Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 15/37

José Rui Figueira, Carlos Fonseca, Pascal Halffmann, Kathrin Klamroth, Luís Paquete, Stefan Ruzika, Britta Schulze, Michael Stiglmayr, David Willems

Easy to say they're hard, but hard to see they're easy

—

Toward a categorization of tractable multiobjective combinatorial optimization problems

November 2015

http://www.math.uni-wuppertal.de

Easy to say they're hard, but hard to see they're easy

Toward a categorization of tractable multiobjective combinatorial optimization problems

José Rui Figueira(1), Carlos Fonseca(2), Pascal Halffmann(3), Kathrin Klamroth(4), Luís Paquete(2), Stefan Ruzika(3), Britta Schulze(4), Michael Stiglmayr*(4), and David Willems(3)

(1)CEG-IST Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

(2)CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

(3)Mathematical Institute, University of Koblenz-Landau, Germany

(4)School of Mathematics and Natural Sciences, University of Wuppertal, Germany

Abstract: Multiobjective combinatorial optimization problems are known to be hard problems for two reasons: their decision versions are often NP-complete and they are often intractable. Apart from this general observation, are there also variants or cases of multiobjective combinatorial optimization problems which are easy and, if so, what causes them to be easy? This article is a first attempt to provide an answer to these two questions. Thereby, a systematic description of reasons for easiness is envisaged rather than a mere collection of special cases. In particular, the borderline of easy and hard multiobjective optimization problems is explored.

Keywords: multiobjective combinatorial optimization, tractability, complexity theory, bottleneck objective, binary coefficients, dynamic programming

*Corresponding author, Email: stiglmayr@math.uni-wuppertal.de, FAX: +49 202 439-3061 School of Mathematics and Natural Sciences, University of Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany

1 Introduction

It is a well known fact that multiobjective combinatorial optimization (MOCO) problems are not efficiently solvable in general. This is true for several reasons. First, from an asymptotic worst case perspective, the number of nondominated points may grow exponentially with the input size. This property is referred to as *intractability* in the MOCO literature (it should be pointed out that *intractability* is typically used in a different meaning in the theoretical computer science community). Second, the computation of nonsupported efficient solutions may be significantly more demanding than the solution of the single objective analogon. Linear scalarization methods capable of yielding nonsupported efficient solutions introduce new (knapsack-type) constraints to the combinatorial structure. Thus, these scalarized problems lead to resource constrained versions of combinatorial problems with the consequence that these scalarized single objective combinatorial optimization problems turn out to be NP-hard. To sum it up, a MOCO problem can only be polynomially solvable, if it has a polynomially bounded number of nondominated points and if the nonsupported efficient solutions (if existent) can be computed efficiently. These two observations motivate this article.

The article of Serafini [1987] is one of the first publications surveying multiobjective combinatorial optimization while emphasizing complexity theory. Serafini distinguishes between nine different notions of solving a multiobjective optimization problem: it can be understood as computing (1) all efficient solutions, (2) all nondominated points and one corresponding efficient solution for each, (3) all nondominated points, (4) a given number of nondominated points ((5) and one corresponding effcient solution for each), (6) one efficient solution, (7) one nondominated point, and (8/9) two decision problems. We adapt Serafini's second notion which is nowadays referred to as computing a minimal complete set of efficient solutions.

Many articles have been published on different types of MOCO problems since then. Complexity analysis of the considered MOCO problems is an integral part in many articles published (see e. g. Ulungu and Teghem [1994], Ehrgott and Gandibleux [2000], Bökler et al. [2016] and the references given therein). However, most of the complexity results for MOCO problems stress their difficulty, i. e. most of the MOCO problems are intractable and NP-hard in general. Ehrgott [2000] discusses reasons for this difficulty and concludes that it is "hard to say it's easy" in general. In this article, we revisit this question and focus on particular cases of MOCO problems, which are polynomially solvable. We aim at categorizing them, explaining their polynomial solvability in terms of general structural properties, and, finally, explore the grey zone between easy and hard MOCO problems.

The remainder of this article is organized as follows. Section 2 provides some theoretical background and notation. The following sections comprise particular cases of MOCO problems which are polynomially solvable. In Section 3, we consider multiobjective MOCO problems with binary coefficients in sum objective functions. The following section generalizes this by weakening this assumption of binary coefficients. Section 5 then surveys easy MOCO problems with a different kind of objective function, so-called bottleneck objectives. In Section 6, the borderline between easy and hard MOCO problems is studied. The article concludes with a brief summary of the main observations.

2 Definitions and Notation

In the following, let $\mathcal{E}$ be a finite set $\mathcal{E} := \{e_1, \ldots, e_n\}$, $|\mathcal{E}| = n$, and let $c^l\colon \mathcal{E} \to \mathbb{Z}$ be a cost function for each $l = 1, \ldots, q$, $q \in \mathbb{N}$, which maps the elements of $\mathcal{E}$ to integer values. A *multiobjective combinatorial optimization problem* (MOCO) is then characterized by

- a feasible set $\mathcal{X} \subseteq 2^{\mathcal{E}}$, where $2^{\mathcal{E}}$ denotes the power set of $\mathcal{E}$,
- and q (generally conflicting) objective functions $f^l\colon \mathcal{X} \to \mathbb{Z}, l = 1, \ldots, q$, which are to be minimized in general.

With this notation we can write this optimization problem in the usual form

$$\min_{x \in \mathcal{X}} f(x) = \min_{x \in \mathcal{X}} \left(f^1(x), \ldots, f^q(x)\right). \tag{MOCO}$$

Here, $x \in \mathcal{X}$ denotes a subset of $\mathcal{E}$, $x \subseteq \mathcal{E}$. For each $l \in \{1, \ldots, q\}$, the objective function value $f^l(x)$ of a feasible solution $x \in \mathcal{X}$ depends on the cost functions $c^l(e)$ with $e \in x$. Two types of objective functions are predominantely considered in multiobjective combinatorial optimization, *sum objectives* and *bottleneck objectives*, i. e.

$$f^l(x) = \sum_{e \in x} c^l(e) \qquad \text{and} \qquad f^l(x) = \max_{e \in x} c^l(e).$$

Consequently, the problem

$$\min_{x \in \mathcal{X}} f(x) = \min_{x \in \mathcal{X}} \left(\sum_{e \in x} c^1(e), \ldots, \sum_{e \in x} c^q(e)\right)$$

is called a *sum problem* and the problem

$$\min_{x \in \mathcal{X}} f(x) = \min_{x \in \mathcal{X}} \left(\max_{e \in x} c^1(e), \ldots, \max_{e \in x} c^q(e)\right)$$

is referred to as a *bottleneck problem*.

Bottleneck objective functions focus on the largest cost coefficient among all selected elements in a feasible solution. This concept can be generalized as follows (cf. Gorski and Ruzika [2009]). For an integer k such that $1 \leq k \leq \min_{x \in \mathcal{X}} |x|$, the *$k$-max objective* for $x \in \mathcal{X}$ is $k\text{-max}_{e \in x}\, c(e)$, where $k\text{-max}_{e \in x}$ refers to the kth largest cost coefficient among the elements of x. Note that the 1-max function is equivalent to the bottleneck function.

Throughout the article we use the Pareto concept of optimality (cf. Steuer [1986], Ehrgott [2005]). For $y, z \in \mathbb{R}^q$, $q \geq 2$, we define the following componentwise orderings:

$$\begin{aligned} y \leq z \;&:\Longleftrightarrow\; y^l \leq z^l, \text{ for } l = 1, \ldots, q, \text{ but } y \neq z, \\ y < z \;&:\Longleftrightarrow\; y^l < z^l, \text{ for } l = 1, \ldots, q. \end{aligned}$$

For minimization problems, an image $y = f(x^1)$ is called *dominated* by another image $z = f(x^2)$, if $z \leq y$. An image is called *nondominated*, if it is not dominated by any other image. Analogously, for $x \in \mathcal{X}$, we call x *efficient*, if $f(x)$ is nondominated. We call a feasible solution x^1 *weakly efficient* if there does not exist $x^2 \in \mathcal{X}$ such that $f(x^2) < f(x^1)$. The image $y = f(x^1)$ is then called *weakly nondominated*.

The set of all efficient solutions and weakly efficient solutions is called the *efficient set* and *weakly efficient set*, respectively. They are denoted by $\mathcal{X}_E$ and $\mathcal{X}_{wE}$. Analogously, the set of all (weakly) nondominated points is called the *(weakly) nondominated set*, and abbreviated by $\mathcal{Y}_N$ and $\mathcal{Y}_{wN}$, respectively. This notation can be easily adapted to the case of maximization problems (which will not be done here explicitly due to intended brevity).

