Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 18/25

B. Schulze, K. Klamroth und M. Stiglmayr

Multi-Objective Unconstrained Combinatorial Optimization: A Polynomial Bound on the Number of Extreme Supported Solutions

January 7, 2019

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

Noname manuscript No.
(will be inserted by the editor)

Multi-Objective Unconstrained Combinatorial Optimization: A Polynomial Bound on the Number of Extreme Supported Solutions

Britta Schulze · Kathrin Klamroth · Michael Stiglmayr

the date of receipt and acceptance should be inserted later

Abstract The multi-objective unconstrained combinatorial optimization problem (MUCO) can be considered as an archetype of a discrete linear multi-objective optimization problem. It can be interpreted as a specific relaxation of any multi-objective combinatorial optimization problem with linear sum objective function. While its single criteria analogon is analytically solvable, MUCO shares the computational complexity issues of most multi-objective combinatorial optimization problems: intractability and NP-hardness of the ε-constraint scalarizations.

In this article interrelations between the supported points of a MUCO problem, arrangements of hyperplanes and a weight space decomposition, and zonotopes are presented. Based on these interrelations and a result by Zaslavsky on the number of faces in an arrangement of hyperplanes, a polynomial bound on the number of extreme supported solutions can be derived, leading to an exact polynomial time algorithm to find all extreme supported solutions. It is shown how this algorithm can be incorporated into a solution approach for multi-objective knapsack problems.

Keywords multi-objective combinatorial optimization · multi-objective unconstrained optimization · weight space decomposition · arrangement of hyperplanes · zonotopes · knapsack problem

Britta Schulze
University of Wuppertal, Gaußstr.20, 42119 Wuppertal
Tel.: +49-202-439-3489
E-mail: schulze@math.uni-wuppertal.de

Kathrin Klamroth
University of Wuppertal
E-mail: klamroth@math.uni-wuppertal.de

Michael Stiglmayr
University of Wuppertal
E-mail: stiglmayr@math.uni-wuppertal.de

1 Introduction and notation

Seeking exact solutions for a multi-objective combinatorial optimization problem (MOCO) is a classical example for a global optimization problem. In addition to being intractable most MOCO problems have a majority of solutions that are unsupported, i. e., that can not be found by convex optimization methods [see e. g. 14, for a recent review]. Despite this general difficulty, it will be shown that a representative subset of nondominated points that is of tractable size can be computed efficiently. A prototypical multi-objective discrete problem is the *multi-objective unconstrained combinatorial optimization problem* MUCO which is defined as

$$\begin{aligned} \max \quad & f(x) = \left(\sum_{i=1}^{n} p_{1,i}\, x_i, \ldots, \sum_{i=1}^{n} p_{m,i}\, x_i \right) \\ \text{s. t.} \quad & x_i \in \{0,1\}, \qquad i = 1, \ldots, n. \end{aligned} \tag{MUCO}$$

It is a multi-objective binary optimization problem with m sum objective functions $f_j(x) = \sum_{i=1}^{n} p_{j,i}\, x_i$, for $j = 1 \ldots, m$, but without any further constraint. MUCO can be interpreted as a relaxation of the well known *multi-objective* $\{0,1\}$ *knapsack problem* (MOKP)

$$\begin{aligned} \max \quad & f(x) = \left(\sum_{i=1}^{n} p_{1,i}\, x_i, \ldots, \sum_{i=1}^{n} p_{m,i}\, x_i \right) \\ \text{s. t.} \quad & \sum_{i=1}^{n} w_i\, x_i \leq W \\ & x_i \in \{0,1\}, \qquad i = 1, \ldots, n, \end{aligned} \tag{MOKP}$$

with integral coefficients $w_i > 0$, $i = 1, \ldots, n$, and budget $W > 0$, from which it can be obtained by relaxing the capacity constraint. Due to the close relation between MUCO and MOKP, the objective function coefficients are denoted by $p_{j,i}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$ and call $p_{j,i}$ *profit* of item i in objective function j. Throughout the article, it is assumed that all profits $p_{j,i}$ have integer values for $i = 1, \ldots, n$ and for $j = 1, \ldots, m$. Furthermore, the vector of profits $(p_{1,i}, \ldots, p_{m,i}) \in \mathbb{Z}^m$ of item i is denoted by $p_{\cdot,i}$ and it is assumed that $p_{\cdot,i} \neq 0_m = (0, \ldots, 0)^\top \in \mathbb{Z}^m$ for all items $i = 1, \ldots, n$.

For knapsack problems, the profit coefficients are, in general, assumed to be non-negative. For MUCO this assumption is not reasonable. Having only non-negative profit values, the maximization objectives tend to select all items. It is assumed that the objective functions are conflicting and, thus, negative coefficients are explicitly allowed.

For a survey of the computational complexity of multi-objective optimization problems in general, and of the multi-objective knapsack problem in particular, it is referred to [14]. The difficulty of many multi-objective combinatorial optimization problems has two reasons: (1) intractability, i. e., in the worst case the size of the nondominated set grows exponentially with the problem size, and (2) the $\mathcal{NP}$-hardness of scalarizations that can be used to compute unsupported solutions. Ehrgott [10] proves that MUCO is intractable and that the corresponding decision problem is $\mathcal{NP}$-hard. If the nondominated set grows exponentially with the problem size, then a polynomial time algorithm for the exact computation of the complete nondominated set is impossible in general. Bökler et al [7] suggest a so-called output sensitive complexity measure, that relates the computational complexity to the encoding length not only of the input, but also of the output. Note that in the case that the output is of polynomial size this differentiation is not necessary.

Despite the fact that MUCO is hard to solve in general, the *single-objective unconstrained combinatorial optimization problem*

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} p_{1,i}\, x_i \\ \text{s.t.} \quad & x_i \in \{0,1\}, \qquad i = 1,\dots,n \end{aligned} \tag{1}$$

is very easy. In this case, the set of all optimal solutions can be given explicitly: All solutions x with

$$x_i \begin{cases} = 0 & \text{if } p_{1,i} < 0 \\ \in \{0,1\} & \text{if } p_{1,i} = 0 \\ = 1 & \text{if } p_{1,i} > 0 \end{cases} \tag{2}$$

for $i = 1,\dots,n$, are optimal for (1), and there are no further optimal solutions.

One important observation is in this case that the decision on one item can be made independently of the decisions on all other items. Since there is no constraint, the variables are not interlinked. Certainly, the multi-objective version MUCO is more complicated, but this independence of decisions is preserved. As long as all coefficients of an item $i \in \{1,\dots,n\}$ have equal signs for all objective functions, equation (2) can still be applied. However, in general the objective functions are conflicting and, therefore, the signs of the coefficients differ.

Before proceeding some basic definitions are given to clarify the used notation. For an introduction to multi-objective optimization in general and multi-objective combinatorial optimization in particular it is referred to [10, 11]. Throughout the article the concept of Pareto optimality is used. Let a multi-objective combinatorial optimization problem (MOCO) be given as

$$\max_{x\in\mathcal{X}} \; f(x) = (f_1(x),\dots,f_m(x))^\top. \tag{MOCO}$$

Here, $\mathcal{X}$ denotes the *set of feasible solutions* which is assumed to be discrete and finite. The image of $\mathcal{X}$ in the objective space is called the *set of feasible points* and is denoted by $\mathcal{Y} := f(\mathcal{X})$. A solution $x \in \mathcal{X}$ is called *efficient (or Pareto optimal)* if there is no other solution $\bar{x} \in \mathcal{X}$ such that

$$f_j(x) \le f_j(\bar{x}) \;\text{ for all } j = 1,\dots,m \quad \text{with } f(x) \ne f(\bar{x}).$$

The corresponding point $f(x)$ is called *nondominated* in this case. Let $x, \bar{x} \in \mathcal{X}$. If $f_j(\bar{x}) \le f_j(x)$ for all $j = 1,\dots,m$ and $f(\bar{x}) \ne f(x)$, then solution x *dominates* solution $\bar{x}$ and point $f(x)$ *dominates* point $f(\bar{x})$. The *set of efficient solutions* is denoted by $\mathcal{X}_E \subseteq \mathcal{X}$ and the *set of nondominated points* by $\mathcal{Y}_N \subseteq \mathcal{Y}$.

A solution $x \in \mathcal{X}$ is called *weakly efficient* if there is no other solution $\bar{x} \in \mathcal{X}$ such that

$$f_j(x) < f_j(\bar{x}) \;\text{ for all } j = 1,\dots,m.$$

The Minkovski-sum of sets $\mathcal{A}$ and $\mathcal{B}$ in $\mathbb{R}^m$ is defined as $\mathcal{A} + \mathcal{B} = \{a + b \colon a \in \mathcal{A},\ b \in \mathcal{B}\}$. Let $\mathbb{R}^m_{\geqq} := \{y \in \mathbb{R}^m \colon y_j \geq 0 \ \forall j = 1, \ldots, m\}$ denote the non-negative orthant of $\mathbb{R}^m$ and let $\operatorname{conv}(\mathcal{Y}_N)$ denote the convex hull of the nondominated set. The nondominated set of $\operatorname{conv}(\mathcal{Y}_N)$, that is $\big\{y \in \operatorname{conv}(\mathcal{Y}_N) \colon \operatorname{conv}(\mathcal{Y}_N) \cap (\{y\} + \mathbb{R}^m_{\geqq}) = \{y\}\big\}$, is called the *nondominated frontier* [cf. 10].

The nondominated points of MOCO can be classified into two categories: The set of *supported* points $\mathcal{Y}_{sN}$ is defined as $\mathcal{Y}_{sN} = \mathcal{Y}_N \cap \operatorname{conv}(\mathcal{Y}_N)$, i. e., all points $y \in \mathcal{Y}_{sN}$ are nondominated for MOCO *and* are located on the nondominated frontier (and hence on $\operatorname{conv}(\mathcal{Y}_N)$). The set of *unsupported* points $\mathcal{Y}_{uN}$ is given by $\mathcal{Y}_{uN} = \mathcal{Y}_N \setminus \mathcal{Y}_{sN}$. The set of supported points can be further partitioned into those nondominated points that are extreme points of $\operatorname{conv}(\mathcal{Y}_N)$, and those that are nonextreme (i.e., points that can be found as a convex combination of other points of $\operatorname{conv}(\mathcal{Y}_N)$). The set of *extreme supported* points is denoted by $\mathcal{Y}_{eN}$, and the set of *nonextreme supported* points is denoted by $\mathcal{Y}_{nN}$. The corresponding sets of *supported* solutions, *unsupported* solutions, *extreme supported* solutions, and *nonextreme supported* solutions are denoted by $\mathcal{X}_{sE}$, $\mathcal{X}_{uE}$, $\mathcal{X}_{eE}$, and $\mathcal{X}_{nE}$, respectively.

From an algorithmic perspective, the extreme supported points play a central role in multi-objective combinatorial optimization: They can be computed using weighted sum scalarizations (as can all supported points as well), they provide information on achievable ranges of objective values and, in this way, support the decision making process. Furthermore, they are "maximal" in the sense that they lie on the convex hull of feasible points and, thus, define an upper bound set of $\mathcal{Y}_N$.

