have cardinality one. This can also be observed in Example 6 below, where all three problems (MPO), (MPCmin) and (MPCmax) have the same efficient and non-dominated sets.

The result can be briefly summarized as follows: Corollary 1 states that the efficient set of (MPO) is a superset of that of (MPCmin) and (MPCmax). If there were two non-dominated bases B_1, B_2 for (MPO) and only one of them, say, basis B_1, was optimal for problem (MPCmin), then the basis exchange property would imply that basis B_2 could be improved w.r.t. $c^{\min}$ by an appropriate swap operation. However, this would lead to a basis that also ordinally dominates B_2, contradicting the ordinal efficiency of B_2. This leads to the following result:

Theorem 4. *Let $\mathcal{M}_1 = (E, \mathcal{I}_1)$ be a matroid, let $\mathcal{X}_1 \neq \emptyset$ be the set of bases of $\mathcal{M}_1$, and let the functions o, $c^{\min}$ and $c^{\max}$ be given as defined above. Then problems* (MPO), (MPCmin) *and* (MPCmax) *have the same efficient set, and the corresponding non-dominated sets have cardinality one.*

Proof. We show the equality of the efficient sets of (MPO) and (MPCmin). The equality of the efficient sets of (MPO) and (MPCmax) follows analogously.

First observe that the non-dominated set of problem (MPCmin) has cardinality one since the lexicographical order is a total order. Moreover, Theorem 1 implies that every efficient solution of (MPCmin) is also efficient for (MPO). Consequently, it is sufficient to show that all efficient solutions of (MPO) map to a unique non-dominated outcome vector. We prove this result by contradiction.

Suppose, to the contrary, that there are two efficient bases B_1 and B_2 for (MPO) with $o(B_1) \neq o(B_2)$ and hence also $c^{\min}(B_1) \neq c^{\min}(B_2)$. W.l.o.g. assume that $c^{\min}(B_1)$ lexicographically dominates $c^{\min}(B_2)$.

Let $e \in B_2 \setminus B_1$ be chosen such that $o(\hat{e}) \preceq o(e)$ for all $\hat{e} \in B_2 \setminus B_1$, i.e., e is an element of highest category among all elements in $B_2 \setminus B_1$. Then the basis exchange property (4) implies that there exists an element $e' \in B_1 \setminus B_2$ such that $B^* \coloneqq (B_2 \cup \{e'\}) \setminus \{e\} \in \mathcal{X}_1$, and the choice of e and the fact that $c^{\min}(B_1) <_{\text{lex}} c^{\min}(B_2)$ imply that $o(e') \preceq o(e)$. Now, if $o(e') \prec o(e)$, then B^* dominates B_2 w.r.t. o, contradicting the assumption. Otherwise, i.e., if $o(e') = o(e)$, then B^* has one more element in common with B_1 than B_2, and iterating this procedure at most r times eventually yields a swap where $o(e') \prec o(e)$. $\qquad\square$

Corollary 5. *The ordinally non-dominated set of* (MPO) *can be computed by a greedy algorithm.*

Proof. This follows immediately from Theorem 4 and the matroid properties, see also Hamacher and Ruhe (1994). $\qquad\square$

Note that, while Theorem 4 states that the non-dominated sets of problems (MPO), (MPCmin) and (MPCmax) have cardinality one, this does in general not transfer to the respective efficient sets. Indeed, the size of the efficient sets may grow exponentially with the problem size. As an example, consider instances with exponentially growing feasible sets and assume that all elements of E are in the same ordinal category. Then, all feasible solutions of a considered problem are both ordinally and lexicographically efficient.

4. Multi-objective Matroid Optimization

We extend the settings of the previous section and consider multi-objective matroid optimization problems (on a matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$ with rank r and set of bases $\mathcal{X}_1$) where we combine an ordinal objective with a sum objective function with non-negative integer coefficients $w : E \to \mathbb{Z}_{\geq}$. The cost of a basis $B \in \mathcal{X}_1$ w.r.t. this sum objective is given by $w(B) \coloneqq \sum_{e \in B} w(e)$.

4.1. Multi-objective Ordinal and Lexicographic Optimality and their Interrelation

If we add a sum objective function to the problems described in Section 3.1 above, we obtain the following four variants of bi- or multi-objective optimization problems involving additive as well as ordinal objective coefficients:

The *bi-objective matroid problem with a binary objective function* (BBMP)

$$\begin{array}{ll} \min & (w(B), b(B)) \\ \text{s.\,t.} & B \in \mathcal{X}_1, \end{array} \tag{BBMP}$$

the *multi-objective matroid problem with an ordinal objective function* (MMPO)

$$\begin{array}{ll} \min & (w(B), o(B)) \\ \text{s.\,t.} & B \in \mathcal{X}_1, \end{array} \tag{MMPO}$$

and two *multi-objective matroid optimization problems with a counting objective function* (MMPCmax) and (MMPCmin)

$$\begin{array}{ll} \min & w(B) \\ \text{lexmax} & c^{\max}(B) \\ \text{s.\,t.} & B \in \mathcal{X}_1 \end{array} \tag{MMPCmax}$$

$$\begin{array}{ll} \min & w(B) \\ \text{lexmin} & c^{\min}(B) \\ \text{s.\,t.} & B \in \mathcal{X}_1. \end{array} \tag{MMPCmin}$$

The problem (BBMP) is investigated in detail in Gorski (2010) and Gorski et al. (2021), where an exhaustive swap algorithm is presented that determines a minimal complete representation of the non-dominated set (i.e., all non-dominated points and one efficient solution for each of them) of (BBMP) in polynomial time. This assumes that an oracle can determine in polynomial time if a given subset $I \subseteq E$ is independent or not. This is, e.g., the case for graphic matroids, uniform matroids and partition matroids, see Gabow and Tarjan (1984).

The similarities and differences between the problems (MMPO), (MMPCmax) and (MMPCmin) are illustrated at the following example of a graphic matroid:

Example 6. *Consider the graphic matroid introduced in Figure 2(b). Its bases are enumerated and illustrated with their weight functions w, o, $c^{\min}$ and $c^{\max}$ in Figure 3. It is easy to see that, in accordance with Theorem 4, the unique efficient solution w.r.t. all of the individual objective functions o, $c^{\min}$ and $c^{\max}$ is the basis B_9.*

The corresponding multi-objective problems that additionally consider the sum objective function w all have larger non-dominated sets in this example. The respective non-dominated outcome vectors of the multi-objective problems that combine w with the objective functions o, $c^{\min}$ and $c^{\max}$, respectively, are highlighted in Figure 3 by printing the latter components, i.e., o, $c^{\min}$ and $c^{\max}$, in bold. Note that the basis B_1 is efficient in all three cases since it is the unique minimizer of w.