Throughout this paper, we understand the task of multiobjective optimization as computing a minimal complete set of efficient solutions, i. e. finding all nondominated images $y \in \mathcal{Y}_N$ and, for each image y, a preimage $x \in \mathcal{X}_E$ with $f(x) = y$.

A MOCO problem is called *intractable*, if the size of the set of nondominated points can be exponential in the size of the problem instance. This means there exists no polynomial p such that the cardinality of the nondominated set $|\mathcal{Y}_N|$ is bounded by $\mathcal{O}(p(n))$, where n denotes the encoding length of the problem instance.

For some weight vector $0 \leq \lambda \in \mathbb{R}^q$, the single objective optimization problem

$$\min_{x \in \mathcal{X}} \sum_{l=1}^{q} \lambda_l f^l(x) \tag{1}$$

is called a *weighted sum scalarization* of (MOCO). This scalarization was introduced in Zadeh [1963]. It is well-known that every optimal solution of (1) is a weakly efficient solution for (MOCO). Solutions that can be obtained in this way are called *supported* (weakly) efficient solutions, all other efficient solutions are called *nonsupported* (weakly) efficient solutions.

Further, a supported (weakly) efficient solution x is an extreme supported (weakly) efficient solution, if its objective value $y = f(x)$ cannot be expressed by a convex combination of points in $\mathcal{Y}_N \backslash \{y\}$.

The *ϵ-constraint method*, which was introduced in Haimes et al. [1971], is another well-studied technique for solving multiobjective optimization problems. It can be formulated as

$$\begin{aligned} \min \quad & f^j(x) \\ \text{s.t.} \quad & f^l(x) \leq \epsilon_l \qquad l = 1, \dots, q, \quad l \neq j \\ & x \in \mathcal{X} \end{aligned} \tag{2}$$

where $\epsilon \in \mathbb{R}^q$ and $j \in \{1, \dots, q\}$. In contrast to the weighted sum method, the objectives are not aggregated in the ϵ-constraint method. Instead, one of the objectives is chosen to be minimized (or maximized) while the others are added to the problem's constraints. Similar to the weighted sum scalarization, every optimal solution x of (2) is weakly efficient for (MOCO) and every efficient solution can be obtained by solving (2) with an appropriate choice of ϵ.

Also, in particular sections we examine the connectedness of the set of efficient solutions $\mathcal{X}_E$, which we define accordingly to Gorski et al. [2011]. That is, $\mathcal{X}_E$ is connected, if the graph $G = (V, E)$ with $V = \mathcal{X}_E$ and $E = \{(u, v) \; : \; u, v \in \mathcal{X}_E \wedge u \text{ and } v \text{ are adjacent}\}$ is connected. The adjacency property is problem specific; for example, the exchange of variables in MOCO problems (for $i \neq j$ and $x_i = 1$, $x_j = 0$, set $x_i = 0$ and $x_j = 1$) is a common adjacency property. Likewise, for multiobjective linear problems two basic solutions of the simplex algorithm, where one can be generated from the other by exchanging base indices, are adjacent regarding a canonical adjacency property.

3 Sum Objectives with Binary Coefficients

In this section, we consider MOCO problems with sum objectives. Moreover, we assume that the coefficients c_i^l of all individual objective functions f^l (but one) are binary.

Theorem 3.1. *Consider a multiobjective combinatorial optimization problem with q linear sum objectives, where $q-1$ of them contain only binary coefficients, i. e. $c_i^k \in \{0, 1\}$ for all $i = 1, \ldots, n$, and $k = 2, \ldots, q$:*

$$\begin{aligned} \min \quad & \Big(\sum_{i=1}^n c_i^1 x_i, \sum_{i=1}^n c_i^2 x_i, \ldots, \sum_{i=1}^n c_i^q x_i\Big) \\ \text{s.t.} \quad & x \in \mathcal{X} \subseteq \{0,1\}^n. \end{aligned} \tag{3}$$

Then $|\mathcal{Y}_N| \leq (n+1)^{q-1}$, i. e. the cardinality of the nondominated set $\mathcal{Y}_N$ is polynomially bounded in the coding length of the input.

Before stating the proof, it should be noted that we slightly abuse the notation of feasible solutions: For convenience, we simply identify each solution $x \subseteq \mathcal{E}$ with a binary indicator variable x specifying which elements of the ground set $\mathcal{E}$ are contained in x.

Proof. We solve the problem using an iterative application of the ϵ-constraint scalarization by restricting the objectives with binary coefficients. Since objective functions with binary coefficients can only attain integer values in $\{0, \ldots, n\}$, one has to consider $n+1$ different values of the right hand side for each ϵ-constraint on the objective functions $2, \ldots, q$. Thus, $(n+1)^{q-1}$ is an upper bound on the number of nondominated points. Consequently, the number of nondominated points is polynomially bounded by the size of the problem instance. □

In the following, we make use of this result and combine it with particular properties for distinct problems in order to get easy MOCO cases. Starting with matroid problems, we present a variety of cases for which the nondominated set of multiobjective combinatorial problems can easily be computed. Nevertheless, the mere presence of binary coefficients does not guarantee a polynomial time algorithm for the whole nondominated set as will be shown later in Section 6.1.

3.1 Matroid Problems

Single objective optimization problems on matroids have been extensively studied, with a prominent example being minimum spanning tree problems [Oxley, 2004]. Single objective optimization problems on matroids are considered *easy* problems since they can be solved using greedy algorithms. More precisely, if there exists a (polynomial time) black box algorithm that can check, if a particular set is contained in the matroid structure, we can indeed apply a greedy type algorithm for this class of problems.

Definition 3.2. *A matroid $\mathcal{M}$ is an ordered pair $(\mathcal{E}, \mathcal{I})$ consisting of a finite set $\mathcal{E}$ and a collection $\mathcal{I}$ of subsets of $\mathcal{E}$ satisfying*

1. $\emptyset \in \mathcal{I}$,
2. $x^1 \subseteq x^2$, $x^2 \in \mathcal{I} \Longrightarrow x^1 \in \mathcal{I}$ *and*
3. $x^1, x^2 \in \mathcal{I}$ *and* $|x^1| < |x^2| \Longrightarrow \exists\, e \in x^2 \setminus x^1 : x^1 \cup \{e\} \in \mathcal{I}$.

If these conditions are satisfied, $x \in \mathcal{I}$ is called an independent set. *Moreover, an independent set $x \in \mathcal{I}$ is called* maximal independent set *or* basis *of $\mathcal{M}$, if $x \cup \{e\} \notin \mathcal{I}$ for all $e \in \mathcal{E} \setminus x$.*

Example 3.3. *Let $G = (V, E)$ be an undirected simple finite graph. The graphic matroid $\mathcal{M}(G)$ is then given by $\mathcal{E} = E$ and $\mathcal{I} = \{T \subseteq E : T \text{ contains no cycle}\}$. The bases of $\mathcal{M}(G)$ correspond to the spanning trees of G.*

Let $\mathcal{X} \subseteq \mathcal{I}$ denote the set of all bases of a matroid $\mathcal{M} = (\mathcal{E}, \mathcal{I})$. Then the *multiobjective matroid problem* is given by

$$\begin{aligned} \min \quad & (c^1(x), \dots, c^q(x)) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned}$$

In the special case $q = 2$, we obtain the *biobjective matroid problem.*

An intuitive conjecture is that optimization problems on matroids remain easy also in the bi- and multiobjective setting. However, it turns out that this can be shown only under quite restrictive assumptions on the problem setting while it is not the case in general.

Hard Cases More precisely, the biobjective matroid problem is in general intractable, i. e., the nondominated set may grow exponentially with the size of the instance even for $q = 2$. Moreover, the decision version of the biobjective matroid problem is NP-complete (see Ehrgott [1996, 2005]).

Easy Cases On the other hand, Gorski [2010] showed for the biobjective case, $q = 2$, that, if the coefficients in one of the objectives are binary, then the nondominated set of the biobjective matroid problem has at most m elements, where m denotes the cardinality of a basis of $\mathcal{M}$. Note that this result also follows from Theorem 3.1 above. Moreover, the nondominated set can be computed in polynomial time using a simple exchange argument adapted from an algorithm of Gabow and Tarjan [1984]. This method implies that (1) all nondominated points of the biobjective matroid problem are supported, and (2) the nondominated set of this problem is connected (for an introduction to and results concerning connectedness of efficient solutions we refer to Ehrgott and Klamroth [1997] and Gorski et al. [2011]).

Open Questions Generalizations to other types of bounded coefficients and/or to problems with more than two objective function pose several interesting research questions.

3.2 Greedy Algorithms

For some cases, the combinatorial structure of the problem is not matroid or, at least, its verification is not easy. Nonetheless, the correctness of a greedy approach can be sometimes proved directly as shown by the following variant of the multiobjective knapsack problem [Gorski et al., 2012b].