The computation of the set of supported points in a first phase followed by the computation of all efficient points using the precomputed information in a second phase is a successful concept for MOCO problems. Visée et al [29] were motivated by the results of numerical experiments on the bi-objective knapsack problem to introduce the *two phase method*. The authors observed that the number of supported points usually grows only linearly with the number of items whereas the number of unsupported points grows exponentially. However, there are examples with an exponential number of supported points, see, for example, [10, 28]. It is an open question whether the number of extreme supported points of MOCO problems is in general polynomially bounded. Seipp [27] presents a polynomial bound on the number of extreme supported points for multi-objective minimum spanning tree problems. The author proves his result using arrangements of hyperplanes, a concept that is also used in this paper. Aissi et al [1] study the number of supported cuts in graphs and hypergraphs with multiple edge cost functions and prove a polynomial bound with respect to the number of nodes and edges. Their result is based on bounds on the number of approximate global minimum cuts and not on arrangements of hyperplanes.

In this paper, a polynomial bound on the number of extreme supported solutions for MUCO is proven using concepts from combinatorial geometry, namely *arrangements of hyperplanes* and *zonotopes*. Furthermore, conditions for the existence of nonextreme supported solutions are presented. This is achieved by deriving correspondences between solutions of MUCO, weight space decomposition, zonotopes, and arrangement of hyperplanes. Furthermore, the results on MUCO are used to improve existing algorithms for computing the set of extreme supported points of a general version of MOKP, i. e., assuming that the profit coefficients can be positive or negative integers.

This article is structured as follows: In Section 2 basic definitions and results on zonotopes, weight space decomposition and arrangements of hyperplanes are given. The interrelations between MUCO and these concepts from combinatorial geometry are presented in Section 3. Based on these interrelations a polynomial bound on the number of extreme supported solutions of MUCO is proven. In Section 4, these interrelations are used to develop an algorithm to compute all supported points. In a case study this approach is applied on tri-objective unconstrained optimization problems (TUCO) and computational results are shown. Since nondominated solutions of MUCO are also optimal for the corresponding MOKPs as long as they are feasible for it, the presented approach can be incorporated into existing solution algorithms for MOKP, which is described in Section 5. Computational results are again presented for a case study on tri-objective knapsack problems (TOKP). Section 6 summarizes and concludes this article.

2 Definitions and basic properties

In this section, some concepts from combinatorial geometry and multi-objective optimization are reviewed, that will be used in later sections of this paper. For further reading on the following definitions and properties on polyhedra, zonotopes and arrangements of hyperplanes it is referred to the book of Edelsbrunner [9]. Further details on multi-objective optimization and on the weight space decomposition can be found, for example, in Ehrgott [10] and Przybylski et al [23].

2.1 Polyhedra and zonotopes

A set $\mathcal{P} \in \mathbb{R}^m$ is called a *polyhedron* if it is the intersection of finitely many halfspaces. If $\mathcal{P}$ is additionally bounded, it is called a *polytope*. It can be shown that the convex hull $\operatorname{conv}(\mathcal{A})$ of any finite set of points $\mathcal{A}$ in $\mathbb{R}^m$ is a polytope. Let k be the maximal number of affinely independent points in $\mathcal{P}$. The dimension $\dim(\mathcal{P})$ of $\mathcal{P}$ is defined as $\dim(\mathcal{P}) = k - 1$.

For a polyhedron $\mathcal{P} \subseteq \mathbb{R}^m$, an inequality $\lambda\, y \leq \lambda_0$ with $\lambda \in \mathbb{R}^m, \lambda_0 \in \mathbb{R}$ is called a *valid inequality* if it is satisfied for all $y \in \mathcal{P}$. A subset $\varphi \subseteq \mathcal{P}$ is called a *face* of $\mathcal{P}$ if $\varphi = \{y \in \mathcal{P} \colon \lambda\, y = \lambda_0\}$ for some valid inequality $\lambda\, y \leq \lambda_0$ of $\mathcal{P}$. It can be shown that a face of a polyhedron is again a polyhedron. A face is called k-face if it is a polyhedron of dimension k. $(m-1)$-faces are called facets [cf. 30].

For all binary optimization problems, particularly for MUCO and MOKP, the convex hull $\operatorname{conv}(\mathcal{Y})$ of the set of feasible points is a polytope since all variables are binary and, therefore, the feasible points in the objective space are bounded by the sum of all objective function coefficients. Faces of $\operatorname{conv}(\mathcal{Y})$ are called nondominated if they are part of the nondominated frontier.

A set $\mathcal{Z} \subset \mathbb{R}^m$ is called a *zonotope* if it is the Minkovski-sum of a finite number of closed line segments $[u_i, v_i] = \{y \in \mathbb{R}^m \colon y = u_i + \mu\,(v_i - u_i), \mu \in [0,1]\}$, with vectors u_i, $v_i \in \mathbb{R}^m$, $u_i \neq v_i$, for $i = 1, \ldots, n$. Zonotopes are polytopes.

The center of a zonotope can be generated by summing over the midpoints $\bar{y}_i$ of each line segment, where $\bar{y}_i = u_i + \frac{1}{2}\,(v_i - u_i)$, $i = 1, \ldots, n$. Zonotopes are centrally symmetric, which can easily be seen by translating each line segment such that the origin is its midpoint [cf. 9].

2.2 Weighted sum scalarization and weight space decomposition

Gaas and Saaty [15] introduced the weighted sum scalarization for linear programming problems with two objectives. Applied to MOCO problems, this implies the "parametric function"

$$\max_{x\in\mathcal{X}} \sum_{j=1}^{m} \lambda_j\, f_j(x) = \langle \lambda, f(x)\rangle \tag{WS(λ)}$$

where the *weights* λ are in $\mathbb{R}^m$. It is well-known [16] that for $\lambda \in \mathbb{R}^m_{\geq} := \{\lambda \in \mathbb{R}^m \colon \lambda_j \geq 0,\ j = 1,\dots,m,\ \lambda \neq 0_m\}$ every optimal solution of WS(λ) is a weakly efficient solution of MOCO. It holds that for $\lambda \in \mathbb{R}^m_{>} := \{\lambda \in \mathbb{R}^m \colon \lambda_j > 0,\ j = 1,\dots,m\}$ every optimal solution of WS(λ) is a supported solution of MOCO. In this context, the set $\mathbb{R}^m_{>}$ is called the *weight space*. Since the set of feasible points $\mathcal{Y}$ of MOCO is discrete and finite, every supported solution of the initial problem can be computed as an optimal solution of WS(λ) using appropriate weights $\lambda \in \mathbb{R}^m_{>}$. Since multiples of a weight vector lead to the same optimal solutions, the *normalized weight space* $\widetilde{\mathcal{W}}$ is defined as the simplex

$$\widetilde{\mathcal{W}} := \left\{\lambda \in \mathbb{R}^m_{>} \colon \sum_{j=1}^{m} \lambda_j = 1\right\}.$$

A projection of the normalized weight space on $\mathbb{R}^{m-1}_{>}$ is done by setting $\lambda_1 := 1 - \sum_{j=2}^{m} \lambda_j$ and, referring to this, the *projected weight space* $\mathcal{W}$ is defined as:

$$\mathcal{W} := \left\{(\lambda_2,\dots,\lambda_m) \in \mathbb{R}^{m-1}_{>} \colon \sum_{j=2}^{m} \lambda_j < 1\right\}.$$

As introduced in the beginning of this section, the weighted sum scalarization can be applied to find the extreme supported solutions of a MOCO problem. Benson and Sun [3] show for multi-objective linear programming problems that there exists a correspondence between supported solutions and subsets of the projected weight space $\mathcal{W}$. The weight space decomposition identifies the weight vectors that lead to each supported solution, or efficient basic solution, respectively. Przybylski et al [23] analyze this interrelation for multi-objective integer programming problems with m objective functions: For a supported point $y \in \mathcal{Y}_{sN}$, it is defined

$$\mathcal{W}(y) := \left\{(\lambda_2,\dots,\lambda_m) \in \mathcal{W} \colon \left(1 - \sum_{j=2}^{m} \lambda_j, \lambda_2,\dots,\lambda_m\right) \in \widetilde{\mathcal{W}}(y)\right\}$$
$$\text{with}\quad \widetilde{\mathcal{W}}(y) := \left\{\lambda \in \widetilde{\mathcal{W}} \colon \langle\lambda, y\rangle = \min\{\langle\lambda,\bar{y}\rangle \colon \bar{y} \in \mathcal{Y}_{eN}\}\right\}$$

which is the subset of weights λ in the projected weight space $\mathcal{W}$ that define weighted sum problems for which y is optimal. Note that a weight λ can be contained in several sets $\mathcal{W}(y)$.

The set $\mathcal{W}(y)$ is a polytope, and $y \in \mathcal{Y}$ is an extreme supported point if and only if $\dim(\mathcal{W}(y)) = m - 1$ [23]. Furthermore, for two supported points y and $\bar{y}$ either the intersection of $\mathcal{W}(y)$ and $\mathcal{W}(\bar{y})$ is empty or they intersect in a common face. The sets $\mathcal{W}(y)$, for $y \in \mathcal{Y}_{eN}$, define a decomposition of $\mathcal{W}$, i. e.,

$$\mathcal{W} = \bigcup_{y \in \mathcal{Y}_{eN}} \mathcal{W}(y).$$

Note that all definitions and statements can be formulated analogously for the weight space $\mathbb{R}^m_>$ and for the normalized weight space $\widetilde{\mathcal{W}}$.

2.3 Arrangements of hyperplanes

Let $p \in \mathbb{R}^m$, $p \neq 0_m$, and $c \in \mathbb{R}$ be given. A *hyperplane* h in $\mathbb{R}^m$ is defined as

$$h(p, c) := \{\lambda \in \mathbb{R}^m : \langle p, \lambda \rangle = c\}.$$

Given a finite set of hyperplanes $H = \{h_1(p_1, c_1), \ldots, h_n(p_n, c_n)\}$, the hyperplanes subdivide $\mathbb{R}^m$ into a set of convex polyhedra of different dimensions. This is called the *arrangement of hyperplanes* [9].

Every hyperplane $h_i = h_i(p_i, c_i)$, $i = 1, \ldots, n$, subdivides $\mathbb{R}^m$ into two open half-spaces h_i^+ and h_i^- given by

$$h_i^+ := \{\lambda \in \mathbb{R}^m : \langle p_i, \lambda \rangle > c_i\} \quad \text{and} \quad h_i^- := \{\lambda \in \mathbb{R}^m : \langle p_i, \lambda \rangle < c_i\}.$$

For a point λ in $\mathbb{R}^m$ the *position vector of* λ is defined as

$$\mathrm{Pos}(\lambda) = (\mathrm{Pos}_1(\lambda), \ldots, \mathrm{Pos}_n(\lambda))$$

with

$$\mathrm{Pos}_i(\lambda) = \begin{cases} -1 & \text{if } \lambda \in h_i^- \\ 0 & \text{if } \lambda \in h_i \\ +1 & \text{if } \lambda \in h_i^+ \end{cases}$$

for $i = 1, \ldots, n$. Two points are called equivalent if their position vectors are equal. This defines an equivalence relation on $\mathbb{R}^m$ where the equivalence classes are called *faces* φ of the arrangement of hyperplanes. Note that the arrangement of hyperplanes defines a partition of $\mathbb{R}^m$ and that each point λ is contained in exactly one face. The position vector $\mathrm{Pos}(\varphi)$ of a face φ is set to $\mathrm{Pos}(\varphi) = \mathrm{Pos}(\lambda)$, for an arbitrary point λ in φ. A face of dimension k is called a *k-face* denoted by $\varphi^{(k)}$. Furthermore, a 0-face is called a *vertex*, a 1-face is called an *edge*, an $(m-1)$-face is called a *facet*, and an m-face is called a *cell*.