4.2. Interrelation Between (MMPO), (MMPCmin) *and* (MMPCmax)

When moving from optimization problems with only one ordinal objective function to multi-objective problems that additionally include a sum objective
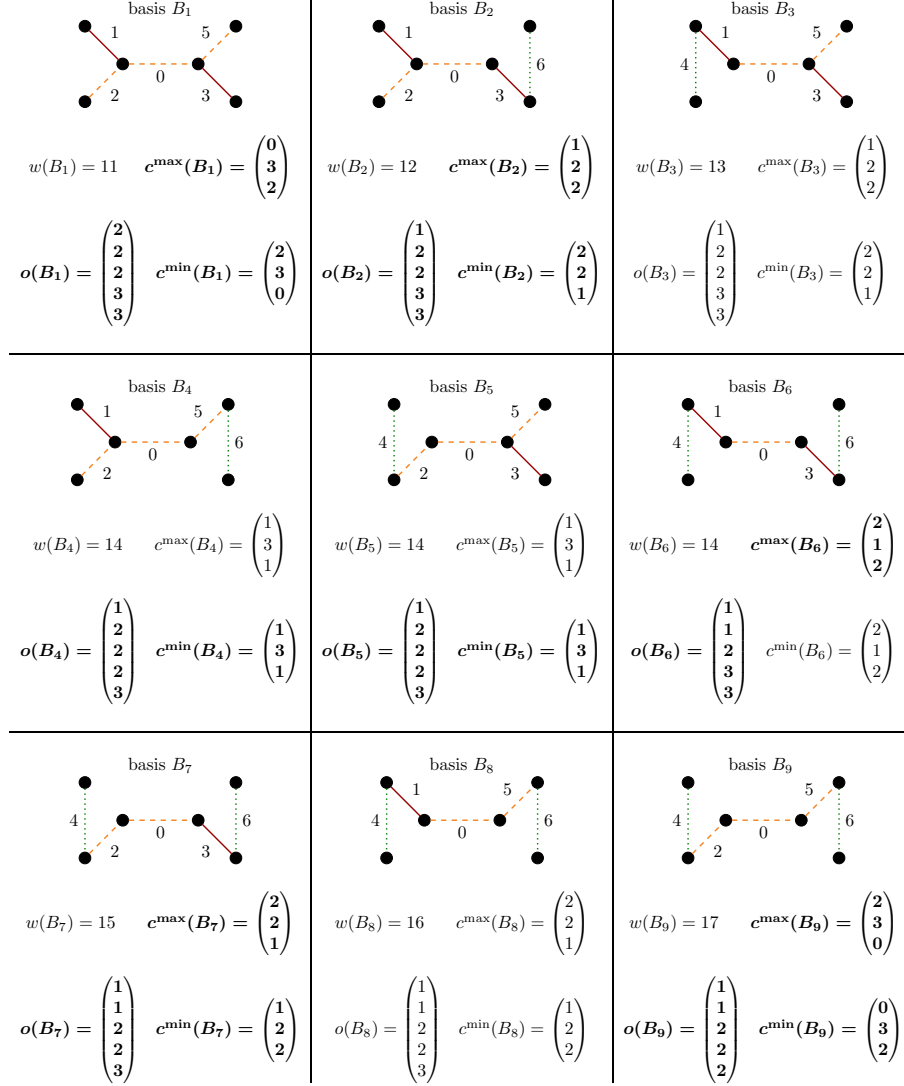
basis B_1

$$w(B_1) = 11 \qquad c^{\max}(B_1) = \begin{pmatrix} \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \end{pmatrix}$$

$$o(B_1) = \begin{pmatrix} \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{3} \end{pmatrix} \qquad c^{\min}(B_1) = \begin{pmatrix} \mathbf{2} \\ \mathbf{3} \\ \mathbf{0} \end{pmatrix}$$

basis B_2

$$w(B_2) = 12 \qquad c^{\max}(B_2) = \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{2} \end{pmatrix}$$

$$o(B_2) = \begin{pmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{3} \end{pmatrix} \qquad c^{\min}(B_2) = \begin{pmatrix} \mathbf{2} \\ \mathbf{2} \\ \mathbf{1} \end{pmatrix}$$

basis B_3

$$w(B_3) = 13 \qquad c^{\max}(B_3) = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

$$o(B_3) = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 3 \end{pmatrix} \qquad c^{\min}(B_3) = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

basis B_4

$$w(B_4) = 14 \qquad c^{\max}(B_4) = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

$$o(B_4) = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 3 \end{pmatrix} \qquad c^{\min}(B_4) = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

basis B_5

$$w(B_5) = 14 \qquad c^{\max}(B_5) = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

$$o(B_5) = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 3 \end{pmatrix} \qquad c^{\min}(B_5) = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}$$

basis B_6

$$w(B_6) = 14 \qquad c^{\max}(B_6) = \begin{pmatrix} \mathbf{2} \\ \mathbf{1} \\ \mathbf{2} \end{pmatrix}$$

$$o(B_6) = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{3} \end{pmatrix} \qquad c^{\min}(B_6) = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$$

basis B_7

$$w(B_7) = 15 \qquad c^{\max}(B_7) = \begin{pmatrix} \mathbf{2} \\ \mathbf{2} \\ \mathbf{1} \end{pmatrix}$$

$$o(B_7) = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{3} \end{pmatrix} \qquad c^{\min}(B_7) = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

basis B_8

$$w(B_8) = 16 \qquad c^{\max}(B_8) = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

$$o(B_8) = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \end{pmatrix} \qquad c^{\min}(B_8) = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

basis B_9

$$w(B_9) = 17 \qquad c^{\max}(B_9) = \begin{pmatrix} \mathbf{2} \\ \mathbf{3} \\ \mathbf{0} \end{pmatrix}$$

$$o(B_9) = \begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \end{pmatrix} \qquad c^{\min}(B_9) = \begin{pmatrix} \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \end{pmatrix}$$

Figure 3: All bases of the graphic matroid introduced in Figure 2(b) together with the objective values w, o, $c^{\min}$ and $c^{\max}$, where we write 1 for green-dotted, 2 for orange-dashed, and 3 for red-solid edges. When only considering the sum objective w, then B_1 is optimal, and when only considering the objective functions o, $c^{\min}$ or $c^{\max}$, respectively, then B_9 is the unique efficient basis. For the problems (MMPO), (MMPCmin) and (MMPCmax) that combine w with o, $c^{\min}$ and $c^{\max}$, respectively, the non-dominated outcome vectors are indicated by printing the partial objective vectors o, $c^{\min}$ and $c^{\max}$ in bold.

w, as in the multi-objective problems (MMPO), (MMPCmin) and (MMPCmax), the situation is much more complex than that described in Section 3.2 above. Indeed, while Corollary 2 can be adapted to the new situation, Theorem 4 does not transfer to the multi-objective case. A corresponding counter example will be given below.

Theorem 7. *The set of efficient bases of* (MMPO) *is a superset of the set of efficient bases of* (MMPCmin) *and of* (MMPCmax).

Proof. We prove the result for $c^{\min}$. The corresponding result for $c^{\max}$ follows analogously, noting that (MMPCmin) involves lexicographic minimization while (MMPCmax) involves lexicographic maximization.