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} c_i^1 x_i \\ \min \quad & \left(\sum_{i=1}^{n} c_i^2 x_i, \ldots, \sum_{i=1}^{n} c_i^q x_i \right) \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & x \in \{0,1\}^n, \end{aligned}$$

where $\mathcal{X}$ denotes the set of all feasible solutions specified by multiple knapsack constraints. For $q = 2$, an $O(n \log n)$ time algorithm that pre-sorts the coefficients solves the problem. In [Gorski et al., 2012b], the authors propose a greedy strategy for the special case of $q = 3$ and $c_i^l \in \{0,1\}$ for $i = 1, \ldots, n$ and $l = 2, 3$ by pre-sorting the coefficients. Their greedy algorithm runs in $O(n^2)$. This running time is asymptotically optimal, since there exist $O(n^2)$ nondominated objective vectors in general. However, the greedy strategy fails for $q = 4$ [Gorski et al., 2012b].

3.3 Dynamic Programming

Dynamic programming is a well-established algorithmic technique for solving optimization problems which exhibit Bellman's *Principle of Optimality* [Bellman, 1957]: optimal solutions of the overall problem can be easily constructed by extending optimal solutions of smaller subproblems. Dynamic programming often leads to very efficient algorithms (cf. several variants of the shortest path problem) or, for NP-hard problems, it often implies pseudo-polynomial running time of solution algorithms (cf. various versions of the knapsack problem). In the following we use the knapsack problem

$$\begin{aligned} \max \quad & f^1(x) = \sum_{i=1}^{n} c_i^1 x_i \\ \text{s. t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W \\ & x_i \in \{0,1\} \quad \forall i \in \{1, ..., n\} \end{aligned}$$

as an illustrative example.

In a dynamic programming algorithm, the solution process is divided into S stages. For the knapsack problem, for example, one stage corresponds to the decision of fixing one variable. Thus, the number of stages is equal to the number of variables, $S = n$. Each stage contains at most T different states corresponding to solutions of exactly one subproblem. For the knapsack problem, there is one state for every possible value of the left hand side of the constraint, i. e., overall we obtain $T = W + 1$ states.

The states in stage i can be evaluated through recursive equations applied on the states of stage $(i-1)$ (or on all previously computed states), which retain the feasibility of each solution. As described before, in the context of the knapsack problem the recursion is based on iteratively fixing one additional variable x_i to 0 or 1, respectively, leading to new partial solutions that extend partial solutions from the predecessor states. Bellman's principle then guarantees that one optimal solution for each state in stage i can be generated by only using the optimal solutions of the states in stage $(i-1)$ (or in all predecessor states, respectively). Therefore, an overall optimal solution can be computed recursively. For the knapsack problem, the optimal solution in state t of stage i can be obtained by comparing the value of state t of stage $(i-1)$ and c_i^1 added to the value of state $t - w_i$ of stage $(i-1)$, if it exists.

The dynamic programming process takes $O(S \cdot T \cdot R)$-time, where R is the complexity of the recursion to obtain a new state.

Bellman's principle of optimality can be extended to multiobjective optimization problems with new objective functions $f^2, \ldots, f^q$. In this case, instead of one optimal solution, a state contains a set of efficient solutions. The cardinality of these sets distinguishes between easy and hard instances:

Hard Cases In general, adding one or more objectives make these problems intractable. As for matroid problems the nondominated set for each state may grow exponentially with the size of the instance. See, for example, Bazgan et al. [2009b] for the knapsack problem and Guerriero and Musmanno [2001] for the shortest path problem.

Easy Cases However, as shown in Theorem 3.1, restricting the additional $(q-1)$ objectives to binary coefficients also restricts the number of nondominated objective vectors to at most $\ell \leq (n+1)^{q-1}$. This directly transfers to every of the $S \cdot T$ states of the dynamic programming process, which, therefore, handles at most $O(S \cdot \ell \cdot T \cdot R)$ solutions.

The recursion requires to filter on efficient solutions, which can be done for every state in $O(\ell^2)$-time by pairwise comparison of the at most ℓ objective vectors in this state. Figure 1 illustrates how a state t in stage i is built using states t and $t - w_i$ of stage $(i-1)$ for an instance of the knapsack problem with two objectives, the second one with binary coefficients.

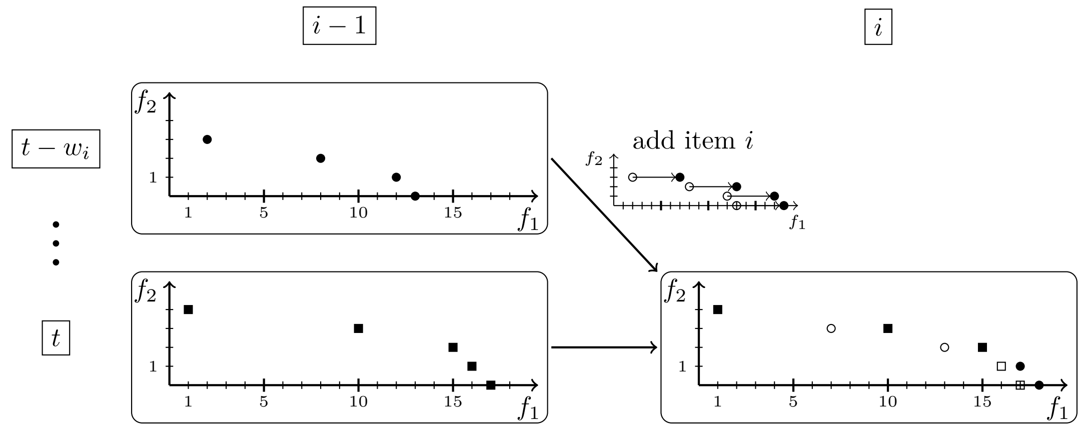


Figure 1: For an instance of the knapsack problem with two objectives f_1 and f_2, f_2 with binary coefficients, this figure illustrates the transformation from two states in stage $(i-1)$ to a new state in stage i. The symbols • and ▪ show the nondominated objective vectors of the respective states. The symbols ∘ and ▫ show dominated objective vectors in stage i that are filtered out during the transformation.

Sequence Alignment Problem A particular case of a biobjective problem with binary objectives that can be solved by dynamic programming in polynomial amount of time is the pairwise sequence alignment problem [Roytberg et al., 1999]. Sequence alignment aims to identify regions of similarity in sequences of biological data, such as nucleotide and amino acid residues and it has many applications in Bioinformatics. The procedure consists of inserting gaps between the residues so that similar symbols from several sequences become aligned. The biobjective pairwise problem consists of finding alignments of two sequences that maximize the number of aligned symbols and minimize the number of inserted gaps. It can be reformulated as finding a path in a directed acyclic graph with polynomial number of nodes and edges and binary cost coefficients, leading to a linear number of nondominated solutions; see the graph formulation for the scalarized version of the problem in [Gusfield, 1997] and algorithms for the biobjective version in [Roytberg et al., 1999, Abbasi et al., 2013]. However, for an arbitrary number of sequences, the problem becomes NP-hard [Gusfield, 1997].

Open Questions Dynamic programming algorithms are often assumed inefficient for practical applications. However, recent progress in efficient implementations using, among others, preprocessing techniques and bound computations, has shown that dynamic programming algorithms can indeed be competitive in comparison with other exact solution methods (see Figueira et al. [2013] and Bazgan et al. [2009b]). A strong point in favor of dynamic programming algorithms is their versatility with respect to multiobjective problems. Moreover, dynamic programming algorithms can be used to derive polynomial time approximation schemes, see, for example, Erlebach et al. [2002] and Bazgan et al. [2009a] in the context of the multiobjective knapsack problem. The implications of structural properties of the considered problem instances may lead to new insights in this context.

3.4 Total Unimodularity

Inspired by the existence of polynomial time solvable single objective combinatorial optimization problems with totally unimodular constraint matrices, one might ask about the possibility of generalizing this result.

Definition 3.4. *A matrix $A \in \mathbb{R}^{m\times n}$ is* totally unimodular*, if every square submatrix of A has a determinant of $0, +1$ or -1.*

Total unimodularity has rarely been considered in the multiobjective context. There are only few articles that address this structural property in the presence of multiple objectives (Isermann [1979], Serafini [1987], Kouvelis and Carlson [1992], Williams [2002]). This might be explained by the fact that (linear) scalarization methods introduce new constraints (except for the weighted sum scalarization), which may destroy total unimodularity. Hence, at first glance, total unimodularity seems to help only in computing the supported nondominated points.

However, there are particular cases where the property of total unimodularity is compatible with the ϵ-constraint method, Brockhoff et al. [2015]. Obviously, these cases have objective functions with a particular structure which can be reformulated in terms of ϵ-constraints.