Let $\varphi_1^{(k)}$ and $\varphi_2^{(k-1)}$ be faces of an arrangement of hyperplanes with dimensions k and $k-1$, respectively, where $1 \leq k \leq m$. If $\varphi_2^{(k-1)}$ is contained in the boundary of $\varphi_1^{(k)}$, then $\varphi_2^{(k-1)}$ is called a *subface of* $\varphi_1^{(k)}$. If $\varphi_2^{(k-1)}$ is a subface of $\varphi_1^{(k)}$, consequently, the position vectors $\mathrm{Pos}(\varphi_1^{(k)})$ and $\mathrm{Pos}(\varphi_2^{(k-1)})$ differ in positions $i \in J \subseteq \{1, \ldots, n\}$, with $|J| \geq 1$, and the corresponding entries in the position vector of $\varphi_2^{(k-1)}$ are 0, i. e., $\mathrm{Pos}_i(\varphi_1^{(k)}) = \mathrm{Pos}_i(\varphi_2^{(k-1)})$ for $i \in \{1, \ldots, n\} \setminus J$ and $\mathrm{Pos}_i(\varphi_2^{(k-1)}) = 0$ and $\mathrm{Pos}_i(\varphi_1^{(k)}) \neq 0$ for $i \in J$. A pair of faces $\varphi_0^{(\ell)}$ and $\varphi_{k-\ell}^{(k)}$, with

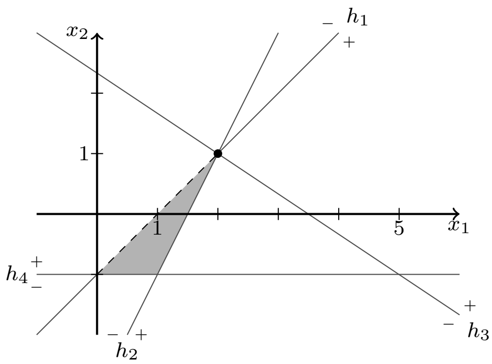

Fig. 1 Arrangement of hyperplanes of Example 1.

$0 \leq \ell < k \leq m$, is called *adjacent*, if there exists a set of faces $\{\varphi_1^{(\ell+1)}, \dots, \varphi_{k-\ell-1}^{(k-1)}\}$ such that $\varphi_s^{(\ell+s)}$ is a subface of $\varphi_{s+1}^{(\ell+s+1)}$, for $0 \leq s < k-\ell-1$. This implies that $\varphi_0^{(\ell)}$ is part of the closure of $\varphi_{k-\ell}^{(k)}$.

Example 1 Consider the following four hyperplanes in $\mathbb{R}^2$:

$$h_1\Big(\big(\begin{smallmatrix}1\\-1\end{smallmatrix}\big), 1\Big), \quad h_2\Big(\big(\begin{smallmatrix}2\\-1\end{smallmatrix}\big), 3\Big), \quad h_3\Big(\big(\begin{smallmatrix}2\\3\end{smallmatrix}\big), 7\Big), \quad \text{and } h_4\Big(\big(\begin{smallmatrix}0\\1\end{smallmatrix}\big), -1\Big).$$

They define an arrangement of hyperplanes in $\mathbb{R}^2$ with 10 cells (2-faces), 13 facets/edges (1-faces), and four vertices (0-faces), see Figure 1 for an illustration of the half-spaces h_i^- and h_i^+, for $i = 1, \dots, 4$. The highlighted vertex ($\bullet$) has the vector $(0, 0, 0, +1)^\top$ as position vector and is a subface of six facets. The highlighted facet (dashed line) has the vector $(0, -1, -1, +1)^\top$ as position vector and is a subface of two cells. The highlighted cell (shaded area) has the vector $(+1, -1, -1, +1)^\top$ as position vector. The highlighted vertex and cell are adjacent.

An arrangement of n hyperplanes in $\mathbb{R}^m$ with $m \leq n$ is called *simple* if the intersection of any subset of m hyperplanes is a unique point and if the intersection of any subset of $(m+1)$ hyperplanes is empty. This implies that the position vectors of $\varphi_1^{(k)}$ and any of its subfaces $\varphi_2^{(k-1)}$ differ in exactly one position. The arrangement of Example 1 is not simple, since the three hyperplanes h_1, h_2 and h_3 intersect in one point in $\mathbb{R}^2$.

In the following, the number of cells of an arrangement of hyperplanes in $\mathbb{R}^m$ plays an important role.

Theorem 1 (Buck [8]) *For simple arrangements of hyperplanes in* $\mathbb{R}^m$*, the number of cells is equal to*

$$\sum_{i=0}^{m} \binom{n}{i}.$$

This number is an upper bound for non-simple arrangements. In general, the number of k-faces is bounded by $\mathcal{O}(n^m)$ for each k, $0 \leq k \leq m$.

An arrangement of n hyperplanes in $\mathbb{R}^m$ with $m \leq n$ is called *central*, if 0_m is contained in every hyperplane. Trivially, unless n is equal to m, central arrangements are not simple and simple arrangements cannot be central.

Theorem 2 (Zaslavsky [31]) *For central arrangements of hyperplanes in $\mathbb{R}^m$, the number of cells is bounded by*

$$2 \sum_{i=0}^{m-1} \binom{n-1}{i}.$$

Hence, fixing one central intersection point for all hyperplanes reduces the number of cells to $\mathcal{O}(n^{m-1})$.

Edelsbrunner [9] presents an algorithm to compute a graph-based representation of the whole structure of an arrangement of hyperplanes, i. e., for representing all faces and all relations between the faces. The algorithm runs in $\mathcal{O}(n^m)$ time, which is asymptotically optimal since the number of faces is also in $\mathcal{O}(n^m)$. The space complexity is as large as the output size. Ferrez et al [13] present a reverse search algorithm that identifies all cells of a central arrangement of hyperplanes. The authors take advantage of the centrality of the arrangement to reduce the dimension by one and, hence, work with a general arrangement in $\mathbb{R}^{m-1}$. The algorithm has a time complexity of $\mathcal{O}(n\, c\, \mathrm{LP}(n, m))$, where c is the number of cells of the arrangement, which is bounded by $\mathcal{O}(n^{m-1})$; $\mathrm{LP}(n, m)$ denotes the complexity for solving a linear program with n inequalities and m variables, which can be done in polynomial time with interior-point methods [see, e. g., 18]. This is, in fact, a weaker bound than for the approach by Edelsbrunner [9]. However, the space complexity of their algorithm is in $\mathcal{O}(n\, m)$, improving the bound of the approach by Edelsbrunner. Moreover, the authors argue that their algorithm is easier to implement.

3 A polynomial bound on the number of extreme supported solutions

In the following, it is shown that MUCO, the weight space, zonotopes and arrangements of hyperplanes are closely related. In Example 2 an instance of MUCO is introduced, which is used for illustration purposes throughout this section.

Example 2 Consider the folowing instance of MUCO:

$$\begin{array}{lll} \max & -x_2 + 3\,x_3 + 6\,x_4 - 5\,x_5 + x_6 & \\ \max & x_1 + 2\,x_2 - 3\,x_3 - 2\,x_4 - \;\; x_5 + x_6 & \\ \text{s.t.} & x_i \in \{0,1\}, \qquad i = 1, \ldots, 6. & \end{array}$$

This problem has $2^6 = 64$ feasible solutions, including 11 weakly efficient solutions, out of which five are efficient solutions, out of which four are extreme supported solutions. The corresponding extreme supported points are $(0,4)^\top$, $(6,2)^\top$, $(9,-1)^\top$, and $(10,-3)^\top$. Additionally, there is only one unsupported point at $(7,0)^\top$, see also the left part of Figure 2.

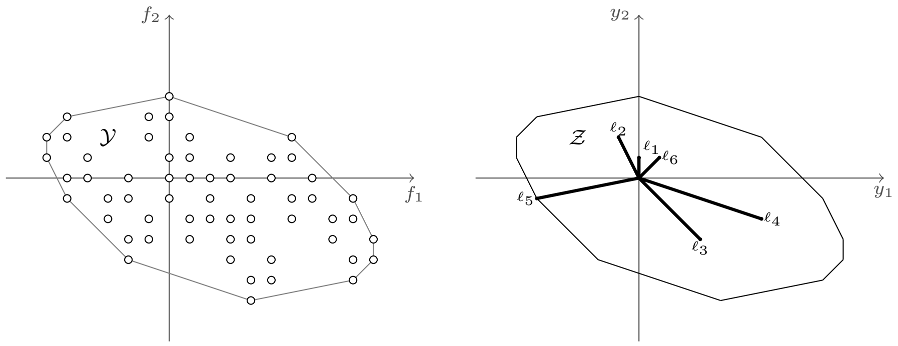

Fig. 2 On the left: set of feasible points $\mathcal{Y}$ and its convex hull for Example 2. On the right: associated zonotope $\mathcal{Z}$ with line segments ℓ_1 to ℓ_6 for Example 3.

3.1 MUCO and zonotopes

Zonotopes and MUCO are related to each other. The LP-relaxation of MUCO provides a link between both concepts. Let $\mathcal{Y}^{\text{LP}}$ be the set of feasible points of the LP-relaxation of MUCO. It is shown that $\mathcal{Y}^{\text{LP}} = \text{conv}(\mathcal{Y})$. To see this, let $\{x^1, \ldots, x^{2^n}\} = \mathcal{X} = \{0,1\}^n$ denote the set of all feasible solutions of MUCO and recall that $p_{\cdot,i} = (p_{1,i}, \ldots, p_{m,i})^\top \in \mathbb{Z}^m$ is the vector of profits of item i, for $i = 1, \ldots, n$. Then

$$\begin{aligned}
\text{conv}(\mathcal{Y}) &= \left\{ \sum_{k=1}^{2^n} \mu_k \sum_{i=1}^{n} p_{\cdot,i}\, x_i^k \colon \sum_{k=1}^{2^n} \mu_k = 1,\ \mu_k \geq 0\ \forall k \in \{1, \ldots, 2^n\} \right\} \\
&= \left\{ \sum_{i=1}^{n} p_{\cdot,i} \sum_{k=1}^{2^n} \mu_k\, x_i^k \colon \sum_{k=1}^{2^n} \mu_k = 1,\ \mu^k \geq 0\ \forall k \in \{1, \ldots, 2^n\} \right\} \\
&\overset{(*)}{=} \left\{ \sum_{i=1}^{n} p_{\cdot,i}\, \hat{x}_i \colon \hat{x}_i \in [0,1]\ \forall i \in \{1, \ldots, n\} \right\} = \mathcal{Y}^{\text{LP}}.
\end{aligned}$$

The equality in $(*)$ holds since $[0,1]^n$ is an integral polytope, the extreme points of which are the feasible solutions $x \in \mathcal{X}$ of MUCO.