We prove this result by contradiction. Hence, let $\bar{B}$ be an efficient basis for (MMPCmin) but not for (MMPO). Then there exists a basis B^* with $w(B^*) \leq w(\bar{B})$, $o(B^*) \leqq o(\bar{B})$, and $(w(B^*), o(B^*)) \neq (w(\bar{B}), o(\bar{B}))$. First note that $o(B^*) \leqq o(\bar{B})$ implies that $c^{\min}(B^*) \leqq_{\text{lex}} c^{\min}(\bar{B})$, by Theorem 1. We distinguish two cases: Either $w(B^*) < w(\bar{B})$ and $c^{\min}(B^*) \leqq_{\text{lex}} c^{\min}(\bar{B})$, or $w(B^*) = w(\bar{B})$ and $c^{\min}(B^*) <_{\text{lex}} c^{\min}(\bar{B})$. However, both cases are in contradiction with the efficiency of $\bar{B}$ for problem (MMPCmin). $\square$

However, as was to be expected, Theorem 4 does not generalize to the multi-objective case as is shown by the following counter example:

Example 8. *Consider again the graphic matroid introduced in Example 6 and the set of all of its bases illustrated in Figure 3. The efficient bases for problem* (MMPO) *are the bases* $B_1, B_2, B_4, B_5, B_6, B_7, B_9$, *while the efficient bases for the problem* (MMPCmin) *are given by* $B_1, B_2, B_4, B_5, B_7, B_9$, *and the efficient bases for problem* (MMPCmax) *are given by* B_1, B_2, B_6, B_7, B_9. *Hence, basis* B_6 *is efficient for* (MMPO) *but not for* (MMPCmin), *and the two bases* B_4 *and* B_5 *are efficient for* (MMPCmin) *but not for* (MMPCmax). *Thus, Theorem 4 does not generalize to the multi-objective problems* (MMPO), (MMPCmin) *and* (MMPCmax).

One could conjecture from Example 6 that every efficient basis for (MMPO) is efficient for at least for one of the problems (MMPCmin) or (MMPCmax). However, this also does not hold in general as the following example shows.

Example 9. *Consider the graphic matroid shown in Figure 4. We focus on all bases* $B \in \mathcal{X}_1$ *that have an objective value of* $w(B) = 4$ *in the sum objective. Note that these bases can only be dominated by other bases* $\hat{B}$ *with* $w(\hat{B}) \leq w(B)$, *and hence we restrict our analysis on those bases in Figure 4. First observe that all bases* $B \in \mathcal{X}_1$ *with* $w(B) = 4$ *map to one of the three possible outcome vectors* $o(B) \in \{(1,1,3,3),(1,2,2,3),(2,2,2,2)\}$, *which are all non-dominated for* (MMPO). *Their corresponding counting vectors* $c^{\min}$ *are* $(2,0,2)$, $(1,2,1)$ *and* $(0,4,0)$, *where the last one is the only one that is lexicographically non-dominated. For* $c^{\max}$ *the counting vectors are the same, but the first one is lexicographically non-dominated. Consequently, the counting vector* $(2,0,2)$ *is neither lexicographically non-dominated for* (MMPCmin) *nor for* (MMPCmax), *but it is non-dominated for* (MMPO).
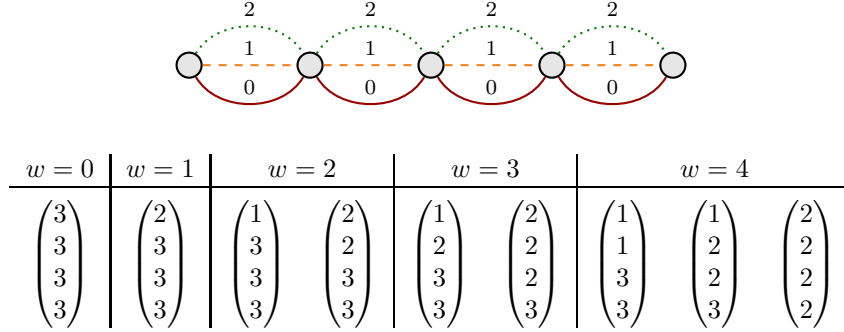
Figure 4: All possible outcome vectors $o(B)$ with $w(B) \in \{0,\ldots,4\}$ for a graphic matroid with non-negative integer-valued costs w and three categories (1:green-dotted, 2:orange-dashed and 3:red-solid).

5. Matroid Intersection for Ordinal Constraints

In the following we show that the three problems (MMPO), (MMPCmin) and (MMPCmax) can be solved using a series of matroid intersection problems. The approach is based on variants of ε-constraint scalarizations of problem (MMPO) with appropriately selected optimization objective and constraints. Furthermore, we show that matroid intersection problems can be used to solve even problems with several ordinal objective functions and one sum objective.

5.1. Variants of ε-Constraint Scalarizations

We consider an equality-constrained scalarization of (MMPO) (where equality constraints are used rather than inequality constraints as is commonly the case in ε-constraint scalarizations), given by

$$
\begin{aligned}
\min \quad & w(B) \\
\text{s.t.} \quad & o_i(B) = \varepsilon_i, \quad i = 1,\ldots,r \\
& B \in \mathcal{X}_1
\end{aligned}
\tag{5}
$$

with right-hand side vector $\varepsilon \in \mathcal{C}^r$. Intuitively, problem (5) specifies *exactly* how many elements of each category must be chosen, and hence each feasible basis $B \in \mathcal{X}_1$ of (5) maps to the same ordinal vector $o(B)$. Depending on the choice of ε, problem (5) may be infeasible (if there is no $B \in \mathcal{X}_1$ with $o(B) = \varepsilon$), yield an efficient solution B^* for (MMPO) (if there is *no* $B \in \mathcal{X}_1$ with $w(B) = w(B^*)$ and $o(B) \prec o(B^*)$), or yield a dominated solution $\hat{B}$ for (MMPO) (if there *is* a $B \in \mathcal{X}_1$ with $w(B) = w(\hat{B})$ and $o(B) \prec o(\hat{B})$). Note that suitable choices for ε satisfy $\varepsilon_1 \preceq \cdots \preceq \varepsilon_r$ since the components of $o(B)$ are always in non-decreasing order and hence problem (5) is certainly infeasible otherwise. In the following, we denote all such *suitable right-hand-side vectors* by $\Upsilon := \{\varepsilon \in \mathcal{C}^r : \varepsilon_1 \preceq \cdots \preceq \varepsilon_r\}$.

15

Since problem (5) can be interpreted as a variant of the "classical" ε-constraint scalarization in multi-objective optimization, see, e.g., Ehrgott (2005), the following result is not surprising and follows basically by the same arguments.

Theorem 10. *The non-dominated set of problem* (MMPO) *can be determined by solving problem* (5) *for all suitable right-hand side vectors $\varepsilon \in \Upsilon$ and filtering out all dominated outcome vectors.*

Proof. Let $(w(B^*), o(B^*))$ be a non-dominated outcome vector for (MMPO) with pre-image $B^* \in \mathcal{X}_1$. Then B^* is optimal for problem (5) with $\varepsilon := o(B^*) \in \Upsilon$. Thus, every non-dominated outcome vector of problem (MMPO) can be determined by solving an appropriate scalarization (5). The non-dominated set is then obtained by employing a dominance filtering to the set of all obtained outcome vectors. $\qquad\square$

Now let a suitable constraint vector $\varepsilon \in \Upsilon$ be given, i.e., ε satisfies $\varepsilon_1 \preceq \cdots \preceq \varepsilon_r$. Then we can define an associated suitable counting vector $u \in \mathbb{Z}_{\geq}^K$ by setting $u_i := |\{j \in \{1, \ldots, r\} : \varepsilon_j = \eta_i\}|$ for all $i = 1, \ldots, K$, where, by definition, we have that $\sum_{i=1}^K u_i = r$. We denote by $U := \{u \in \mathbb{Z}_{\geq}^K : \sum_{i=1}^K u_i = r\}$ the set of all *suitable counting vectors*.