Theorem 3.5 (Maintaining total unimodularity at ϵ-constraint)**.** *Consider a multiobjective program with $q \geq 2$ objectives. Let $A \in \mathbb{R}^{m\times n}$ be a totally unimodular matrix and let $\tilde{C} \in \mathbb{R}^{(q-1)\times n}$ contain the cost vectors $c^l, l = 1, \dots, q-1$ with $c^l_i \in \{0, \pm 1\}$ for all i, l. Then the constraint matrix*

$$\tilde{A} := \begin{pmatrix} A & 0 \\ \tilde{C} & 1 \end{pmatrix}$$

is totally unimodular, if each cost vector $c^l, l = 1, \dots, q-1$ fulfills one of the following cases:

(i) $c^l = (0, ..., 0)$*, that is, the cost vector is a zero-valued vector.*
(ii) $c^l = a_{j\cdot}$ *or* $c^l = (-1) \cdot a_{j\cdot}$*, for some row* $a_{j\cdot}, j \in \{1, ..., m\}$ *of* A*.*
(iii) c^l *contains* $(n-1)$ *zero entries and exactly one* ± 1 *entry. That is, the cost vector is a row of the identity matrix or such a row multiplied by a scalar* -1*.*

(iv) *A has at most two non-zero entries for each column and there are columns with less than two non-zero entries. Further, for each column, where A has only a non-zero entry with value* 1*, there is at most one* c^l *that has an entry with value* -1 *in this column. Vice versa, for each column, where A has only a non-zero entry with value* -1*, there exists at most one* c^l *with an entry with value* 1*. For each zero-valued column of A, there are at most one entry with value* 1 *and at most one entry with value* -1 *in this column of* $\tilde{C}$*. All other entries of* $\tilde{C}$ *are zero.*

Proof. We prove this for $q = 2$, that is, we add one row c^l. As we can successively add rows, the proof for $q > 2$ follows immediately.

(i) Trivial.

(ii) Let $c^l = a_{j\cdot}$ and B be a submatrix of $\tilde{A}$. We only have to consider the cases, where B contains entries of c^l.

- Case 1: B does not contain entries of $a_{j\cdot}$. Then, $\det B \in \{0, \pm 1\}$, because B is a submatrix of A.
- Case 2: B contains entries of $a_{j\cdot}$. Then, B is singular and hence $\det B = 0$.

Now, if $c^l = (-1) \cdot a_{j\cdot}$, we do not have to consider the second case, as singularity is maintained. For case 1, we use the Laplace expansion. Then, we get

$$\det B = (-1) \cdot \det B(a_{j\cdot}) = (-1)^r \cdot \det \tilde{B}(a_{j\cdot}),$$

where $B(a_{j\cdot})$ is the matrix, where we substituted row c^l by $a_{j\cdot}$ and at $\tilde{B}(a_{j\cdot})$, we put the row $a_{j\cdot}$ in the right position, such that this matrix is a submatrix of A and therefore has a determinant of $0, \pm 1$. Further $r \in \{1, 2\}$, depending on the change of position of $a_{j\cdot}$. Hence, $\det B$ is $0, \pm 1$.

(iii) Let c^l be a row of the identity matrix and $c^l_i = 1$. Then, we use the Laplace Expansion at the first row of B:

$$\det B = 1 \cdot \det B_{1l} \in \{0, \pm 1\},$$

clearly B_{1l} is a submatrix of A. For $c^l_i = -1$, we get analogously $\det B = (-1) \cdot \det B_{1l}$.

(iv) In this case, $\tilde{A}$ suffices the requirements for the theorem by Hoffman and Kruskal [2010] and this implies the total unimodularity of $\tilde{A}$.

□

The following example shows that even one of the easiest objectives, a sum objective with all weights 1, destroys the property of total unimodularity.

Example 3.6. *If* c^l *only contains entries with value* 1*,* $\tilde{A} = \begin{pmatrix} c^l & 1 \\ A & 0 \end{pmatrix}$ *is in general not totally unimodular. We use the following totally unimodular matrix*

$$A = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{pmatrix}.$$

Total unimodularity of A can be proven by the theorem of Hoffman and Kruskal [2010]. If we now add a row with only 1 *entries, we get*

$$\tilde{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 \end{pmatrix}.$$

Now, if we choose rows 1 *and* 5 *and columns* 5 *and* 6*, the resulting submatrix is*

$$B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

with $\det B = -2 \notin \{0, \pm 1\}$*. Hence, $\tilde{A}$ is not totally unimodular.*
The same holds, when using a cost vector with only -1 *entries.*

Also, it is easy to see that, if the cost vector is equal to the sum or difference of two rows of A, $\tilde{A}$ is in general not totally unimodular.

Still, there are multiobjective combinatorial problems, where total unimodularity can be preserved while scalarizing. For example, the constraint matrix of the binary knapsack problem or the binary assignment problem are totally unimodular. Hence, there exist some instances, where both supported and nonsupported efficient solutions can be found efficiently. Nevertheless, our result does not hold for every instance of the particular problem (see Section 6.1).
However, if we consider a biobjective problem that fulfills Theorem 3.5, Brockhoff et al. [2015] showed that all solutions are supported and we get the whole nondominated set by dichotomic search (Aneja and Nair [1979]).

Conclusively, the requirements on the objective functions so that total unimodularity is preserved during scalarizations are very restrictive. Remark, that a complete characterization of all possible cost vectors that maintain total unimodularity for the scalarized problem is not achievable; for objective functions that are not affected by Theorem 3.5, the matrix $\tilde{A}$ has to be tested by the definition or the known equivalent propositions of total unimodularity.

4 Sum Objective Functions with Non-Binary Coefficients

In this section, we weaken the assumption of having binary coefficients in the objective functions and only demand bounded coefficients. We obtain a result similar to Theorem 3.1:

Corollary 4.1. *The cardinality of $\mathcal{Y}_N$ is also polynomially bounded by the size of the problem instance, if the value of the largest coefficient of $q-1$ objective functions, i. e.* 2-$\max_{l \in \{1,\dots,q\}} \max_{i \in \{1,\dots,n\}} c_i^l$*, is polynomially bounded in the input size n.*

Corollary 4.1 implies that the size of the nondominated set $\mathcal{Y}_N$ remains polynomial for all MOCO problems with $q \geq 2$ objective functions, if for $q-1$ of them their coefficients grow at most polynomially with the instance size.
This immediately transfers to the complexity of the dynamic programming processes of Section 3.3, since in general the number of nondominated objective vectors in a state can be bounded by the same value, say $O(\ell)$, where ℓ is a polynomial of the instance size. These can be filtered for dominance in $O(\ell^2)$, for example, by straightforward pairwise comparisons, leading to an overall polynomial time algorithm.

Besides dynamic programming there are other noteworthy applications. Exemplarily, we present results on a special case of the biobjective minimum spanning tree problem.

4.1 Minimum Spanning Trees

Seipp [2013] considered the biobjective minimum spanning tree problem with cost coefficients in $\{0, 1, 2\}$. He first proved that all efficient solutions are supported and all efficient solutions are connected. Moreover, he showed that the cardinality of the nondominated set is bounded from above by a polynomial in the size of the underlying graph and, more precisely, it holds $|\mathcal{Y}_N| \leq 2n - 1$. This bound is tight. Interestingly, this cost structure causes that all efficient points can be computed by solving a weighted sum scalarization problem with three different values of λ only.

5 Bottleneck Objective Functions

In a bottleneck objective function (or generalization of it), one selects exactly one cost coefficient out of n possibilities. In the presence of q objective functions, the cardinality of $|\mathcal{Y}_N|$ is in $O(n^q)$, which is a polynomial in the input length of the instance. In the following, we first present results for multiobjective problems with (generalized) bottleneck objective functions. Then, the same type of function is considered to achieve a representation of the whole nondominated set that meets certain requirements.

5.1 Bottleneck and Generalized Bottleneck Objectives

In the context of multiobjective optimization, Gorski [2010] and Gorski et al. [2012a] considered problems with one general objective function $f : \mathcal{X} \to \mathbb{R}$ which could, for example, be the weighted sum of cost coefficients, and $(q-1)$ k-max objectives with possibly different values of k and different cost coefficients $c^2, \ldots, c^q$:

$$\begin{aligned} \min \quad & \left(\sum_{e \in x} c^1(e),\ k_2\text{-}\max_{e \in x} c^2(e), \ldots, k_q\text{-}\max_{e \in x} c^q(e) \right) \\ \text{s.t.} \quad & x \in \mathcal{X}. \end{aligned}$$

Since k-max objective functions can attain at most $|\mathcal{E}| = n$ different values, a bound on the cardinality of the nondominated set similar to the case of binary coefficients (c.f. Theorem 3.1 in Section 3) can be obtained (see Gorski et al. [2012a]):

Theorem 5.1. *Consider a multiobjective combinatorial optimization problem with one arbitrary objective function and with* $(q-1)$ k-max *objective functions. Then the cardinality of the nondominated set* $\mathcal{Y}_N$ *is bounded by* $(n+1)^{q-1}$*, i.e.,* $|\mathcal{Y}_N| \leq (n+1)^{q-1}$.