For a given instance of MUCO, an *associated zonotope* can now be defined: For each item $i \in \{1, \ldots, n\}$ of MUCO a line segment $[0, p_{\cdot,i}]$, using the corresponding profit vector $p_{\cdot,i}$ of item i, is defined as $[0, p_{\cdot,i}] = \{y_i \in \mathbb{R}^m \colon y_i = \mu\, p_{\cdot,i},\ \mu \in [0,1]\}$. The zonotope $\mathcal{Z}$ defined by these line segments is equal to the convex hull $\text{conv}(\mathcal{Y})$ of the set of feasible points of MUCO:

$$\begin{aligned}
\text{conv}(\mathcal{Y}) = \mathcal{Y}^{\text{LP}} &= \left\{ \sum_{i=1}^{n} p_{\cdot,i}\, \hat{x}_i \colon \hat{x}_i \in [0,1]\ \forall i \in \{1, \ldots, n\} \right\} \\
&= \left\{ \sum_{i=1}^{n} y_i \colon y_i \in [0, p_{\cdot,i}]\ \forall i \in \{1, \ldots, n\} \right\} = \mathcal{Z}.
\end{aligned}$$

In particular, the extreme points of $\mathcal{Z}$ and of $\operatorname{conv}(\mathcal{Y})$ are equal.

Example 3 The zonotope in $\mathbb{R}^2$ defined by the line segments

$$\begin{aligned}\ell_1 &= \left[\binom{0}{0}, \binom{0}{1}\right], \quad \ell_2 = \left[\binom{0}{0}, \binom{-1}{2}\right], \ell_3 = \left[\binom{0}{0}, \binom{3}{-3}\right],\\ \ell_4 &= \left[\binom{0}{0}, \binom{6}{-2}\right], \ell_5 = \left[\binom{0}{0}, \binom{-5}{-1}\right], \ell_6 = \left[\binom{0}{0}, \binom{1}{1}\right]\end{aligned}$$

is equal to the convex hull of the set of feasible points of Example 2, see Figure 2.

Conversely, for a given zonotope $\mathcal{Z}$ an associated instance of MUCO can be defined: For each defining line segment $[u_i, v_i]$, $i \in \{1, \ldots, n\}$, a profit vector $p_{\cdot,i} := v_i - u_i$ for item i, for $i = 1, \ldots, n$, is defined. Additionally, each objective function has a constant term $\sum_{i=1}^{n} u_i^j$, for $j = 1, \ldots, m$. The convex hull of the set of feasible points $\operatorname{conv}(\mathcal{Y})$ of this instance, as well as the corresponding set of feasible points $\mathcal{Y}^{\mathrm{LP}}$ of the LP-relaxation, is equal to the zonotope $\mathcal{Z}$:

$$\begin{aligned}\mathcal{Z} &= \left\{\sum_{i=1}^{n} y_i \colon y_i \in [u_i, v_i] \; \forall i \in \{1, \ldots, n\}\right\}\\ &= \left\{\sum_{i=1}^{n} y_i \colon y_i = u_i + \mu_i(v_i - u_i),\; \mu_i \in [0,1] \; \forall i \in \{1, \ldots, n\}\right\}\\ &= \left\{\sum_{i=1}^{n} u_i + \sum_{i=1}^{n} \mu_i(v_i - u_i) \colon \mu_i \in [0,1] \; \forall i \in \{1, \ldots, n\}\right\}\\ &= \left\{\sum_{i=1}^{n} u_i + \sum_{i=1}^{n} \mu_i \, p_{\cdot,i} \colon \mu_i \in [0,1] \; \forall i \in \{1, \ldots, n\}\right\}\\ &= \mathcal{Y}^{\mathrm{LP}} = \operatorname{conv}(\mathcal{Y}).\end{aligned}$$

In the following, an extreme point of a zonotope is called nondominated if the corresponding point of the associated problem MUCO is nondominated. Note that the corresponding nondominated points of MUCO in the objective space are extreme supported points since they are extreme points of the nondominated frontier.

3.2 Arrangements of hyperplanes and zonotopes

It is well known [see e. g., 9], that for every zonotope it is possible to generate an associated arrangement of hyperplanes, and vice versa, such that there is a one-to-one correspondence between the respective faces. Let a zonotope $\mathcal{Z} \subset \mathbb{R}^m$ be defined by n line segments $[0, p_{\cdot,i}]$ with $p_{\cdot,i} \in \mathbb{Z}^m \setminus \{0\}$, for $i = 1, \ldots, n$. An *associated arrangement of hyperplanes* can be defined by the hyperplanes

$$h_i = \{\lambda \in \mathbb{R}^m \colon \langle p_{\cdot,i}, \lambda \rangle = 0\}$$

and the corresponding half-spaces

$$h_i^+ = \{\lambda \in \mathbb{R}^m : \langle p_{\cdot,i}, \lambda \rangle > 0\}$$
$$h_i^- = \{\lambda \in \mathbb{R}^m : \langle p_{\cdot,i}, \lambda \rangle < 0\}$$

for $i = 1, \ldots, n$. This arrangement is central (assuming $m \leq n$,), since $0_m \in h_i$ for all $i = 1, \ldots, n$.

Example 4 The zonotope of Example 3 corresponds to the arrangement of hyperplanes $H = \{h_1, \ldots, h_6\}$ with

$$h_1 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{0}{1}, \lambda \rangle = 0\right\}, \qquad h_2 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{-1}{2}, \lambda \rangle = 0\right\},$$

$$h_3 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{3}{-3}, \lambda \rangle = 0\right\}, \qquad h_4 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{6}{-2}, \lambda \rangle = 0\right\},$$

$$h_5 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{-5}{-1}, \lambda \rangle = 0\right\}, \qquad h_6 = \left\{\lambda \in \mathbb{R}^2 : \langle \tbinom{1}{1}, \lambda \rangle = 0\right\},$$

which is central, see the right part of Figure 3.

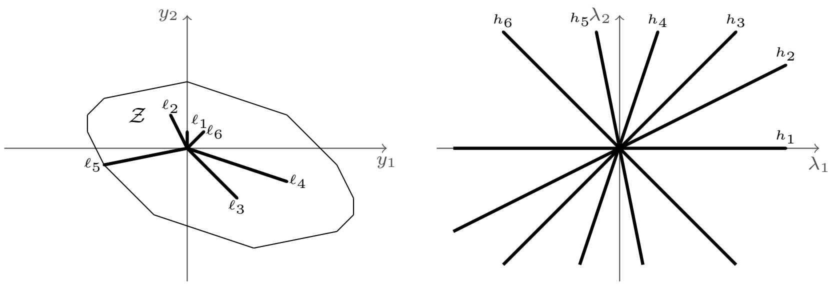

Fig. 3 On the left: zonotope $\mathcal{Z}$ with line segments ℓ_1 to ℓ_6 (cf. Example 3). On the right: associated arrangement of hyperplanes $H(h_1, \ldots, h_6)$ (cf. Example 4).

3.3 MUCO, weight space decomposition and arrangements of hyperplanes

As mentioned above, extreme supported points of MUCO can be computed using the weighted sum scalarization, where the weights are in $\mathbb{R}^m_>$. The objective function of the weighted sum problem WS(λ) can be reorganized as follows:

$$\sum_{j=1}^{m} \lambda_j f_j(x) = \sum_{j=1}^{m} \lambda_j \left(\sum_{i=1}^{n} p_{j,i}\, x_i\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{m} \lambda_j\, p_{j,i}\right) x_i = \sum_{i=1}^{n} \langle p_{\cdot,i}, \lambda \rangle\, x_i.$$

The coefficients $p_{\cdot,i}$, for $i = 1, \ldots, n$, define an arrangement of hyperplanes in $\mathbb{R}^m_>$ where $h_i := h(p_{\cdot,i}, 0)$. Recall that the optimal solution of a single-objective unconstrained combinatorial optimization problem (1) can be built by deciding on each variable independently, depending on the sign of the coefficient. Since the weighted sum problem WS(λ) is equivalent to (1), the position of the weight λ

in $\mathbb{R}^m_>$ defines the optimal choice for x_i, for $i = 1, \dots, n$: If λ is in h_i^+, then $x_i = 1$ is optimal, if λ is in h_i^-, then $x_i = 0$ is optimal, and if λ is in h_i, then both alternatives are optimal. This last case indicates that, as is known from the weighted sum scalarization, one weight λ can correspond to several nondominated points that define a face of the nondominated frontier. All weights $\lambda \in \mathbb{R}^m_>$ with equal position vectors correspond to the same set of nondominated points.

This shows that the correspondence between MUCO (zonotopes) and the arrangement of hyperplanes has an order reversing characteristic: A nondominated k-face of the convex hull of feasible points in $\mathbb{R}^m$ corresponds to an $(m-k)$-face of the associated arrangement of hyperplanes, in $\mathbb{R}^m_>$. Thus, extreme supported points of MUCO, i. e., 0-faces of $\operatorname{conv}(\mathcal{Y})$ correspond to cells, i. e., m-faces of the associated arrangement and vice versa. Since the number of cells in the arrangement is bounded, see [31] (Theorem 2 above), the same bound holds for the number of extreme supported points of MUCO. The above findings prove the following theorem:

Theorem 3 *The number of extreme supported solutions of MUCO with m objective functions and n items is bounded by:*

$$|\mathcal{X}_{eE}| \leq 2 \sum_{i=0}^{m-1} \binom{n-1}{i},$$

i. e., for fixed m, MUCO has at most $\mathcal{O}(n^{m-1})$ extreme supported solutions.

Furthermore, it is known that a cell of the arrangement is either in h_i^+ or in h_i^-, for all $i = 1, \dots, n$. Either $x_i = 0$ or $x_i = 1$ is optimal in the corresponding solution, but not both alternatives. Thus, the cell corresponds to one unique solution of MUCO.

Corollary 1 *Every extreme supported point of MUCO is realized by exactly one extreme supported solution.*

The arrangement of hyperplanes h_i, for all $i = 1, \dots, n$, can be used to define the decomposition of the weight space $\mathbb{R}^m_>$ for MUCO. Given an extreme supported solution x and the corresponding point $y = f(x)$, the set $\mathcal{W}(y)$ consists of all faces of the arrangement of hyperplanes that correspond to the solution x. These faces are the cell of the arrangement corresponding to x and all adjacent faces in its boundary. The term "weight space decomposition" is also used for the arrangement of hyperplanes in the remainder of this article since the arrangement is in fact a decomposition of the weight space and the weight space decomposition defined in Section 2.2 can be determined knowing the arrangement of hyperplanes.

3.4 Implications

The centrality of the arrangement confirms that the reduction from the weight space $\mathbb{R}^m_>$ to the normalized weight space $\widetilde{\mathcal{W}}$ is justified for the weighted sum scalarization: Every cell of the arrangement in $\mathbb{R}^m_>$ intersects $\widetilde{\mathcal{W}}$ and, hence, every extreme supported point is represented in this intersection. The decomposition of the normalized weight space $\widetilde{\mathcal{W}}$ is still defined by an arrangement of hyperplanes.

Due to the condition that the sum of weights λ_i, for $i = 1, \dots, n$, should be equal to 1 this arrangement is not central.