Lemma 11. *There is a one-to-one correspondence between suitable right-hand-side vectors $\varepsilon \in \Upsilon$ and suitable counting vectors $u \in U$.*

Proof. First consider the case that a suitable constraint vector $\varepsilon \in \Upsilon$ is given. Then an associated suitable counting vector $u \in U$ can be determined from ε as described above, i.e., by setting $u_i := |\{j \in \{1, \ldots, r\} : \varepsilon_j = \eta_i\}|$ for all $i = 1, \ldots, K$. Conversely, if a suitable counting vector $u \in U$ is given, then we can determine associated suitable values for $\varepsilon \in \Upsilon$ by setting $\varepsilon_j := \eta_i$, where the ordinal level $i \in \{1, \ldots, K\}$ is chosen such that $\sum_{l=1}^i u_l \leq j$ and $\sum_{l=1}^{i-1} u_l > j$, for all $j = 1, \ldots, r$. $\qquad\square$

Lemma 11 implies that problem (5) can be equivalently written as

$$
\begin{aligned}
\min \quad & w(B) \\
\text{s.t.} \quad & c_i(B) = u_i, \quad i = 1, \ldots, K \\
& B \in \mathcal{X}_1,
\end{aligned}
\tag{6}
$$

where the right-hand side vector $u \in U$ is chosen as a suitable counting vector, i.e., $u \in \mathbb{Z}_{\geq}^K$ and $\sum_{i=1}^K u_i = r$, and c is the counting objective introduced in Section 3.1. Moreover, since $\sum_{i=1}^K c_i(B) = r = \sum_{i=1}^K u_i$ for all feasible bases $B \in \mathcal{X}_1$, the equality constraints in (6) can be replaced by inequality constraints without changing the feasible set. Problem (6) is thus equivalent to the following variant of ε-constraint scalarization that relates to problem (MMPCmin)

$$
\begin{aligned}
\min \quad & w(B) \\
\text{s.t.} \quad & c_i^{\min}(B) \leq u_{K-i+1}, \quad i = 1, \ldots, K \\
& B \in \mathcal{X}_1.
\end{aligned}
\tag{7}
$$

16

Corollary 12. *The non-dominated set of problem* (MMPO) *can be determined by solving problem* (6) *(or problem* (7)*) for all suitable counting vectors* $u \in U$, *and filtering out all dominated outcome vectors. The non-dominated sets of problems* (MMPCmin) *and* (MMPCmax) *can be obtained from this set by further filtering out all lexicographically dominated outcome vectors.*

Proof. The result follows immediately from Theorems 7 and 10, using the equivalence of the formulations (5), (6) and (7). □

We emphasize that problem (7) remains meaningful when a non-suitable counting vector $u \in \mathbb{Z}_{\geq}^K$ with $\sum_{i=1}^K u_i > r$ is used as right-hand-side vector. Indeed, when considering a suitable counting vector $u \in U$ and a non-suitable counting vector $\hat{u} \geqslant u$, then $\hat{u}$ yields a relaxation of problem (7) with u as the right-hand-side vector. Nevertheless, the constraint $B \in \mathcal{X}_1$ guarantees that only bases of $\mathcal{M}_1$ are returned, and hence $\sum_{i=1}^K c_i^{\min}(B) = r$ remains satisfied also in this case. Moreover, using $\bar{u} \coloneqq (r, \ldots, r) \in \mathbb{Z}_{\geq}^K$ yields a "complete" relaxation in the sense that constraints $c_i^{\min}(B) \leq \bar{u}_{K-i+1} = r$, $i = 1, \ldots, K$, are satisfied for *all* bases $B \in \mathcal{X}_1$ and hence redundant in this case.

Using the above results, the cardinality of the non-dominated set of problem (MMPO) (and hence also of problems (MMPCmin) and (MMPCmax)) can be bounded. Note that this is in analogy to the results obtained in Schäfer et al. (2020) and Schäfer et al. (2021) for ordinal shortest path and ordinal knapsack problems, respectively. Indeed, the number of equality-constraint scalarizations (6) that need to be solved in order to guarantee that all non-dominated outcome vectors of problem (MMPO) are found is polynomially bounded. This can be seen from the fact that the number of suitable counting vectors $u \in U$, i.e., the number of K-dimensional non-negative integer vectors that satisfy $\sum_{j=1}^K u_j = r$, is given by $\binom{r+K-1}{K-1} = \mathcal{O}(r^{K-1})$ (assuming that K is constant), i.e., it is equal to the number of multisets of cardinality $K - 1$ taken from a set of size $r + 1$. This number is also known as occupancy number, see, e.g., Feller (1968). We obtain the following result.

Theorem 13. *The cardinality of the non-dominated set of problem* (MMPO) *is bounded by* $\mathcal{O}(r^{K-1})$, *which is polynomial in* r *as long as* K *is constant.*

5.2. Matroid Intersection

We focus on the ε-constraint scalarization-variant (7) in the following and show that it can be equivalently formulated as a matroid intersection problem. Towards this end, let an arbitrary but fixed, suitable counting vector $u \in U$ be given as right-hand-side vector in problem (7).

Now consider the partition $E = E_1 \cup E_2 \cup \ldots \cup E_K$ of the ground set E of $\mathcal{M}_1$, where $E_j \coloneqq \{e \in E \colon o(e) = \eta_j\}$ for $j = 1, \ldots, K$, i.e., E_j contains all elements from E that are in category η_j. Given this partition of E, let $\mathcal{M}_2(u) = (E, \mathcal{I}_2(u))$ be an associated partition matroid with independent sets given by $\mathcal{I}_2(u) \coloneqq \{J \subseteq E \colon |J \cap E_j| \leq u_j, \ 1 \leq j \leq K\}$. Then problem (7) can

be solved using the matroid intersection problem

$$
\begin{aligned}
\min \quad & w(I) \\
\mathrm{s.\,t.} \quad & I \in \mathcal{I}_1 \cap \mathcal{I}_2(u) \\
& |I| = \max\{|J| : J \in \mathcal{I}_1 \cap \mathcal{I}_2(u)\}.
\end{aligned}
\tag{8}
$$

Note that the second constraint in (8) is needed since otherwise, $B = \emptyset$ would always be optimal.