This result has been used in Gorski et al. [2012a] to derive efficient solution methods for problems with one sum objective and $(q-1)$ k-max objectives. These methods are based on the recursive solution of at most $O(n^{q-1})$ ϵ-constraint scalarizations of this class of problems, with ϵ-values defined by the cost coefficients in the respective objectives. These ϵ-constraint scalarizations can be reformulated by assigning binary weights $\hat{c}^i(e) \in \{0,1\}$ to all elements in $\mathcal{E}$ such that, in objective i, $2 \leq i \leq q$, $\hat{c}^i(e) = 1$ if and only if $c^i(e) > \epsilon$. We thus want to minimize objective one (i.e., $\sum_{e\in x} c^1(e)$) under the constraints that the weighted sum of the binary coefficients (i.e., $\sum_{e\in x} \hat{c}^i(e)$) stays below k_i in objective i, $i = 2,\ldots,q$:

$$\begin{aligned} \min \quad & \sum_{e\in x} c^1(e) \\ \text{s. t.} \quad & \sum_{e\in x} \hat{c}^i(e) \leq k_i, \quad i = 2,\ldots,q \\ & x \in \mathcal{X}. \end{aligned}$$

Hard Cases Budget constrained versions of combinatorial optimization problems are NP-hard in general, see, for example, Garey and Johnson [1979]. The fact that the constraints have binary coefficients may lead to polynomial special cases. An example where this can be proven is given below.

Easy Cases Similar to the discussion in Section 3, ϵ-constraint scalarizations with binary coefficients can be solved in polynomial time if, for example, the underlying problem is a matroid. Moreover, efficient dynamic programming implementations can be realized whenever the problem structure allows this. This has been utilized, for example, for the case of knapsack problems with k-min objectives in Rong et al. [2013].

Open Questions It is an interesting open question to analyse which types of budget constrained combinatorial optimization problems can be solved in polynomial time in the case that all budget constraints have binary coefficients.

5.2 Representations of the Nondominated Set

Uniformity, coverage and ϵ-indicator are often used quality measures for representing subsets of the set of nondominated points (Sayin [2000], Ruzika and Wiecek [2005], Vassilvitskii and Yannakakis [2005]). Since they focus on different aspects of a good representation, one can integrate them in a multiple objective setting and find a triobjective representation problem given a maximal cardinality of the representing subset.

Easy Cases For the nondominated set $\mathcal{Y}_N$ of a discrete biobjective optimization problem, finding a representing subset $R \subseteq \mathcal{Y}_N$, which is Pareto optimal with respect to these three quality measures, can be solved in polynomial time. For biobjective minimization problems the triobjective representation problem optimizing uniformity, coverage and ϵ-indicator can be stated as follows:

$$\begin{aligned}
&\max \min_{\substack{r_i, r_j \in R \\ r_i \neq r_j}} ||r_i - r_j|| \\
&\min \max_{y \in \mathcal{Y}_N} \min_{r \in R} ||r - y|| \\
&\min \max_{y \in \mathcal{Y}_N} \min_{r \in R} \max_{i \in \{1,2\}} \frac{y^i}{r^i} \\
&\text{s.t. } R \subseteq \mathcal{Y}_N \\
&\qquad |R| \leq k.
\end{aligned}$$

Since we only consider pairwise nondominated points in $\mathbb{R}^2$, we can presume that they are positioned on a line. In the following, we will assume that the nondominated points are sorted with respect to their first component. Together with the bottleneck objective functions this makes two solution approaches applicable: dynamic programming and threshold algorithms.

Due to the bottleneck objectives, there is only a polynomial number of feasible solutions. Since the corresponding feasibility problems can be reformulated to longest path problems, which are efficiently solvable on these special type of graphs, a threshold algorithm can solve this triobjective representation problem in a polynomial amount of time of the input size. The Bellman principle of optimality holds, if one considers in each state $t(i, j)$ only these partial solutions, which consist of i points up to point j and contain this point in the representing subset. Thus, any representation problem for a biobjective discrete optimization problem using a combination of these quality measures as objectives can be solved in polynomial time by either dynamic programming or a threshold algorithm (for details see Vaz et al. [2014]).

Hard Cases Solving the uniformity problem for the nondominated set of multiobjective discrete optimization problems is NP-hard in general, since it is equivalent to the geometric dispersion problem (proven by Baur and Fekete [2001]). However, it is shown that the uniformity problem is efficiently solvable for the nondominated points of a biobjective problem.

Open Questions Is the uniformity problem NP-hard for the nondominated points of a triobjective problem? To the best of our knowledge there is no proof for this.

6 On the Borderline of Hard and Easy

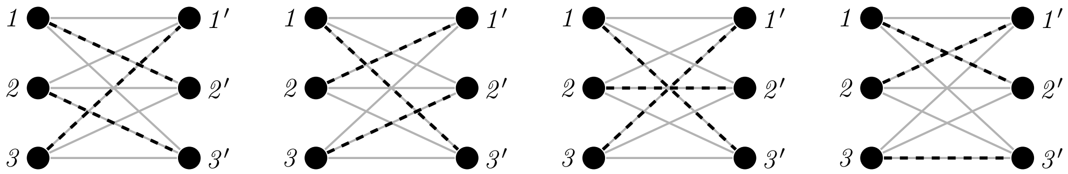
So far, we have seen both, easy and hard cases for numerous multiobjective combinatorial problems. As we observed, the separation of these two categories always depends on the problem structure itself and hence there is no all-embracing condition for having an easy case. The adjustments that are necessary for a distinct MOCO problem always differ, which leads us to three extreme cases, while we traverse the borderline between hard and easy:

- First, we present a particular problem, which, regarding its structure, would fit into Section 3, but nevertheless remains hard to solve. Very strict additional requirements are necessary to obtain an easy case, although similar problem cases of Section 3 are easy even without additional requirements. Figuratively speaking, the borderline for this problem is very thick compared to other problems.
- Second, combining two cases of a problem with different additional requirements for the input of an instance can result in not only an easy but trivial problem case, although the original cases belong to hard cases in general. Hence, we have a thin but sharp borderline that passes directly through to triviality.
- A third approach is to constrain the nondominated set one is interested in. In particular, we omit nonsupported solutions and observe cases, where the set of nondominated supported solutions can be easily obtained. This leads us to the conclusion that the borderline not only separates different problem cases but also divides the nondominated set $\mathcal{Y}_N$.

We start with the former case that tackles the biobjective assignment problem.

6.1 Biobjective Assignment Problem with One Binary Objective

In this section, we consider a special case of the biobjective assignment problem. By allowing only binary coefficients in one objective, we obtain a simplification of the generally NP-complete and intractable binary assignment problem. We show that the number of nondominated points is polynomially bounded in the instance size. However, there exist nonsupported nondominated points and the complexity of the corresponding decision problem is still an open question. In Papadimitriou [1984] the constrained bipartite matching problem with bounded coefficients in the constraint is mentioned as a problem with unknown complexity. Alfakih and Murty [1998] state that the complexity of the equality constrained assignment problem with binary coefficients in the constraint up to future research. In contrast to the general biobjective assignment problem, for which Serafini [1987] proved NP-completeness by reduction to Partition, for the version with binary coefficients there is neither a polynomial algorithm nor a NP-completeness proof. Since the supported nondominated points can be computed in polynomial time using the Hungarian method (Kuhn [1955]), the computational complexity of the problem is due to the nondominated nonsupported points.



(a) Assignment 1 (b) Assignment 2 (c) Assignment 3 (d) Assignment 4

Figure 2: The four efficient assignments in Example 6.1

We consider a biobjective assignment problem with only binary coefficients in the second objective function (i. e. $c^2_{ij} \in \{0,1\}\ \forall i,j \in \{1,\dots,n\}$):

$$\begin{aligned}
\min \quad & f^1(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c^1_{ij} x_{ij} \\
\min \quad & f^2(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c^2_{ij} x_{ij} \\
\text{s. t.} \quad & \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i = 1,\dots,n \\
& \sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j = 1,\dots,n \\
& x \in \{0,1\}^{n\times n}.
\end{aligned}$$

The following example shows: There are nonsupported nondominated points in the outcome space even in this simplified version of the biobjective assignment problem.