The original arrangement of hyperplanes can also be projected on $\mathbb{R}^{m-1}$, such that the projected weight space $\mathcal{W}$ is subdivided by this arrangement of hyperplanes. The arrangement in the projected weight space is called the *associated projected arrangement* of MUCO.

Example 5 The instance of Example 2 has four extreme supported points. Hence, the associated arrangement of hyperplanes has four corresponding cells intersecting with $\mathbb{R}^2_>$. Also the normalized weight space $\widetilde{\mathcal{W}}$ and the projected weight space $\mathcal{W}$ are subdivided into four segments by the associated projected arrangement, cf. Figure 4.

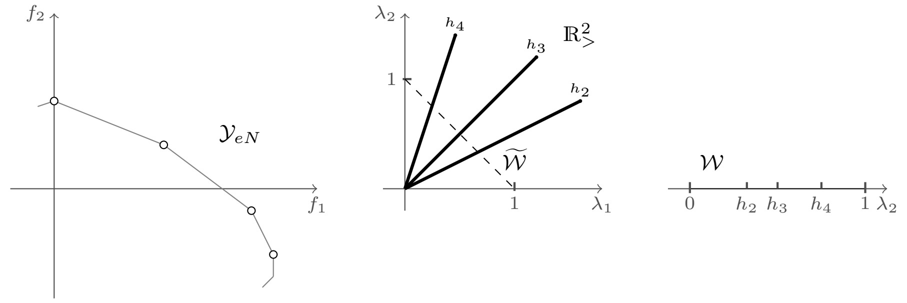

Fig. 4 On the left: extreme supported points of Example 2. In the middle: intersection of the associated arrangement of hyperplanes with the first quadrant in $\mathbb{R}^2$ and intersection with the normalized weight space $\widetilde{\mathcal{W}}$. On the right: associated projected arrangement of hyperplanes in the projected weight space $\mathcal{W}$.

Consider again the complete arrangement of hyperplanes. Each cell of the arrangement corresponds to an extreme point of the associated zonotope and vice versa. Each orthant of $\mathbb{R}^m$ corresponds to a combination of maximization and minimization objectives and the associated notion of dominance. To be more precise: If the weight value λ_j, for $j \in \{1, \dots, m\}$, is positive, then the corresponding extreme points of the zonotope are nondominated for maximizing objective function $f_j(x)$. If the weight value λ_j, for $j \in \{1, \dots, m\}$, is negative, then the corresponding extreme points of the zonotope are nondominated for minimizing objective function $f_j(x)$.

Example 6 Consider the following modification of Example 2 where max is switched to min in the second objective:

$$\begin{array}{ll} \max & \quad - \; x_2 + 3\,x_3 + 6\,x_4 - 5\,x_5 + x_6 \\ \min & x_1 + 2\,x_2 - 3\,x_3 - 2\,x_4 - \; x_5 + x_6 \\ \text{s.t.} & x_i \in \{0,1\}, \qquad i = 1, \dots, 6. \end{array}$$

On the left side of Figure 5, the convex hull of feasible points for this instance of MUCO is shown. The symbols $\bullet$ highlight three extreme points that are nondominated w. r. t. the maximization of the first and minimization of the second objective. The corresponding part of the arrangement of hyperplanes is inside the second quadrant of $\mathbb{R}^2$, i. e., cells intersecting with $\{(\lambda_1, \lambda_2) \in \mathbb{R}^2 : \lambda_1 > 0,\ \lambda_2 < 0\}$, see the right part of Figure 5.

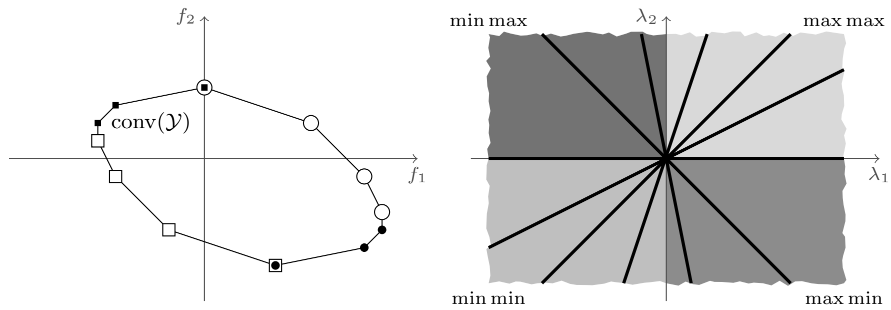

Fig. 5 On the left: extreme points of the convex hull of the feasible set conv($\mathcal{Y}$) of Example 2, where the symbols ○ indicate extreme supported points and the symbols □, •, and ▪ indicate extreme supported points if the objectives would be changed to $(\min f_1, \min f_2)$, $(\max f_1, \min f_2)$, and $(\min f_1, \max f_2)$, respectively. On the right: associated arrangement of hyperplanes in $\mathbb{R}^2$ corresponding to the associated zonotope and parts of the arrangement of hyperplanes that have to be considered for the respective notion of nondominance.

In this context, it is very intuitive to see that switching from maximization to minimization in all objective functions generates the same number of supported points where the supported solutions with respect to maximization are reverse to the supported solutions with respect to minimization. Since the arrangement of hyperplanes is central, all cells reappear in the opposite orthant of $\mathbb{R}^m$ with reversed position vector. Thus, the associated zonotope and the convex hull of the feasible set of the corresponding instance of MUCO are centrally symmetric.

Figure 6 summarizes the interrelations between MUCO, zonotopes and the associated arrangements of hyperplanes. It can be concluded that the number of extreme supported solutions of MUCO is equal to the number of extreme supported points of MUCO. This number is again equal to the number of nondominated extreme points of the associated zonotope and also equal to the number of cells of the arrangement of hyperplanes associated to MUCO.

3.5 Nonextreme supported solutions

The interrelation between MUCO and arrangements of hyperplanes also reveals a necessary condition for the existence of nonextreme supported solutions.

If the associated projected arrangement of hyperplanes is simple, no nonextreme supported solutions occur. As described in Section 2.3, in this case the position vectors of a cell and an adjacent face $\varphi^{(k)}$ differ in $(n-k)$ positions, and in these positions the corresponding entries of $\varphi^{(k)}$ are zero. Recall that a zero-entry in the position vector induces two alternative solutions with $x_i = 0$ and $x_i = 1$, respectively. Hence, a face $\varphi^{(k)}$ corresponds to 2^{n-k} extreme supported solutions that are also induced by the adjacent cells.

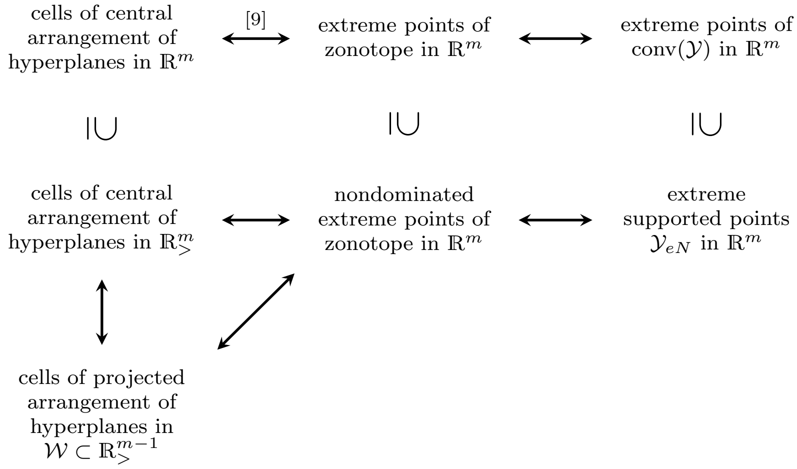

Fig. 6 Summary of the interrelations between MUCO, zonotopes and the associated arrangements of hyperplanes: an bidirectional arrow indicates a one-to-one correspondence. Note that the inclusions are strict in all nontrivial instances.

If the associated projected arrangement of hyperplanes is not simple, a face $\varphi^{(k)}$ may have ℓ, with $\ell > n-k$, zero-entries and will correspond to 2^ℓ supported solutions. Only 2^{n-k} of these solutions correspond to the adjacent cells and, therefore, to extreme supported solutions. Thus, in this case there is an exponential number $(2^\ell - 2^{n-k})$ of nonextreme supported solutions, which might correspond to an exponential number of nonextreme supported points. If, for example, all hyperplanes h_i, $i = 1, \dots, n$, are identical, the problem has two extreme and $(2^n - 2)$ nonextreme supported solutions. The number of corresponding nonextreme supported points can be smaller if there are equivalent solutions, where two solutions x and $\bar{x}$, $x \neq \bar{x}$, are called *equivalent* if their objective function values are equal, i. e., if $f(x) = f(\bar{x})$. Ehrgott [10] presents an instance with 2^n feasible, non-equivalent solutions that all correspond to supported points, where only two of them are extreme:

$$\begin{aligned} \max &\sum_{i=1}^{n} -2^{i-1}\, x_i \\ \max &\sum_{i=1}^{n} 2^{i-1}\, x_i \\ \text{s. t.} \quad & x_i \in \{0,1\}, \qquad i = 1, \dots, n. \end{aligned}$$

The two extreme supported points $(0,0)$ and $(-\sum_{i=1}^n 2^{i-1}, \sum_{i=1}^n 2^{i-1})$ define a line, the bisecting line of the second and forth quadrant, and all other supported points, which are all other feasible points, are lying on that line. For this instance the associated projected arrangement of hyperplanes has two cells and one facet.

4 Solution approach and case study

The interrelation between MUCO and the associated arrangement of hyperplanes in the weight space can be used to set up a solution approach to compute the set of extreme supported solutions of MUCO. In the following the focus will be set on the associated projected arrangement of hyperplanes in $\mathcal{W}$ but, with small adaptions, the solution approach could also be applied on the associated arrangements of hyperplanes in the weight space $\mathbb{R}^m_>$ or the normalized weight space $\widetilde{\mathcal{W}}$.

The determination of the position vectors of all cells of the arrangement induces the set of extreme supported points. The position vectors can be generated by visiting every cell of the arrangement as in Ferrez et al [13], see also Section 2.3. Alternatively, the search can be focused on the intersection points of hyperplanes in $\mathcal{W}$: As described above, each item i, for $i = 1, \dots, n$, defines a hyperplane in $\mathbb{R}^{m-1}$:

$$h_i = \left\{ (\lambda_2, \dots, \lambda_m) \in \mathbb{R}^{m-1} \; : \; \langle p_{\cdot,i}, \lambda \rangle = 0 \text{ with } \lambda = \Big(1 - \sum_{j=2}^{m} \lambda_j, \lambda_2, \dots, \lambda_m\Big) \right\}$$

and two half-spaces

$$h_i^- = \left\{ (\lambda_2, \dots, \lambda_m) \in \mathbb{R}^{m-1} \; : \; \langle p_{\cdot,i}, \lambda \rangle < 0 \text{ with } \lambda = \Big(1 - \sum_{j=2}^{m} \lambda_j, \lambda_2, \dots, \lambda_m\Big) \right\}$$

$$h_i^+ = \left\{ (\lambda_2, \dots, \lambda_m) \in \mathbb{R}^{m-1} \; : \; \langle p_{\cdot,i}, \lambda \rangle > 0 \text{ with } \lambda = \Big(1 - \sum_{j=2}^{m} \lambda_j, \lambda_2, \dots, \lambda_m\Big) \right\}$$

In the case of a simple arrangement of hyperplanes, the intersection point $\lambda \in \mathcal{W}$ of $(m-1)$ hyperplanes is a vertex of the arrangement of hyperplanes and has a position vector $\mathrm{Pos}(\lambda)$ with $(m-1)$ zero-entries. The vertex λ is adjacent to 2^{m-1} cells, each of which has one of the 2^{m-1} possible combinations of values (-1) and $(+1)$ in the positions where λ has a zero-entry, cf. Figure 7. All remaining entries of the position vectors are equal for the vertex and all of its adjacent cells. Thus, the corresponding supported solutions can be generated by identifying the indices of the intersecting hyperplanes and the position of the vertex with respect to all other hyperplanes.