Theorem 14. *Let $u \in \mathbb{Z}_{\geq}^K$ be arbitrary but fixed. If problem (7) is feasible, then problems (7) and (8) are equivalent. Moreover, if problem (7) is infeasible, then every optimal solution B^* of problem (8) satisfies $|B^*| < r$.*

Proof. We first show that when problem (7) is feasible for the given suitable counting vector u, then problems (7) and (8) have the same feasible sets. Indeed, in this case there exists a basis $\hat{B} \in \mathcal{X}_1$ that satisfies $c(\hat{B}) \leqq u$, and hence $\hat{I} := \hat{B}$ with $|\hat{I}| = r$ is feasible for (8). This implies that all feasible solutions of (8) have cardinality r and are thus bases of $\mathcal{M}_1$. In this situation, the constraints $c_i^{\min}(B) \leq u_{K-i+1}$, $i = 1, \ldots, K$ (for (7)) and $I \in \mathcal{I}_2(u)$ (for (8)) are equivalent. Since both problems also have the same objective function, they are clearly equivalent in this case. If, however, problem (7) is infeasible for the current choice of u, then the matroid intersection problem (8) is still feasible, but returns an optimal solution B^* with $|B^*| < r$. This situation can be easily recognized. $\qquad\square$

The advantage of this reformulation is that the matroid intersection problem (8) can be solved by the polynomial time *matroid intersection algorithm* (MI) of Edmonds (1971). We refer to Schrijver (2017) for a proof of its correctness and of its polynomial run time. Note that, while the matroid intersection algorithm is originally formulated for maximization problems, it can also be applied to problem (8) by multiplying all weights $w(e)$ by -1, $e \in E$, and hence maximizing $-w(e)$ rather than minimizing $w(e)$ in (8). In our implementation we use a variant of the Floyd-Warshall algorithm (see, e.g., Ahuja et al. (1993)) to realize the required shortest path computations. Note that, alternatively, the weights could be further transformed such that non-negative weight coefficients are obtained. Then, the algorithm of Dijkstra could be applied for the relevant distance computations, see Frank (1981) and Brezovec et al. (1986) for more details.

Note that the cardinality constraint in the general formulation (8) of the matroid intersection problem can be omitted if $\mathcal{I}_1$ is replaced by $\mathcal{X}_1$, i.e., the set of all bases of the matroid $\mathcal{M}_1$. Hence, we can alternatively solve the problem

$$
\begin{aligned}
\min \quad & w(B) \\
\mathrm{s.\,t.} \quad & B \in \mathcal{X}_1 \cap \mathcal{I}_2(u).
\end{aligned}
\tag{9}
$$

This is realized in Algorithm 1 by only considering optimal solutions of problem (8) that are actually bases of $\mathcal{M}_1$, see lines 4 and 5 in Algorithm 1 below.

5.3. Algorithmic Consequences

Theorems 10 and 13 imply that all solutions of (MMPO) can be determined by a polynomial number of matroid intersections applied on (8). The structure of this procedure is given in Algorithm 1. Note that this algorithm can be easily adapted to solve (MMPCmin) or (MMPCmax). Since the non-dominated set of (MMPO) is a superset of the corresponding non-dominated sets of (MMPCmin) and (MMPCmax), only a slight modification of the filtering step in line 6 is necessary.

Algorithmus 1: Matroid Intersection for Ordinal Constraints $(MIOC(\mathcal{M}_1, w, o))$

Input: Matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$, sum objective function w and ordinal objective function o
Output: Non-dominated set of problem (MMPO)

1   $X := \emptyset$
2   **foreach** $u \in U$ **do**
3   Solve (8) with (MI) and save the obtained independent set I^*
4   **if** $|I^*| = r$ **then**
5   Set $X = X \cup \{I^*\}$
6   Filter the efficient independent sets of X w.r.t. (MMPO) and save the corresponding outcome vectors in Y_{nd}^O
7   **return** Y_{nd}^O

It is possible to improve the performance of Algorithm 1 by reducing the number of considered bounds $u \in U$, i.e., the number of solved matroid intersections. This can be achieved by initially solving (8) with $u = (r, ..., r) \in \mathbb{R}^K$, which returns a weakly efficient basis B^* (assuming $\mathcal{X}_1 \neq \emptyset$) with the smallest possible cost w^*. Consequently, only upper bounds $u \in U$ such that $(u_K, \ldots, u_1) \leq_{\mathrm{lex}} c^{\min}(B^*)$ have to be considered. Thus, we modify lines 1–2 in Algorithm 1 accordingly and obtain Algorithm 2.

Note that, in the worst case, this initialization yields no reduction of the running time, since there might exists a basis B' that minimizes w and for which $c^{\min}(B') = (r, 0, \ldots, 0)$. However, in our numerical tests this procedure often leads to a significant reduction of the number of iterations, as described in Section 6. Note that the initial bound in Algorithm 2 is not a suitable counting vector as defined in Section 5.1, i.e., $u = (r, \ldots, r) \notin U$. Since we consider in the following often relaxations of suitable subproblems, we use the notation $\bar{U} := \{u \in \mathbb{Z}_{\geq}^K : \sum_{i=1}^K u_i \geq r, \ u_i \leq r, \ i = 1, \ldots, K\}$ to denote the considered upper bound set.

Based on the fact that the lexicographic order is a total order, Algorithm 2 can be further improved when applied on problem (MMPCmin). In this case, the considered upper bound set can also be reduced during the course of the algorithm. The initialization of the bound set U is analogous to Algorithm 2, i.e., we solve the matroid intersection problem for $u = (r, ..., r) \in \mathbb{R}^K$. Let

---

Algorithmus 2: Improved Initialization of Matroid Intersection for Ordinal Constraints ($MIOC^O(\mathcal{M}_1, w, o)$)

Input: Matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$, sum objective function w and ordinal objective function o

Output: Non-dominated set of problem (MMPO)

1   $u = (r, ..., r)$
2   Solve (8) with (MI) and save the obtained basis B^*
3   Set $X = \{B^*\}$
4   **foreach** $u \in \{v \in U : (v_K, \ldots, v_1) \leq_{\text{lex}} c^{\min}(B^*)\}$ **do**
5   *run lines 3–5 of Algorithm 1*
6   Filter the efficient independent sets of X w.r.t. (MMPO) and save the corresponding outcome vectors in Y_{nd}^O
7   **return** Y_{nd}^O

B^* be the obtained weakly efficient basis of (MMPCmin) minimizing the sum objective function w. Then, it is sufficient to solve subproblems with upper bounds $u \in \bar{U}$ such that $(u_K, \ldots, u_1) \leq_{\text{lex}} c^{\min}(B^*)$. Due to the lexicographic order we can explicitly enumerate the new upper bounds u to be considered as $(u_K, \ldots, u_1) \in \{(c_1^{\min}(B^*) - 1, r, \ldots, r), (c_1^{\min}(B^*), c_2^{\min}(B^*) - 1, r, \ldots, r), \ldots, (c_1^{\min}(B^*), \ldots, c_{K-2}^{\min}(B^*), c_{K-1}^{\min}(B^*) - 1, r)\}$ such that $u \geqslant 0$. These upper bounds are added to the list $\bar{U}$ of open subproblems and sorted in lexicographically increasing order. Whenever a new candidate for an efficient basis is found, we update the list of open subproblems $\bar{U}$ and re-sort it. In Algorithm 3 this procedure is repeated until $\bar{U} = \emptyset$. To simplify the notation we slightly abuse the notation and consider $\bar{U}$ to be a sorted list, referring by $\bar{U}[1]$ to the first element of this list. Note that an analogous solution algorithm can be formulated for the corresponding lexicographic maximization problem (MMPCmax).