Example 6.1. *Consider the biobjective assignment problem with 3+3 nodes and the objective function coefficients*

$$C^1 = (c^1_{ij}) = \begin{pmatrix} 1 & 1 & 4 \\ 3 & 4 & 1 \\ 4 & 4 & 1 \end{pmatrix} \qquad \text{and} \qquad C^2 = (c^2_{ij}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then we obtain four efficient solutions, depicted in Figure 2. The objective function values of these four assignments are $(6,2)$*,* $(11,1)$*,* $(12,0)$*, and* $(5,3)$*, respectively. It is easy to see that the objective vector of Assignment 2 (Figure 2(b)) is indeed nonsupported (c. f. Figure 3).*
This does not contradict the results of Section 3.4, as the constraint matrix of the ϵ*-constraint scalarization is not totally unimodular.*

Corollary 6.2. *The biobjective assignment problem with one binary objective has at most* $n+1$ *nondominated points.*

Proof. Follows directly from Theorem 3.1. □

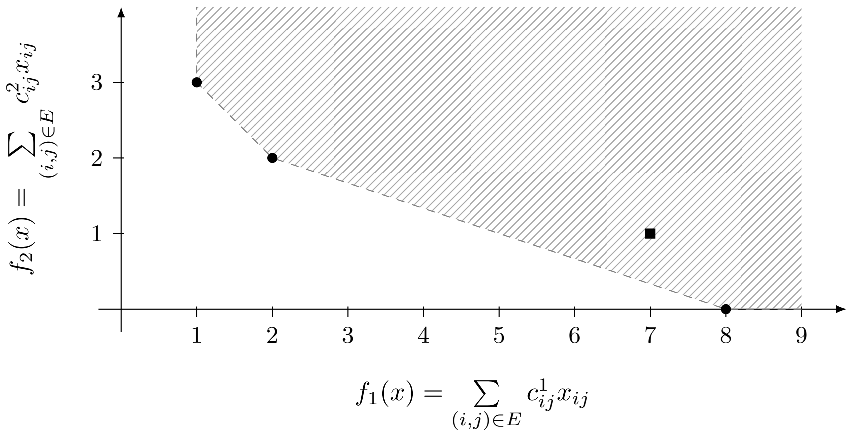


Figure 3: Supported nondominated points are depicted circles, nonsupported nondominated points as rectangles, the $\mathbb{R}^2_+$-convex hull is illustrated by the dashed line.

Easy Cases While the complexity of this biobjective assignment problem with binary coefficients is not known, surely easy instances of it are the ones in which the second objective function maintains the total unimodularity, i.e. the corresponding ϵ-constraint problem is totally unimodular (cf. Subsetion 3.4).

Open Questions Are there other properties than the totally unimodular ϵ-constraint subproblems, which make the assignment problem easy?

6.2 Biobjective Knapsack Problem with Additional Restrictions

In the following, we tackle the second case mentioned above. We summarize some particular interesting variants of the biobjective knapsack problem. The results are based on Gomes da Silva et al. [2004]. Three cases will be presented and we examine their properties and algorithmic aspects for computing the set of efficient solutions. The general integer programming formulation of the biobjective knapsack problem is as follows:

$$
\begin{aligned}
\max \quad & f^1(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_i^1 x_i \\
\max \quad & f^2(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_i^2 x_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W \\
& x \in \{0,1\}^n
\end{aligned}
$$

with $W \geq 0$ and $c_i^1, c_i^2, w_i \geq 0, \forall i = 1, \ldots, n$. Recall that biobjective knapsack problems can be solved in pseudopolynomial time by dynamic programming, c. f. Section 3.3 above.

For further presentation of results, we introduce *Iso-item lines*, which consist of objective vectors of feasible solutions with the same number of elements:

$$ISO(\delta) = \left\{ (f^1(x), f^2(x)) : \sum_{i=1}^{n} x_i = \delta \right\}, \quad \text{for } \delta \in \mathbb{N}_0.$$

Case 1: All Items of the Same Weight At first, we consider the case of all weights being constant, i.e. $w_i = w \geq 0$. Let $\ell := \lfloor \frac{W}{w} \rfloor$. Then all efficient solutions satisfy $\sum_{i=1}^{n} x_i = \ell$. This constraint is also known as *cardinality constraint*. We refer to the biobjective knapsack problem with cardinality constraint as *Case 1* and get the following result:

Lemma 6.3. *The set of nondominated points is contained in the ISO-item line for ℓ: $\mathcal{Y}_N \subseteq ISO(\ell)$. In other words $\mathcal{Y}_N = ISO(\ell)_N$.*

Also, as the cardinality constraint is the only given constraint, the constraint matrix is totally unimodular, i. e., all extreme points of the convex hull of the set of feasible solutions $\mathcal{X}$ are integral. As a consequence, all supported nondominated points can be computed using a biobjective simplex algorithm. However, not every efficient solution is supported. A small example with $\ell = 1$ contains three items with profits $(4, 1), (2, 2)$ and $(1, 5)$. Here, $(2, 2)$ is nondominated but nonsupported. Nonsupported solutions can, for example, be generated within a 2-phase method using a ranking procedure, Murty [1983]. However, in this case we can not guarantee that every instance of Case 1 is solvable in polynomial time. To the contrary, it can be shown that the problem is NP-hard in general.

Theorem 6.4 (Case 1: Same weights)**.** *The decision problem corresponding to the biobjective knapsack problem with equal weights, namely is there a feasible knapsack solutions $x \in \{\{0, 1\}^n : \sum_{i=1}^{n} x_i = \ell\}$ such that $\sum_{i=1}^{n} c_i^1 x_i \geq b_1$ and $\sum_{i=1}^{n} c_i^2 x_i \geq b_2$, is NP-complete.*

Proof. Let $M > \max_{i=1,\ldots,n} c_i^2$ be a sufficiently large integer and consider the single objective knapsack problem with fixed cardinality

$$\max \left\{ \sum_{i=1}^{n} c_i^1 x_i \ : \ \sum_{i=1}^{n} (M - c_i^2) x_i \leq (\ell M - b_2), \ \sum_{i=1}^{n} x_i = \ell \right\}.$$

The corresponding decision problem is NP-complete, see Caprara et al. [2000], and it is euqivalent to the above formulation. □

Case 2: All Items with the Same Sum of Objective Coefficients In the following, we extend the biobjective knapsack problem by demanding that the cost coefficients of all items add up to a constant $\alpha \geq 0$, that is $c_i^1 + c_i^2 = \alpha$ for every item $i = 1, \ldots, n$. Hence, the objective functions can be reformulated to

$$\max\Big(\sum_{i=1}^{n} c_i^1 x_i,\ \sum_{i=1}^{n}(\alpha - c_i^1)x_i\Big).$$

If we consider the sum of these two objectives, each item generates a profit of α. Hence, for an ISO-item line with $\tilde{n}$ items, the total profit for the sum objective is $\tilde{n}\alpha$ for every nondominated point of $ISO(\tilde{n})$. So, the sum of profits increases with increasing cardinality, see Figure 4. This gives us the possibility to state more results for this case:

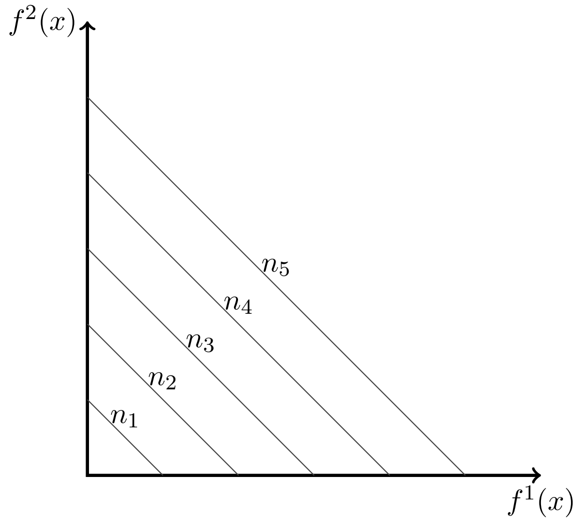


Figure 4: Structure of the outcome space with several cardinalities $n_i = i$. The isolines (lines with same objective value sum) mark possible regions for nondominated points and contain the ISO-item line for the respective cardinality.

Theorem 6.5 (Case 2: Constant sum)**.** *Let x and x' be feasible for Case 2 of the biobjective knapsack problem.*

(i) If $f(x), f(x') \in ISO(\tilde{n})$, then they do not dominate each other.

(ii) If $f(x) \in ISO(n_1)$ and $f(x') \in ISO(n_2)$ with $n_1 < n_2$, then $f(x)$ does not dominate $f(x')$.