However, it may happen that less than $(m-1)$ intersecting hyperplanes define a face but no vertex in the projected weight space and, thus, are not adjacent to any vertex in $\mathcal{W}$. This can be handled by including additional hyperplanes $h_{n+1}, \dots, h_{n+m}$ that define the boundary of the projected weight space. Every face that intersects the projected weight space intersects one or several of these boundary hyperplanes and defines a vertex on the boundary. Thus, the information of cells that would have no adjacent vertices in $\mathcal{W}$ is also available at the newly generated vertices on its boundary. In return, if such a vertex is defined by more than $(m-1)$ hyperplanes, these vertices may also include information about cells that do not intersect $\mathcal{W}$, see the right part of Figure 7. An appropriate definition of the additional boundary hyperplanes and the corresponding half-spaces can be used to prune cells corresponding to dominated solutions.

Assuming that the number of objective functions m is fixed, the vertices of the arrangement of hyperplanes can be computed with a complexity of $\mathcal{O}(n^{m-1})$ using the naïve approach of testing all $\binom{n+m}{m-1}$ possible intersections of $(m-1)$-tuples of hyperplanes. For fixed m, the complexity for computing the intersection points is constant. The corresponding position vectors and, thus, the corresponding

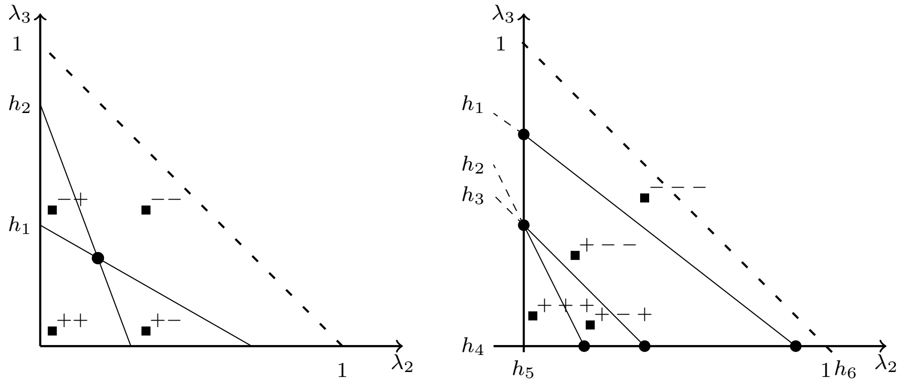

Fig. 7 Arrangements of hyperplanes in $\mathcal{W}$ associated to a tri-objective unconstrained combinatorial optimization problem. The strings in $\{-,+\}^2$ and $\{-,+\}^3$, respectively, refer to the position vectors of corresponding cells (symbol $-$ for entry (-1) and symbol $+$ for entry $(+1)$). On the left: intersection of two hyperplanes in the interior of $\mathcal{W}$ (general case); on the right: cells with no adjacent vertex in the interior of $\mathcal{W}$ but on its boundary and intersection of two hyperplanes on the boundary of $\mathcal{W}$.

extreme supported solutions or the information for pruning can easily be generated by evaluating the position of a vertex with respect to each hyperplane.

Furthermore, in the case of non-simple arrangements, one has to take into account the existence of nonextreme supported points. In the following, the focus will be set on the case of simple arrangements and it is referred to Schulze [25] for more details concerning the non-simple case.

4.1 Case study

In this section, the above results are illustrated on tri-objective unconstrained combinatorial optimization problems:

$$\begin{aligned} \max \quad & f(x) = \left(\sum_{i=1}^{n} p_{1,i}\, x_i \,,\ \sum_{i=1}^{n} p_{2,i}\, x_i \,,\ \sum_{i=1}^{n} p_{3,i}\, x_i \right) \\ \text{s.t.} \quad & x_i \in \{0,1\}, \qquad i = 1, \ldots, n. \end{aligned} \tag{TUCO}$$

A special structure is assumed, namely that all coefficients of the first objective function are positive and all coefficients of the second and third objective function are negative. More precisely, $p_{1,i} > 0$, $p_{2,i} < 0$, and $p_{3,i} < 0$, for all $i = 1, \ldots, n$. Note that this structure occurs, for example, when a maximization objective is combined with two minimization objectives (all with positive coefficients), the latter of which are transformed into equivalent maximization objectives by multiplying all coefficients by (-1).

This structure of TUCO implies that the associated arrangement of hyperplanes also has a special structure. Every hyperplane h_i, for $i = 1, \ldots, n$, intersects with the λ_2- and λ_3-axis in the interval $(0, 1)$:

$$\langle p_{\cdot,i}, \lambda \rangle = 0$$

$$\begin{aligned}
&\Leftrightarrow \quad p_{1,i}(1-\lambda_2-\lambda_3)+p_{2,i}\lambda_2+p_{3,i}\lambda_3=0\\
&\Leftrightarrow \quad p_{1,i}+\lambda_2(-p_{1,i}+p_{2,i})+\lambda_3(-p_{1,i}+p_{3,i})=0\\
\lambda_3=0 \;&\Rightarrow \quad \lambda_2=\frac{-p_{1,i}}{-p_{1,i}+p_{2,i}}\in(0,1)\\
\lambda_2=0 \;&\Rightarrow \quad \lambda_3=\frac{-p_{1,i}}{-p_{1,i}+p_{3,i}}\in(0,1).
\end{aligned}$$

Hence, every hyperplane intersects with the projected weight space $\mathcal{W}=\{(\lambda_2,\lambda_3)\in\mathbb{R}^2_{>} : \lambda_2+\lambda_3\leq 1\}$ and has a negative slope. Furthermore, every half-space h_i^- lies *above* and every half-space h_i^+ lies *below* the corresponding hyperplane h_i. Thus, it is possible to refer to faces of the arrangement as lying above or below a given hyperplane if they are subsets of h_i^- and h_i^+, respectively. Hence, the top-most cell of the arrangement that lies above all hyperplanes has a position vector with only entries equal to (-1). This implies that the solution $x_i=0$ for all $i=1,\dots,n$ is extreme supported. This agrees with the fact that for two of the objective functions the objective function value decreases if any item is included.

Our algorithm for TUCO implements the solution approach presented in the previous paragraph and is, furthermore, based on the work of Bentley and Ottmann [5] for computing the intersections of line segments in the plane. Adopted to our problem, the line segments are defined as the intersections of the respective hyperplanes in the projected weight space. The algorithm sweeps with a vertical line from left to right through the projected weight space, starting at $\lambda_2=0$ and stopping at $\lambda_2=1$ (cf. Figure 8). Every intersection point is reported and the corresponding extreme supported solutions are computed. Note that even though the special structure assumed in this case study facilitates the implementation, it is not essential for the general approach and it has no impact on its worst case complexity.

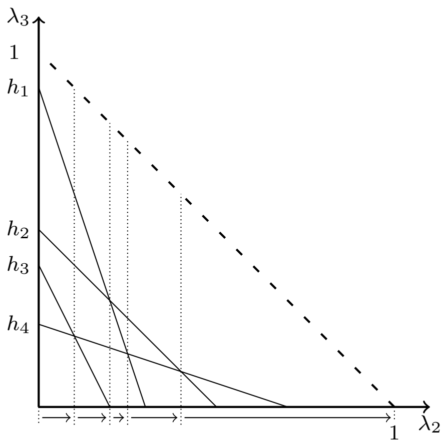

Fig. 8 The method of Bentley and Ottmann [5] sweeps a vertical line through the projected weight space to identify all intersection points.

The algorithm of Bentley and Ottmann [5] has a complexity of $\mathcal{O}(n\log n + k\log n)$, where k is the number of intersection points, which is in $\mathcal{O}(n^2)$ for an arrangement of hyperplanes in $\mathbb{R}^2$. Bentley and Ottmann [5] indicated that the naïve approach of testing all $\binom{n}{2}$ possible intersections of pairs of hyperplanes, which has a complexity of $\mathcal{O}(n^2)$, becomes more efficient than their approach if k is very close to n^2. Nevertheless, in the majority of instances the algorithm of Bentley and Ottmann [5] is preferable since it is very unlikely that nearly all intersection points of the arrangement of hyperplanes are inside the projected weight space $\mathcal{W}$.

Note that the algorithm generates redundant information. A solution is generated as often as the corresponding cell has adjacent vertices. However, the effort for computing and saving the solutions is quite small and no counteractions are needed.

4.2 Numerical results

The general variant of the algorithm was implemented in C++. Experiments were performed on an Intel Quadcore 2,80 GHz with 4 GB RAM. Instances with 100 up to 1000 items were generated. The integer valued coefficients were uniformly distributed and independently chosen in the interval $[1, 10\,n]$ for the first objective function and in $[-10\,n, -1]$ for the second and third objective function.

Table 1 presents average solution times in seconds (t) and numbers of extreme supported solutions ($|\mathcal{X}_{eE}|$), averaged over 30 instances. The algorithm solves TUCO efficiently. In the tests, it can be observed that the number of extreme supported solutions is indeed in $\mathcal{O}(n^2)$, for example, it can be bounded by the quadratic function $g(n) = 0.18\,n^2$. This is again illustrated in Figure 9. The experiments comply with the theoretical results.

n	t	$\lvert\mathcal{X}_{eE}\rvert$	$g(n)$
100	0.05	1 670.43	1 800
200	0.74	6 933.00	7 200
300	3.87	15 117.60	16 200
400	12.78	27 054.57	28 800
500	31.16	41 475.03	45 000
750	177.39	94 228.83	101 250
1000	686.98	168 324.83	180 000

Table 1 CPU-times (in seconds) and number of extreme supported solutions for instances of TUCO with 100 up to 1000 items (always averaged over 30 instances) and the corresponding values of $g(n) = 0.18\,n^2$.