5.4. Problems with Several Ordinal Objective Functions

Problem (MMPO) can be generalized by considering $p \geq 2$ objective functions with ordinal weights. This can be illustrated at a graph whose edges are classified w.r.t. two types of categories, for example, colors (e.g., green, orange, red) and letters (e.g., A, B). Then every edge is in exactly one of the following categories: green-A, green-B, orange-A, orange-B, red-A or red-B. These combinations of categories are a-priori not completely ordered, since, in general, neither green-B is preferred over red-A, nor red-A is preferred over green-B. However, such problems can be considered in the context of combined orderings as introduced in Section 2.2.

Multi-objective matroid problems with one sum objective function and several ordinal objective functions can be handled analogously to multi-objective matroid problems with one ordinal objective function (MMPO). Without going much into detail, we shortly describe the formulation of an associated weighted matroid intersection problem that generalizes problem (8).

Algorithmus 3: Matroid Intersection for Ordinal Constraints for (MMPCmin) ($MIOC^{C_{\min}}(\mathcal{M}_1, w, o)$)

Input: Matroid $\mathcal{M}_1 = (E, \mathcal{I}_1)$, sum objective function w and an ordinal objective function o

Output: Non-dominated set of problem (MMPCmin)

1 $\bar{U} := \{u \in \mathbb{Z}_{\geq}^K : \sum_{i=1}^K u_i \geq r\}$, $u = (r, ..., r) \in \mathbb{Z}_{\geq}^K$

2 Solve (8) with (MI) and save the obtained basis B^*

3 Set $X = \{B^*\}$

4 $\bar{U} := \{u \in \bar{U} : (u_K, \ldots, u_1) \leq_{\text{lex}} c^{\min}(B^*)\}$, sort U in lexicographically increasing order

5 while $\bar{U} \neq \emptyset$

6 $\quad$ $u := \bar{U}[1]$ $\qquad\qquad$ // pop lexicographically smallest bound

7 $\quad$ $\bar{U} := \bar{U}[2, \ldots, \text{end}]$

8 $\quad$ Solve (8) with (MI) and save the obtained independent set I^*

9 $\quad$ **if** $|I^*| = r$ **then**

10 $\quad\quad$ Set $X = X \cup \{I^*\}$

11 $\quad\quad$ $\bar{U} := \{u \in \bar{U} : (u_K, \ldots, u_1) \leq_{\text{lex}} c^{\min}(I^*)\}$, sort U in lexicographically increasing order

12 Filter the efficient independent sets of X with respect to the problem (MMPCmin) and save the corresponding outcome vectors in $Y_{\text{nd}}^{C_{\min}}$

13 return $Y_{\text{nd}}^{C_{\min}}$

Let p denote the number of ordinal objective functions o^i, $i = 1, \ldots, p$, let K_i denote the number of categories for the i-th ordinal objective, and let $\eta_{ij} \in \mathcal{C}$ denote the j-th category of the i-th ordinal objective, $j = 1, \ldots, K_i$, where $\eta_{ij} \prec \eta_{ik}$ whenever $j < k$. Then we can define a partition matroid $\mathcal{M}_3$ (generalizing $\mathcal{M}_2$) by partitioning the ground set $E = \bigcup_{i=1}^p \bigcup_{j=1}^{K_i} E_{ij}$, where $E_{ij} := \{e \in E : o^i(e) = \eta_{ij}\}$ for $i = 1, \ldots, p$ and $j = 1, \ldots, K_i$. The set of independent sets of $\mathcal{M}_3$ is given by $\mathcal{I}_3 := \{J \subseteq E : |J \cap E_{ij}| \leq u_{ij}, 1 \leq i \leq p$ and $j \in \{1, \ldots, K_i\}\}$, where u_{ij} denotes the number of elements that are allowed in category η_{ij} in the ordinal objective o^i. Note that again $0 \leq u_{ij} \leq r$ for all $i = 1, \ldots, p$ and $j \in \{1, \ldots, K_i\}$, and that $\sum_{j \in \{1, \ldots, K_i\}} u_{ij} = r$ for all $i = 1, \ldots, p$. Therefore, it is possible to solve this problem by solving all relevant weighted matroid intersection problems (8) (with $\mathcal{M}_2$ replaced by $\mathcal{M}_3$) and filtering out all dominated outcome vectors w.r.t. the combined ordering relation. In this case, the number of calls of problems (8) is bounded by $\mathcal{O}(p \cdot r^{\tilde{K}-1})$, where r still denotes the rank of the matroid $\mathcal{M}_1$ and $\tilde{K} = \max\{K_i : i = 1, \ldots, p\}$. If p and K_i are fixed, $i = 1, \ldots, p$, then the number of scalarized subproblems is polynomially bounded in the input size. Moreover, in this case every weighted matroid intersection problem can be solved in polynomial time, if we assume that an oracle can determine in polynomial time if a given subset $I \subseteq E$ is independent or not.

6. Numerical Results

The Exhaustive Swap Algorithm suggested in Gorski et al. (2021) as well as the three versions of the Matroid Intersection Algorithm for Ordinal Constraints (Algorithms 1, 2 and 3 with the algorithm of Floyd-Warshall) were implemented and numerically tested. As test instances, we consider graphic matroids and partition matroids for $\mathcal{M}_1$, where in the latter case the groundset is partitioned into three subsets. All computations were done on a computer with an Intel(R) Core(TM) i7-7500U CPU 2.70 GHz processor and 8 GB RAM. The algorithms were implemented and run in MATLAB, Version R2019b.

In the first experiment we compare the two types of algorithms on a graphic matroid with one sum objective function and one binary objective function. The instances were generated based on random connected undirected graphs $G = (V, E)$ with n nodes and m edges using the implementation of Schnepper et al. (2021). The weight coefficients w of the sum objective are randomly chosen integer values in $\{1, \ldots, 2m\}$, and the values of the binary objective b are random binary values. In both cases we used a uniform distribution. We solved the obtained instances of problem (BBMP) by the Exhaustive Swap Algorithm and by Algorithm 2.

The numerical results can be found in Table 1. In the first two columns the instance size is given by the number of nodes and edges (n, m) and the average number of non-dominated outcome vectors $|Y_{\mathrm{nd}}|$ over 100 random instances. The results show clearly that the average running of Algorithm 2 increases much faster with the instance size compared to the Exhaustive Swap Algorithm.

This result is not surprising, because the exhaustive swap algorithm utilizes the specific problem structure, in particular the connectedness of the non-dominated set, as proven in Gorski et al. (2021). Nevertheless, the number of problems that are solved with Algorithm 2 (*iter*) is quite close to the number of non-dominated outcome vectors (Y_{nd}), which indicates that only few redundant problems were solved.