(iii) Let n_{max} be the maximal number of items of a solution in $\mathcal{X}$. Then every $f(x) \in ISO(n_{max})$ is nondominated.

Proof. (i) As both objective vectors are in $ISO(\tilde{n})$, we have $f^1(x) + f^2(x) = \tilde{n}\alpha = f^1(x') + f^2(x')$ and therefore, $f(x)$ cannot dominate $f(x')$ and vice versa.

(ii) As $f^1(x) + f^2(x) = n_1\alpha < n_2\alpha = f^1(x') + f^2(x')$, $f(x)$ cannot dominate $f(x')$.

(iii) This follows immediately from *(i)* and *(ii)*.

□

Further, by the first result of Theorem 6.5, we get: If we add a cardinality constraint to Case 2, all efficient solutions are supported. Also, remark that with the cardinality constraint we have a problem of type Case 1. This implies a procedure for solving biobjective knapsack problems with a constant sum of coefficients: We can solve the problem for each cardinality and afterwards compare the obtained solutions. Theorem 6.5 suggests starting with the highest cardinality and we do not need to compare solutions of the same cardinality. We can improve this procedure by applying it as a dynamic programming algorithm, which is a special case of the single objective dynamic programming algorithm for the $\{0,1\}$-knapsack problam. However, although all nondominated points are supported for one ISO-item line, Case 2 is in general not an easy case of the biobjective knapsack problem. In particular, the number of possible ISO-item lines is in general not bounded by a polynomial in the input size of an instance.

Case 3: All Items of the Same Weight and with the Same Sum of Objective Coefficients Again, we extend our consideration by combining the requirements of the previous setups. We assume that the sum of objective coefficients is constant and all items have the same weights.

$$\begin{aligned} \max \quad & \left(\sum_{i=1}^{n} c_i^1 x_i,\ \sum_{i=1}^{n} (\alpha - c_i^1) x_i\right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = \ell, \\ & x_i \in \{0,1\}, \qquad i = 1,2,\ldots,n. \end{aligned}$$

Here, we can combine the results of the Cases 1 and 2 and immediately state the following theorem:

Theorem 6.6 (Case 3: Same weights and constant sum)**.**

(i) Every solution in $ISO(\ell)$ is nondominated. In particular, every feasible solution is efficient.

(ii) The set of efficient solutions is connected and the extreme points of its convex hull are integer valued.

The computation of all efficient solution requires simply the computation of all combinations using ℓ items. By Theorem 6.6, these solutions are connected and supported and can be found by a single objective simplex method (with $f^1(x)$ as objective function and bounded variables). Also, one can use Murty's ranking procedure, because this procedures computes all adjacent solutions to a starting point that do not decrease the objective function value. In particular, Case 3 is a trivial variant of the biobjective knapsack problem.

6.3 Problems with a Polynomial Number of Supported Solutions

Seipp [2013] considered the minimum spanning tree problem and showed a somewhat surprising result (and extended an earlier result of Chandrasekaran [1977]): for an arbitrary but fixed number of objective functions, the cardinality of the set of nondominated extreme points is polynomially bounded in the input length even of the minimum spanning tree instance. This complements the intractability result for the multiobjective minimum spanning tree problem by delivering more information about a "small" subset of nondominated points.

The idea—which can be easily generalized to some matroid problems—used by Seipp to show this result can be briefly sketched as follows. Consider the *weight space*

$$W^0 := \left\{ \lambda \in \mathbb{R}^q : \lambda > 0,\ \sum_{i=1}^{q} \lambda_i = 1 \right\},$$

which is the set of all (reasonable) weights for the weighted-sum method. It can be shown that a feasible solution $x \in X$ of MOCO is an extreme supported solution. Its objective value $y = f(x) \in Y$ is a nondominated extreme point if and only if there exists a strictly positive weighting vector $\lambda \in \mathbb{R}^q$ such that y is the unique optimal solution of $\min_{\bar{y} \in Y} \lambda^\top \bar{y}$. Equivalently, for some $\lambda \in \mathbb{R}^q_>$ all minimizers of the weighted sum problem are equivalent efficient solutions and have objective value y.

For the multiobjective minimum spanning tree problem, it should be noted that the weighted sum problem is again a minimum spanning tree problem which can be denoted by

$$\begin{aligned} \min \quad & \sum_{l=1}^{q} \lambda_i \sum_{e \in T} c^l(e) = \sum_{e \in T} c_e^\lambda \\ \text{s.t.} \quad & T \in \mathcal{T}, \end{aligned}$$

where $\mathcal{T}$ denotes the set of all spanning trees in the given graph, T denotes some specific tree, e denotes an edge in the tree, and $c^l(e)$ the l-th cost value of edge e, $l = 1, \ldots, q$. This implies that the weighted sum problem can be solved in polynomial time by some greedy algorithm (e. g. Kruskal's or Prim's algorithm). It should be emphasized that the result of the greedy algorithm depends on the sorting of the edges in order of increasing cost only and not on the specific cost values. Thus, each $\lambda \in W^0$ implies a sorting and for $\lambda_1, \lambda_2 \in W^0$, the corresponding weighted sum problems yield the same solution, if the sortings of the edges are the same. On the other hand, several different sortings of the edges may still lead to the same minimum spanning tree. Thus, if W^0 can be subdivided in a polynomial number of subsets such that all weights in one subset imply the same sorting, a polynomial upper bound for the number of nondominated extreme points is obtained.

To establish this subdivision of W^0, subsets of weights, which generate the same ordering of the weighted edge costs, are found. For two edges $e, f \in E$ with $c_e \neq c_f$ consider the set

$$h(e,f) := \{\lambda \in \mathbb{R}^q : c_e^\lambda = c_f^\lambda\} = \{\lambda \in \mathbb{R}^q : \langle \lambda, c_e - c_f \rangle = 0\},$$

which is the set of all weights implying the same weighted edge costs for e and f. Note that $h(e, f) \subset \mathbb{R}^q$ is a separating hyperplane through the origin. It divides $\mathbb{R}^q$ into two half-spaces, one of which corresponds to the set of all weighting vectors for which edge e is cheaper than edge f (and the other one vice versa).

Consider now the set H of all such hyperplanes $h(e, f)$ for $e, f \in E$. This set dissects $\mathbb{R}^q$ into q-dimensional polyhedral subsets. Note that there is a one-to-one relationship between these polyhedral subsets and the sortings of the weighted edge costs. Thus, counting the number of polyhedral subsets yields an upper bound for the number of nondominated extreme points. This set H induces a so-called *arrangement of hyperplanes*. Seipp first proved the following theorem.

Theorem 6.7 (Seipp [2013])**.** *Let* $C := \{c^l : l \in I\} \subseteq \mathbb{R}^q \setminus \{0\}$ *denote a discrete set of nonzero vectors with underlying index set* $I := \{1, \ldots, n\}$*. Then the family of linear hyperplanes* $h_l := h^=(c^l) = \{\lambda \in \mathbb{R}^q : \langle \lambda, c^l \rangle = 0\}$*,* $l \in I$*, which are induced by vectors in* C *divides* $\mathbb{R}^q$ *into at most* $2^{q+1} \cdot n^q$ *full-dimensional polyhedral subsets.*

This implies for the minimum spanning tree problem the following polynomial bound on the number of extremal supported nondominated points.

Theorem 6.8 (Seipp [2013])**.** *Let* P *be an instance of the* q*-objective minimum spanning tree problem with underlying graph* $G = (V, E)$*. Then the number of extremal supported nondominated points of* P *is of the order* $O(m^{2q})$*, where* $m = |E|$ *denotes the number of edges of* G*.*

This observation can be carried over to some matroid problems, since it basically relies on the assumption that the occuring single objective problems can be solved by a greedy algorithm and that the solution solely depends on the sorting of some items.

Seipp later improved his intially found bound.

Theorem 6.9 (Seipp [2013])**.** *Let* P *be an instance of the* q*-objective minimum spanning tree problem with underlying graph* $G = (V, E)$*. Then the number of extremal supported nondominated points of* P *is bounded from above by* $2 \cdot m^{2(q-1)}$*, where* $m = |E|$ *denotes the number of edges of* G*.*

Note that this finding has quite some consequences. Seipp provides several insights in the structure of the weight space since his arrangement of hyperplanes induces a weight space decomposition. Moreover, the dichotomic search (Aneja and Nair [1979]) is now guaranteed to terminate in a polynomial number of steps for the multiobjective minimum spanning tree problem. This influences e. g. the first phase in the so-called two-phases method (Visée et al. [1998]).

7 Concluding Remarks

Despite general intractability and NP-hardness results for many multiobjective combinatorial optimization problems (cf. Ehrgott [2005], Ruzika and Hamacher [2009]), there exist cases of "easy" MOCO problems. These easy MOCO problems exhibit (a) a "small" nondominated set and (b) a decision problem belonging to the complexity class P.