In order to evaluate whether the set of extreme supported points $\mathcal{Y}_{eN}$ is a good representation of the complete nondominated set $\mathcal{Y}_N$, we exemplarily evaluated its ε-indicator value (c.f. [33]) and its contribution to the dominated hypervolume (c.f. [32]) on 30 random TUCO instances with 50 items. In these tests, the point $y_R := (-1, \min\{\bar{y}_2 : \bar{y} \in \mathcal{Y}_N\} - 1, \min\{\bar{y}_3 : \bar{y} \in \mathcal{Y}_N\} - 1)$ was used as a reference point (for ε-indicator and hypervolume), and the ε-indicator was calculated as

$$\varepsilon := \max_{\bar{y} \in \mathcal{Y}_N} \min_{y_i \in \mathcal{Y}_{eN}} \max \left\{ \frac{\bar{y}_1 - y_{R,1}}{y_{1,i} - y_{R,1}}, \frac{\bar{y}_2 - y_{R,2}}{y_{2,i} - y_{R,2}}, \frac{\bar{y}_3 - y_{R,3}}{y_{3,i} - y_{R,3}} \right\},$$

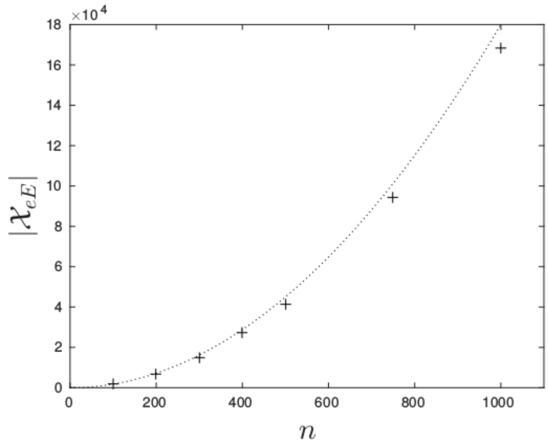

Fig. 9 Comparison of experimental and theoretical results: Number of extreme supported solutions $|\mathcal{X}_{eE}|$ depending on number of items n and $g(n) = 0.18\,n^2$ as dashed line. The quadratic function bounds $|\mathcal{X}_{eE}|$ in our experimental results from above.

i.e., $\bar{y} - y_R \leqq \min_{y_i \in \mathcal{Y}_{eN}} \varepsilon \cdot (y_i - y_R)$ for all $\bar{y} \in \mathcal{Y}_N$.

For all test instances, the complete nondominated set was computed as a reference, using a simple dynamic programming algorithm. On the average, the extreme supported points made up for only 3.9% of all nondominated points, while still providing an excellent 1.0267-approximation w.r.t. the ε-indicator (i.e., when scaling all extreme supported points by a factor of only 1.0267, then they dominate *all* nondominated points) and covering 99.2% of the dominated hypervolume (the latter was computed using the code of [19]).

5 Implications for knapsack problems

In contrast to MUCO, the multi-objective knapsack problem MOKP includes a capacity constraint. Thus, not all subsets of items are feasible in general: the feasible set of MOKP is a subset of that of MUCO. As a consequence, the solution approach presented in Section 4 is not directly applicable. More precisely, since efficient solutions of MUCO may become infeasible for MOKP, other feasible solutions may become efficient for MOKP.

Nevertheless, the projected weight space still provides all information for computing the set of extreme supported solutions using the concept of weight space decomposition (see Section 2.2 and Przybylski et al [23]). The projected weight space can be decomposed into convex polytopes such that the weighted sum scalarization using weights that correspond to one polytope have the same set of optimal solutions. Different from Section 3, the polytopes are in general not generated by an arrangement of hyperplanes. Nevertheless, the definitions of k-faces, cells, facets, edges, vertices and subfaces can be applied analogously.

Note that for MOKP there may exist *equivalent* extreme supported solutions, i. e., extreme supported solutions x and x', $x \neq x'$, that correspond to the same extreme supported point $f(x) = f(x')$. Therefore, the focus is set on computing (extreme) supported points and one corresponding solution in the following.

Several approaches for computing the set of extreme supported points for MOKP have been suggested in the literature. Benson and Sun [4] present an algorithm that uses a weight space decomposition to find all extreme supported points of multi-objective linear optimization problems. Przybylski et al [23] reinterpret the dichotomic search of Aneja and Nair [2] and concentrate on the decomposition of the normalized weight space for computing all extreme supported points of multi-objective integer optimization problems. Özpeynirci and Köksalan [20] develop an algorithm for finding all extreme supported points of multi-objective mixed integer optimization problems on the basis of the ideas of Aneja and Nair [2] and using dummy points. Bökler and Mutzel [6] present an approach for enumerating all extreme supported points of multi-objective combinatorial optimization problems which is based on dual variants of Benson's algorithm for multi-objective linear optimization (see Ehrgott et al [12] and Heyde and Löhne [17]). Przybylski et al [24] introduce two variants of a straightforward dichotomic search algorithm for computing extreme supported points of multi-objective integer linear programming problems. The first variant uses dummy points similar to Özpeynirci and Köksalan [20], and the second solves problems in smaller dimensions on the boundary of the reduced weight space as a preprocessing.

In the following, it is shown how the results on unconstrained problems from Section 4 can be used to improve the computation times for obtaining extreme supported solutions of MOKP when negative profits are present. Again, a case study is performed on tri-objective problems with one positive and two negative objectives, that provides computational results for this case.

5.1 Adaption of solution concepts

As described above, the results of Section 3 are not directly transferable to the constrained problem MOKP. However, the nondominated frontier of the unconstrained problem provides an upper bound set on the nondominated frontier of the same problem with a capacity constraint (see the left part of Figure 10 for an illustration). Efficient solutions of MUCO that do not violate the capacity constraint are also efficient for MOKP. Thereby, a cell of the arrangement of hyperplanes associated to those solutions is contained in the polytope of the weight space decomposition associated to MOKP for the same solution. Hence, the arrangement of hyperplanes associated to MUCO can be computed in a preprocessing step and all resulting feasible points can be used to initialize a multidimensional dichotomic search. All faces of the arrangement corresponding to infeasible solutions have to be analyzed again. These faces are called *infeasible faces*. Similarly, connected parts of infeasible faces in the projected weight space are called *infeasible areas*. Inversely, all faces corresponding to feasible solutions are called *feasible faces* associated to MOKP.

For simple arrangements in the projected weight space, the position vectors of two cells $\varphi_1^{(m-1)}$ and $\varphi_2^{(m-1)}$ that have a common subface $\varphi^{(m-2)}$ differ in exactly one position i that corresponds to the hyperplane h_i which separates the cells. Suppose that $\varphi_1^{(m-1)}$ is a feasible cell and that $\varphi_2^{(m-1)}$ is an infeasible cell for MOKP. In this case, the common subface $\varphi^{(m-2)}$ corresponds to one feasible and one infeasible solution. The two solutions also correspond to the subfaces of $\varphi^{(m-2)}$ and so on. All k-faces $\varphi^{(k)}$, $0 \leq k \leq m-2$, that correspond to at least one feasible and at least one infeasible solution of MOKP are called *critical faces*. For non-simple arrangements, more solutions may correspond to a k-face than in the simple case, but still a face is called critical if feasible *and* infeasible solutions are among them. Thus, feasible and infeasible areas are separated by critical faces.

Example 7

(a) Consider the following bi-objective knapsack problem:

$$\begin{array}{lrl} \max & & - x_2 + 3\,x_3 + 6\,x_4 - 5\,x_5 + x_6 \\ \max & & x_1 + 2\,x_2 - 3\,x_3 - 2\,x_4 - x_5 + x_6 \\ \text{s.t.} & & 5\,x_1 + 3\,x_2 + 2\,x_3 + 4\,x_5 + x_5 + 4\,x_6 \leq 17 \\ & & x_i \in \{0,1\}, \qquad i = 1,\dots,6. \end{array}$$

This is a constrained version of the bi-objective unconstrained combinatorial optimization problem of Example 2. In the left part of Figure 10, the respective nondominated frontiers are shown. Each point in the nondominated frontier of the constrained problem is dominated by or equal to a point in the nondominated frontier of the unconstrained problem.

(b) Consider the following tri-objective knapsack problem:

$$\begin{array}{ll} \max & -72\,x_1 - 22\,x_2 + 46\,x_3 - 36\,x_4 - 8\,x_5 + 4\,x_6 + 11\,x_7 + 22\,x_8 \\ \max & -42\,x_1 + 8\,x_2 + 16\,x_3 + 24\,x_4 - 13\,x_5 + 64\,x_6 - 154\,x_7 - 38\,x_8 \\ \max & 18\,x_1 + 11\,x_2 - 44\,x_3 + 54\,x_4 + 52\,x_5 - 56\,x_6 + 56\,x_7 + 52\,x_8 \\ \text{s.t.} & 13\,x_1 + 6\,x_2 + 8\,x_3 + 8\,x_4 + x_5 + 2\,x_6 + 13\,x_7 + 10\,x_8 \leq 26 \\ & x_i \in \{0,1\}, \qquad i = 1,\dots,8. \end{array}$$

The unconstrained version of this instance has 25 extreme supported solutions. Seven of the cells of the associated arrangement of hyperplanes become infeasible as soon as the constraint is included, see the right part of Figure 10.

In Algorithm 1 an approach for computing all extreme supported points of MOKP is presented, using the concepts of Section 4 for unconstrained problems as a preprocessing. All of the multi-objective approaches for decomposing the weight space presented before can be used in Steps 3 and 4 to compute all extreme supported points of MOKP, since these methods compute extreme supported points iteratively and can directly incorporate already found extreme supported points as starting solution.

Algorithm 1 Generate all extreme supported points of MOKP

Input: MOKP
1: compute the set of extreme supported solutions $\mathcal{X}_{eE}$ of the corresponding MUCO problem
2: $\mathcal{Y}_{eN} := \{y \in \mathbb{Z}^m : y = f(x), x \in \mathcal{X}_{eE} \text{ with } \sum_{i=1}^{n} w_i x_i \leq W\}$ // *filter feasible points*
3: choose an algorithm for computing the weight space decomposition for MOKP and use all points $y \in \mathcal{Y}_{eN}$ to initialize it (e.g., [4, 6, 20, 23, 24])
4: apply the chosen algorithm to complete $\mathcal{Y}_{eN}$
Output: $\mathcal{Y}_{eN}$

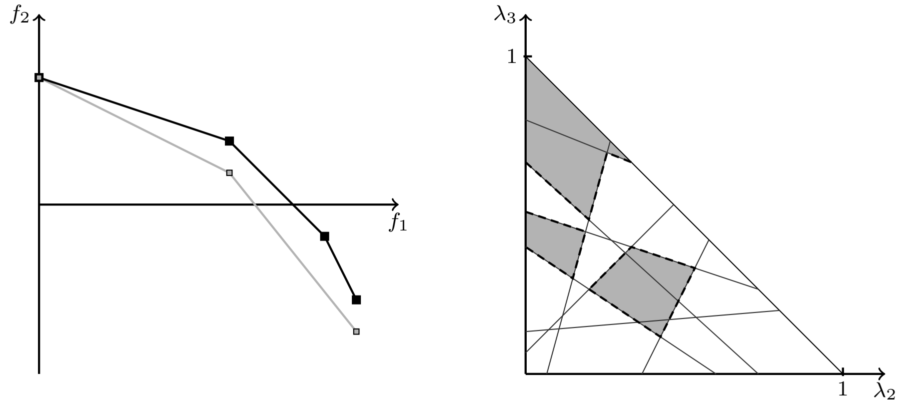

Fig. 10 Objective space (left) and projected weight space (right) corresponding to Examples 7(a) and (b), respectively. On the left: Supported points and nondominated frontiers of the constrained (gray) and unconstrained (black) problem. On the right: Arrangement of hyperplanes associated to the unconstrained version of the instance of 7(b). Including the constraint, all faces inside the gray shaded areas correspond to infeasible solutions. The dashed line segments highlight the critical edges and vertices.