The strength of all three matroid intersection algorithms for ordinal constraints is that they can be applied to a broader class of problems than the exhaustive swap algorithm, which is restricted to two ordinal categories. In the following tests we used again randomly generated graphs $G = (V, E)$ with n nodes and m edges with objective function coefficients w and o. The entries of w and o were generated randomly with uniform distribution in $\{1, \ldots, 2m\}$ and in $\{1, \ldots, K\}$ for w and o, respectively, were $K \in \{3, 4, 5\}$. The results for $K = 3, 4, 5$ can be found in Tables 2, 3 and 4, respectively. We observe that the number of solutions found for the different problems is quite similar for small problem sizes, but for larger instances and more categories the number of non-dominated points is much smaller for the lexicographic models as compared to the ordinal approach. The running time depends obviously on the instance size. However, the effect of an increasing number of edges m is rather limited. A significant influence can be seen by the number of nodes n, which determines the rank of the matroid. Furthermore, the number of categories K has an important effect on the running time.

Instanze Size		$MIOC^O$		ESA
(n, m)	$|Y_{\mathrm{nd}}|$	$iter$	$[s]$	$[s]$
$(7, 10)$	2.39	3.26	0.08	0.03
$(7, 15)$	3.32	3.84	0.16	0.03
$(7, 20)$	3.87	3.97	0.24	0.03
$(10, 20)$	4.36	5.13	0.54	0.04
$(10, 30)$	5.17	5.44	0.97	0.05
$(10, 40)$	5.46	5.53	1.38	0.05
$(15, 30)$	6.05	6.94	2.43	0.05
$(15, 60)$	7.77	7.95	6.93	0.07
$(15, 100)$	7.90	7.99	12.57	0.07
$(20, 40)$	7.58	8.58	7.36	0.07
$(20, 100)$	10.49	10.60	30.26	0.10
$(20, 180)$	10.35	10.39	59.36	0.11

Table 1: Average computation time in seconds to solve 100 instances of problem (BBMP) on a graphic matroid with the Exhaustive Swap Algorithm (ESA) and with Algorithm 2 ($MIOC^O$).

As expected, reducing the number of considered upper bound vectors u for problem (MMPO) generally leads to fewer iterations. On average, only little more than half of the iterations are needed in this case. Nevertheless, note that in worst case this strategy may not lead to an improvement. In the case of the lexicographic variant (MMPCmin) the potential reduction is much more significant. Indeed, the required computation time is drastically reduced in this case, especially for large K. For example, for $K = 5$ we have a reduction of the running time by a factor of around 20 in all cases with $n = 10$.

We get similar results when testing with partition matroids rather than graphic matroids. Here, we consider a ground set of n objects and restrict the analysis to partitions of the ground set into three subsets. The upper bounds on the number of elements from each subset are selected such that every basis consists of $\frac{n}{2}$ elements, and the problem is feasible. After defining an instance of a partition matroid $\mathcal{M}_1$ in this way, the objective functions are generated. Each object has an associated weight between 1 and $10 \cdot n$ and is assigned to one of K categories, where $K \in \{3, 4, 5\}$. The results for $K = 3, 4, 5$ can be found in Tables 5, 6 and 7, respectively. Again, the improved choice of u leads to significantly better running times. Moreover, the running time increases with the number of elements n and the number of categories K.

Note that further speed-ups can be expected by parallel implementations of the matroid intersection algorithms.

7. Conclusion

In this paper we consider single- and multi-objective matroid optimization problems that combine "classical" sum objective functions with one or several ordinal objective functions. Besides the concept of ordinal optimality, we consider two variants of lexicographic optimization that lexicographically maximize

	Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
(n,m)	$|Y_{\mathrm{nd}}^O|$	$|Y_{\mathrm{nd}}^{C_{\min}}|$	$|Y_{\mathrm{nd}}^{C_{\max}}|$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
$(7,10)$	3.90	3.65	3.75	28	0.48	17.10	0.31	6.30	0.16
$(7,15)$	7.30	6.05	6.40	28	1.02	18.55	0.77	9.75	0.45
$(7,20)$	6.20	5.40	5.65	28	1.67	14.45	0.87	8.00	0.56
$(10,20)$	9.10	7.65	7.85	55	5.03	32.50	2.87	12.55	1.47
$(10,30)$	12.55	9.60	11.35	55	9.63	32.80	5.67	13.60	2.61
$(10,40)$	15.90	11.75	12.95	55	13.56	31.90	8.00	15.25	3.94
$(15,30)$	15.20	11.60	11.40	120	36.16	70.40	21.00	18.60	6.43
$(15,60)$	27.20	18.40	20.65	120	101.30	69.00	58.97	24.30	21.07
$(15,100)$	27.90	18.20	22.65	120	184.26	66.95	105.23	23.10	36.19
$(20,40)$	20.95	14.65	15.40	210	158.48	113.85	85.85	23.95	20.53
$(20,100)$	38.45	24.55	27.00	210	578.96	101.00	282.43	30.95	87.42
$(20,180)$	46.55	27.20	33.45	210	1 162.27	115.35	648.08	33.70	190.49

Table 2: Numerical results for a graphic matroid and $K = 3$ categories. For every problem size 20 instances were solved to obtain average results.

	Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
(n,m)	$|Y_{\mathrm{nd}}^O|$	$|Y_{\mathrm{nd}}^{C_{\min}}|$	$|Y_{\mathrm{nd}}^{C_{\max}}|$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
$(7,10)$	4.15	3.70	3.70	84	1.34	44.45	0.73	9.00	0.25
$(7,15)$	7.50	5.85	6.25	84	2.83	43.85	1.61	12.10	0.56
$(7,20)$	10.50	7.85	8.10	84	4.44	52.05	2.88	14.80	1.93
$(10,20)$	12.15	8.65	8.60	220	18.15	120.25	9.38	18.75	2.01
$(10,30)$	20.60	12.40	14.80	220	36.78	132.95	22.01	23.85	4.46
$(10,40)$	23.70	14.35	17.35	220	53.79	118.00	28.27	23.90	5.95
$(15,30)$	29.05	16.80	17.55	680	191.27	395.75	107.14	37.15	12.84
$(15,60)$	56.00	24.95	34.10	680	560.72	350.15	291.72	42.40	35.97
$(15,100)$	63.60	28.00	38.70	680	1 042.33	377.45	597.78	44.15	69.33
$(20,40)$	44.80	21.80	23.50	1 540	1 073.17	708.20	502.36	43.80	37.32

Table 3: Numerical results for a graphic matroid and $K = 4$ categories. For every problem size 20 instances were solved to obtain average results.

	Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
(n,m)	$|Y_{\mathrm{nd}}^O|$	$|Y_{\mathrm{nd}}^{C_{\min}}|$	$|Y_{\mathrm{nd}}^{C_{\max}}|$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
$(7,10)$	4.20	3.85	3.85	210	3.10	106.75	1.57	10.95	0.27
$(7,15)$	10.75	7.55	8.10	210	6.65	121.75	3.88	18.05	0.81
$(7,20)$	14.90	10.00	11.25	210	10.56	124.30	6.14	20.35	1.36
$(10,20)$	22.85	11.75	12.65	715	57.29	429.80	33.25	30.30	3.19
$(10,30)$	24.70	14.55	15.95	715	114.37	396.00	62.16	32.00	5.57
$(10,40)$	30.10	15.65	19.80	715	175.93	410.75	98.27	32.80	8.22

Table 4: Numerical results for a graphic matroid and $K = 5$ categories. For every problem size 20 instances were solved to obtain average results.

Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
n	$\lvert Y^O_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\min}}_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\max}}_{\mathrm{nd}}\rvert$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
10	2.55	2.50	2.50	21	0.23	11.20	0.10	4.60	0.08
20	4.75	4.40	4.45	66	3.26	35.45	1.83	8.40	0.63
30	9.90	8.00	8.40	136	20.26	76.50	11.43	13.45	2.68
40	16.85	12.55	12.75	231	81.26	127.25	45.14	20.40	8.92
50	24.05	17.60	15.90	351	229.91	201.90	131.29	26.30	21.43
60	34.00	21.10	21.75	496	552.54	284.20	318.62	32.80	46.41
70	39.70	24.50	25.40	666	1 177.52	375.90	662.34	37.30	83.06

Table 5: Numerical results for a partition matroid with three subsets and $K = 3$ categories. For every problem size 20 instances were solved to obtain average results.

Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
n	$\lvert Y^O_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\min}}_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\max}}_{\mathrm{nd}}\rvert$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
10	2.55	2.45	2.40	56	0.41	31.10	0.23	6.15	0.10
20	7.10	5.80	5.85	286	12.60	173.90	7.43	14.05	0.94
30	15.15	9.75	10.05	816	117.69	451.70	62.22	22.85	4.33
40	40.40	20.95	21.25	1 771	594.17	967.80	317.36	42.25	18.51
50	44.90	22.50	22.20	3 276	2 081.56	1 946.25	1 191.88	49.70	41.17

Table 6: Numerical results for a partition matroid with three subsets and $K = 4$ categories. For every problem size 20 instances were solved to obtain average results.

Instance Size			$MIOC$		$MIOC^O$		$MIOC^{C_{\min}}$	
n	$\lvert Y^O_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\min}}_{\mathrm{nd}}\rvert$	$\lvert Y^{C_{\max}}_{\mathrm{nd}}\rvert$	$iter$	$[s]$	$iter$	$[s]$	$iter$	$[s]$
10	3.65	3.15	3.25	126	0.84	71.55	0.47	8.20	0.14
20	10.65	8.20	7.75	1 001	43.34	539.15	22.62	23.20	1.67
30	28.25	15.35	16.35	3 876	534.59	2 293.50	308.62	41.50	7.83

Table 7: Numerical results for a partition matroid with three subsets and $K = 5$ categories. For every problem size 20 instances were solved to obtain average results.

the number of "good" elements or minimize the number of "bad" elements, respectively. In the case of (single-objective) ordinal optimization, we show that these concepts are actually equivalent for matroids, and that optimal solutions can be found by a simple and efficient greedy strategy. In the multi-objective setting, we use variants of ε-constraint scalarizations to obtain a polynomial number of matroid intersection problems, from which the non-dominated sets of the respective problems can be derived by simple filtering operations. This yields an overall polynomial-time algorithm for multi-objective ordinal matroid optimization problems. Numerical tests on graphic matroids and on partition matroids validate the efficiency of this approach.

Future research should focus on a further analysis of the similarities and differences between multi-objective optimization problems with classical sum objectives and with ordinal objectives. Moreover, alternative (partial) orderings may be considered and analyzed in the light of different scalarization techniques.

Acknowledgment

References

Ahuja RK, Magnanti TL, Orlin JB. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.

Arroyo JEC, Vieira PS, Vianna DS. A GRASP algorithm for the multi-criteria minimum spanning tree problem. Annals of Operations Research 2008;159(1):125–33. doi:10.1007/s10479-007-0263-4.

Brezovec C, Cornuéjols G, Glover F. Two algorithms for weighted matroid intersection. Mathematical Programming 1986;36(1):39–53. doi:10.1007/BF02591988.

Chen G, Chen S, Guo W, Chen H. The multi-criteria minimum spanning tree problem based genetic algorithm. Information Sciences 2007;177(22):5050–63. doi:10.1016/j.ins.2007.06.005.

Davis-Moradkhan M, Browne WN. A hybridised evolutionary algorithm for multi-criterion minimum spanning tree problems. In: 2008 Eighth International Conference on Hybrid Intelligent Systems. IEEE; 2008. p. 290–5. doi:10.1109/HIS.2008.86.

Edmonds J. Matroids and the greedy algorithm. Mathematical Programming 1971;1(1):127–36. doi:10.1007/bf01584082.

Edmonds J. Submodular functions, matroids, and certain polyhedra. In: Jünger M, Reinelt G, Rinaldi G, editors. Combinatorial Optimization — Eureka, You Shrink!: Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers. Berlin, Heidelberg: Springer Berlin Heidelberg; 2003. p. 11–26. doi:10.1007/3-540-36478-1_2.

Ehrgott M. On matroids with multiple objectives. Optimization 1996;38(1):73–84. doi:10.1080/02331939608844238.

Ehrgott M. Multicriteria Optimization. Springer, 2005. doi:10.1007/3-540-27659-9.

Ehrgott M, Gandibleux X. A survey and annoted bibliography of multiobjective combinatorial optimization. OR Spektrum 2000;22(4):425–60. doi:10.1007/s002910000046.

Ehrgott M, Klamroth K. Connectedness of efficient solutions in multiple criteria combinatorial optimization. European Journal of Operational Research 1997;97(1):159–66. doi:10.1016/S0377-2217(96)00116-6.

Feller W. An Introduction to Probability Theory and its Applications. volume I. Wiley, 1968.

Frank A. A weighted matroid intersection algorithm. Journal of Algorithms 1981;2(4):328–36. doi:10.1016/0196-6774(81)90032-8.

Gabow HN, Tarjan RE. Efficient algorithms for a family of matroid intersection problems. Journal of Algorithms 1984;5:80–131. doi:10.1016/0196-6774(84)90042-7.

Gorski J. Multiple Objective Optimization and Implications for Single Objective Optimization. Aachen, Germany: Shaker Verlag, 2010.

Gorski J, Klamroth K, Sudhoff J. Biobjective optimization problems on matroids with binary costs. 2021. arXiv:2108.00709.

Hamacher HW, Ruhe G. On spanning tree problems with multiple objectives. Annals of Operations Research 1994;52(4):209–30. doi:10.1007/BF02032304.

Oxley J. Matroid Theory. Oxford Univ. Press, 2011.

Schäfer LE, Dietz T, Barbati M, Figueira JR, Greco S, Ruzika S. The binary knapsack problem with qualitative levels. European Journal of Operational Research 2021;289(2):508–14. doi:10.1016/j.ejor.2020.07.040.

Schäfer LE, Dietz T, Fröhlich N, Ruzika S, Figueira JR. Shortest paths with ordinal weights. European Journal of Operational Research 2020;280(3):1160–70. doi:10.1016/j.ejor.2019.08.008.

Schnepper T, Klamroth K, Puerto J, Stiglmayr M. A local analysis to determine all optimal solutions of p-k-max location problems on networks. Discrete Applied Mathematics 2021;296:217–34. doi:`10.1016/j.dam.2020.05.013`.

Schrijver A. Combinatorial Optimization. Springer Berlin Heidelberg, 2002.

Schrijver A. A course in combinatorial optimization. 2017. URL: `https://homepages.cwi.nl/~lex/files/dict.pdf`.