The first requirement, i. e. a "small" nondominated set, means that the number of nondominated points grows at most polynomially with the size of the input data. This can be guaranteed, for example, for MOCO problems with sum objective functions having binary coefficients or for MOCO problems involving bottleneck objective functions.

However, a small nondominated set is in general not sufficient to guarantee that a MOCO problem is easy. Only in combination with additional properties like, for example, total unimodularity, matroid optimization, or when efficient greedy or dynamic programming implementations are available, a minimal complete set of efficient solutions can be computed in polynomial time.

It should be pointed out that the borderline between easy and hard problems is not sharp. For example, if one strives for computing only a certain subset of a minimal complete set of efficient solutions (e. g. those solutions being supported extreme points), this might change the status of difficulty.

It should be pointed out that this categorized collection of "easy" and "halfway easy" multiobjective combinatorial optimization problems is by no means complete – it is better regarded as a starting point of more intense research in this direction.

Acknowledgments This work was supported by the bilateral cooperation project *Tractability in multiobjective combinatorial optimization* funded by the Deutscher Akademischer Austauschdienst (DAAD, Project-ID 57128839) and Fundação para a Ciência e Tecnologia. The research of Pascal Halffmann was supported by DFG grant RU 1524/4-1 and David Willems' activities were funded by BMBF project 13N12825.

References

M. Abbasi, L. Paquete, A. Liefooghe, M. Pinheiro, and P. Matias. Improvements on bicriteria pairwise sequence alignment: algorithms and applications. *Bioinformatics*, 29(8):996–1003, 2013.

A. Y. Alfakih and K. G. Murty. Adjacency on the constrained assignment problem. *Discrete Applied Mathematics*, 87(1–3):269–274, 1998.

Y. P. Aneja and K. P. Nair. Bicriteria transportation problem. *Management Science*, 25(1):73–78, 1979.

C. Baur and S. P. Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30(3):451–470, 2001.

C. Bazgan, H. Hugot, and D. Vanderpooten. Implementing an efficient fptas for the 0–1 multi-objective knapsack problem. *European Journal of Operational Research*, 198(1): 47–56, 2009a.

C. Bazgan, H. Hugot, and D. Vanderpooten. Solving efficiently the 0–1 multi-objective knapsack problem. *Computers & Operations Research*, 36(1):260–279, 2009b.

R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

F. Bökler, M. Ehrgott, C. Morris, and P. Mutzel. Output-sensitive complexity of multiobjective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 2016. submitted.

D. Brockhoff, M. Ehrgott, J. R. Figueira, L. Martí, L. Paquete, M. Stiglmayr, and D. Vanderpooten. Computational complexity. In S. Greco, K. Klamroth, J. D. Knowles, and G. Rudolph, editors, *Understanding Complexity in Multiobjective Optimization (Dagstuhl Seminar 15031)*, volume 5, pages 117–121, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik.

A. Caprara, H. Kellerer, U. Pferschy, and D. Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *European Journal of Operational Research*, 123:333–345, 2000.

R. Chandrasekaran. Minimal ratio spanning tree. *Networks*, 7:335–342, 1977.

M. Ehrgott. On matroids with multiple objectives. *Optimization*, 38:73–84, 1996.

M. Ehrgott. Hard to say it's easy – four reasons why combinatorial multiobjective programmes are hard. In Y. Y. Haimes and R. E. Steuer, editors, *Research and Practice in Multiple Criteria Decision Making*, volume 487 of *Lecture Notes in Economics and Mathematical Systems*, pages 69–80. Springer Berlin Heidelberg, 2000.

M. Ehrgott. *Multicriteria Optimization.* Springer, 2005.

M. Ehrgott and X. Gandibleux. A survey and annoted bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22(4):425–460, 2000.

M. Ehrgott and K. Klamroth. Connectedness of efficient solutions in multiple criteria combinatorial optimization. *European Journal of Operational Research*, 97(1):159–166, 1997.

T. Erlebach, H. Kellerer, and U. Pferschy. Approximating multiobjective knapsack problems. *Management Science*, 48(12):1603–1612, 2002.

J. R. Figueira, L. Paquete, M. Simões, and D. Vanderpooten. Algorithmic improvements on dynamic programming for the bi-objective $\{0, 1\}$ knapsack problem. *Computational Optimization and Applications*, 56(1):97–111, 2013.

H. N. Gabow and R. E. Tarjan. Efficient algorithms for a family of matroid intersection problems. *Journal of Algorithms*, 5:80–131, 1984.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979. ISBN 0-7167-1044-7.

C. Gomes da Silva, G. Clímaco, and J. R. Figueira. Geometrical configuration of the Pareto frontier of bi-criteria $\{0,1\}$-knapsack problems. Research Report 16-2004, INESC-Coimbra, Portugal, 2004.

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

J. Gorski and S. Ruzika. On k-max optimization. *Operations Research Letters*, 37:23–26, 2009.

J. Gorski, K. Klamroth, and S. Ruzika. Connectedness of efficient solutions in multiple objective combinatorial optimization. *Journal of Optimization Theory and Applications*, 150(3):475–497, 2011.

J. Gorski, K. Klamroth, and S. Ruzika. Generalized multiple objective bottleneck problems. *Operations Research Letters*, 40:276–281, 2012a.

J. Gorski, F. Paquete, and F. Pedrosa. Greedy algorithms for a class of knapsack problems with binary weights. *Computers & Operations Research*, 39(3):498–511, 2012b.

F. Guerriero and R. Musmanno. Label correcting methods to solve multicriteria shortest path problems. *Journal of Optimization Theory and Applications*, 111(3):589–613, 2001.

D. Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, 1997.

Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems Man and Cybernetics*, 1(1):296–297, 1971.

A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra. In *50 Years of Integer Programming 1958-2008*, pages 49–76. Springer, 2010.

H. Isermann. The enumeration of all efficient solutions for a linear multiple-objective transportation problem. *Naval Research Logistics Quarterly*, 26(1):123–139, 1979.

P. Kouvelis and R. C. Carlson. Total unimodularity applications in bi-objective discrete optimization. *Operations Research Letters*, 11(1):61–65, 1992.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

K. Murty. *Linear Programming.* John Wiley & Sons, New York, USA, first edition, 1983.

J. Oxley. *Matroid Theory.* Oxford Mathematics, 2004.

C. H. Papadimitriou. Polytopes and complexity. In W. Pulleyblank, editor, *Progress in Combinatorial Optimization*. Academic Press, Canada, 1984.

A. Rong, K. Klamroth, and J. Figueira. Multicriteria 0-1 knapsack problems with k-min objectives. *Computers and Operations Research*, 40:1481–1496, 2013.

M. A. Roytberg, M. N. Semionenkov, and O. I. Tabolina. Pareto-optimal alignment of biological sequences. *Biophysics*, 44(4):565–577, 1999.

S. Ruzika and H. W. Hamacher. A survey on multiple objective minimum spanning tree problems. In J. Lerner, D. Wagner, and K. Zweig, editors, *Algorithmics, Lecture Notes on Computer Science 5515*, pages 104–116. Springer-Verlag Berlin Heidelberg, 2009.

S. Ruzika and M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, 2005.

S. Sayin. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.

F. Seipp. *On Adjacency, Cardinality, and Partial Dominance in Discrete Multiple Objective Optimization*. PhD thesis, TU Kaiserslautern, Germany, 2013.

P. Serafini. Some considerations about computational complexity for multi objective combinatorial problems. In J. Jahn and W. Krabs, editors, *Recent Advances and Historical Development of Vector Optimization*, volume 294 of *Lecture Notes in Economics and Mathematical Systems*, pages 222–232. Springer, 1987.

R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 1986.

E. L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3:83–104, 1994.

S. Vassilvitskii and M. Yannakakis. Efficiently computing succinct trade-off curves. *Theoretical Computer Science*, 348(2âĂŞ3):334 – 356, 2005. Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004)Automata, Languages and Programming: Algorithms and Complexity 2004.

D. Vaz, L. Paquete, C. M. Fonseca, K. Klamroth, and M. Stiglmayr. Representation of the non-dominated set in biobjective combinatorial optimization. *Computers & Operations Research*, 2014. accepted for publication.

M. Visée, J. Teghem, M. Pirlot, and E. Ulungu. Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem. *Journal of Global Optimization*, 12(2):139–155, 1998.

J. C. Williams. A linear-size zero-one programming model for the minimum spanning tree problem in planar graphs. *Networks*, 39(1):53–60, 2002.

L. Zadeh. Optimality and non-scalar-valued performance criteria. *Automatic Control, IEEE Transactions on*, 8(1):59–60, 1963.