5.2 Case study

In this section, the above results are again demonstrated on a more specific problem with three objectives. It is assumed that the first objective has positive coefficients and the second and third objective have negative coefficients.

$$\begin{aligned} \max \quad & f(x) = \left(\sum_{i=1}^{n} p_{1,i}\, x_i \,,\; \sum_{i=1}^{n} p_{2,i}\, x_i \,,\; \sum_{i=1}^{n} p_{3,i}\, x_i \right) \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i x_i \leq W \\ & x_i \in \{0,1\}, \qquad i = 1, \ldots, n \end{aligned} \tag{TOKP}$$

with $p_{1,i} > 0$, $p_{2,i} < 0$, $p_{3,i} < 0$, $W > 0$ and $w_i > 0$ for all $i = 1, \ldots, n$. As in the case of TUCO, the second and third objective may model minimization objectives that were transformed into maximization objectives by multiplication with (-1). Instances with this structure occur, for example, when soft constraints in a single-objective three-dimensional knapsack problem are relaxed and re-interpreted as objective functions in an associated three-objective one-dimensional knapsack problem, see [26].

As discussed in Section 4, the unconstrained version of TOKP, the problem TUCO, has a particular weight space decomposition. For simplicity, the terms *above* and *below* are again used to describe the relations of cells to hyperplanes and intersection points as defined in Section 4. In the upper left part of the projected weight space where the first (positive) objective function is only marginally weighted, the supported solution x corresponding to the cell $\varphi_0^{(2)}$ is $x = 0_n$. I. e., none of the items is included, which is certainly feasible for TOKP. If it is assumed that there do not exist identical hyperplanes, which will be done in the following, it can be concluded that solutions corresponding to cells that are adjacent to the same edge differ in exactly one item. Due to the fact that all half-spaces h_i^+ are lying below the corresponding hyperplanes, all cells above the separating hyperplane correspond to solutions with $x_i = 0$ and all cells below the hyperplane correspond to solutions with $x_i = 1$.

Since every hyperplane has a negative slope, it is possible to define a path that iteratively moves from one cell to an adjacent cell, starting from $\varphi_0^{(2)}$ and arriving at any other cell, such that in each step (from one cell to the next) one additional item is included. Note that this path is, in general, not unique.

Example 8 Consider the following instance of TOKP:

$$\begin{array}{ll}
\max & 16\,x_1 + 21\,x_2 + 10\,x_3 + \;\;9\,x_4 + \;\;3\,x_5 \\
\max & -24\,x_1 - 14\,x_2 - 10\,x_3 - \;\;\;\;x_4 - 27\,x_5 \\
\max & -4\,x_1 - \;\;9\,x_2 - 10\,x_3 - 21\,x_4 - 12\,x_5 \\
\text{s.t.} & \;\;4\,x_1 + \;\;3\,x_2 + \;\;3\,x_3 + \;\;2\,x_4 + \;\;\;\;x_5 \leq 5 \\
& x_i \in \{0,1\}, \qquad i = 1,\ldots,5.
\end{array}$$

The corresponding arrangement of hyperplanes in the projected weight space is illustrated in Figure 11. Now consider the two extreme supported solutions $x_0 = (0,0,0,0,0)^\top$ and $\hat{x} = (0,1,1,1,0)^\top$ and the corresponding cells $\varphi_0^{(2)}$ and $\hat{\varphi}^{(2)}$. The solution $\hat{x}$ can be obtained starting from x_0 by adding the items 2, 3, and 4. Thus, a path from $\varphi_0^{(2)}$ to $\hat{\varphi}^{(2)}$, additionaly including one item at each step, has to cross the hyperplanes h_2, h_3, and h_4. In this case, there are two alternative paths: first crossing h_2 or h_4, and then crossing h_4 or h_2, respectively, and finally crossing h_3.

Clearly, if a cell is feasible, then all previous cells on this path are also feasible. Accordingly, if a cell is infeasible, then all subsequent cells on a path are also infeasible. See Figure 11 for an illustration.

The cell $\varphi_1^{(2)}$ which has $(\lambda_2, \lambda_3)^\top = (0,0)^\top$ in its boundary corresponds to the solution $x = (1,\ldots,1)^\top \in \mathbb{R}^n$ since the second and third (negative) objective function are only marginally weighted. This solution is certainly infeasible for TOKP (unless $\sum_{i=1}^n w_i \leq W$). Analogously to the above discussion, starting with $\varphi_1^{(2)}$, a path to every other cell of the arrangement can be defined where one variable is set to 0 at each step. If one cell is infeasible, then every other cell on this path up to the cell in question is also infeasible. If one cell is feasible, then all subsequent cells on the path are also feasible.

Concluding, all feasible cells are connected as well as all infeasible cells are connected. The projected weight space consists of one feasible and one infeasible area. A set of critical faces separates the two areas and the set of critical faces is also connected. The connectedness of critical faces can be used to compute all feasible cells and a minimum number of infeasible cells by iteratively considering intersections of hyperplanes in non-decreasing order of λ_2. For more details on this approach see [25].

Example 9 Figure 11 shows the resulting subdivision of the projected weight space corresponding to the instance introduced in Example 8 in one feasible and one infeasible area. The two cells corresponding to the solutions $(1,1,0,0,0)^\top$ with a total weight of 7 and $(0,1,1,1,0)^\top$ with a weight of 8 mark the beginning of the infeasible part for any path from $\varphi_0^{(2)}$ to $\varphi_1^{(2)}$. One such path, subsequently including items 2, 4, 3, 1, and 5, is illustrated in the figure.

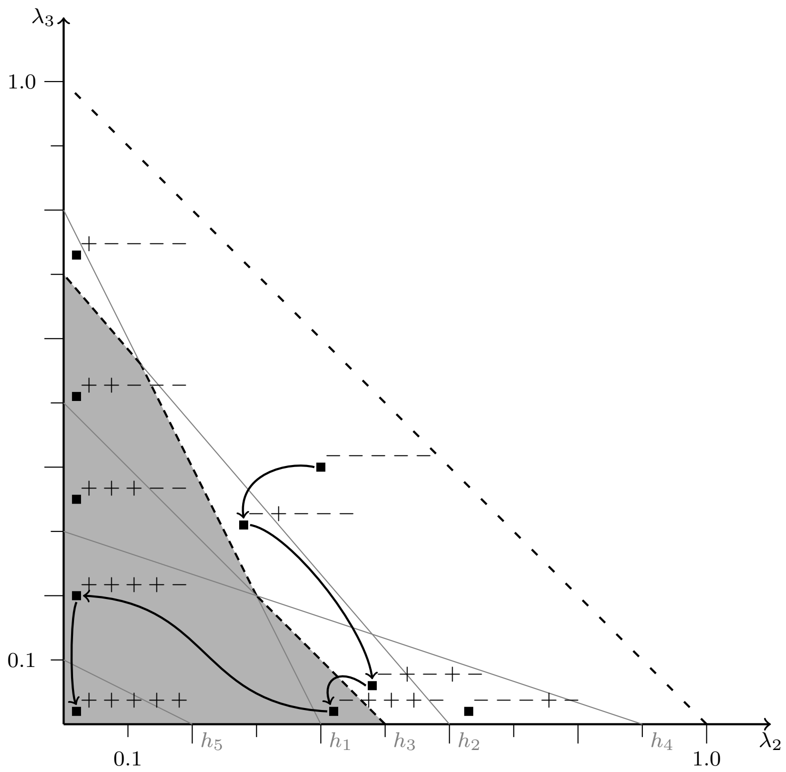

Fig. 11 Projected weight space for the unconstrained version of Example 9, separated into a feasible (white) and an infeasible (gray) area. The critical edges are dashed. An exemplary path from $\varphi_0^{(2)}$ to $\varphi_1^{(2)}$ is displayed. On this path, the decision of including item 3 makes the solution infeasible.

5.3 Numerical results

The approach was implemented in C++. The dichotomic search algorithm (Steps 3 and 4 of Algorithm 1) was coded using the approach of Przybylski et al [24]. The classical one-objective knapsack problem (KP), which arises when solving the weighted sum problems, were solved using the Minknap algorithm of Pisinger [22], see also [21]. The experiments were performed on an Intel Quadcore 2,80 GHz with 4 GB RAM.

Instances were generated following the described scheme for TOKP with positive coefficients in the first objective function and the constraint, and negative coefficients in the second and third objective function. All coefficients are integer values. The absolute values of the coefficients were chosen uniformly distributed in the interval $[1, 10n]$. Three different values for the slackness c of the constraint, such that $W = c \sum_{i=1}^{n} w_i$, were tested. Instances with large value for the slackness usually allow a larger percentage of solutions that are not influenced by the constraint as compared to instances with a small slackness value. Instances with 100 up to 1000 variables were tested. The solution times and numbers of extreme supported points, always averaged over 30 instances, are presented in Table 2.

		time in s			$\lvert\mathcal{X}_{sE}\rvert$		
c	n	A	PP	DS	A	PP	DS
0.25	100	0.077	0.016	0.061	657.5	439.0	218.5
	200	0.783	0.087	0.696	2 554.7	1 737.9	816.8
	300	2.917	0.289	2.628	5 258.1	3 626.4	1 631.7
	400	8.397	0.998	7.399	9 691.0	6 949.7	2 741.4
	500	18.844	2.469	16.375	14 575.0	10 544.2	4 030.7
	750	100.089	12.103	87.986	32 272.8	23 405.5	8 867.3
	1000	414.929	45.000	369.929	57 588.3	42 788.1	14 800.2
0.50	100	0.073	0.033	0.040	1 224.3	1 094.8	129.4
	200	0.954	0.349	0.605	4 969.0	4 459.2	509.8
	300	4.449	1.789	2.660	10 633.2	9 531.1	1 102.1
	400	13.254	6.130	7.124	19 058.7	17 302.8	1 755.9
	500	31.126	15.185	15.941	29 233.6	26 678.7	2 554.9
	750	194.792	80.068	114.724	66 361.0	60 630.6	5 730.3
	1000	879.887	314.798	565.089	117 892.7	108 505.5	9 387.3
0.75	100	0.061	0.049	0.012	1 552.0	1 511.3	40.7
	200	0.870	0.652	0.218	6 443.2	6 289.8	153.4
	300	4.413	3.469	0.944	13 951.2	13 629.1	322.1
	400	14.494	11.757	2.737	25 113.5	24 578.5	535.0
	500	34.734	28.822	5.912	38 529.5	37 788.9	740.6
	750	209.799	158.434	51.365	87 614.3	86 087.6	1 526.7
	1000	925.996	692.988	233.008	155 924.2	153 257.2	2 667.0

Table 2 CPU-times, number of supported points and time per point for instances of TOKP with 100 up to 1000 items (always averaged over 30 instances) and three values for the slackness c. The columns show the results for the overall algorithm (A), and the partial results for the preprocessing (PP) and the dichotomic search (DS).

Note that numerical instabilities may occur in the dichotomic search algorithm that is applied in the second phase, and actually some extreme supported points may be missed. This is due to the fact that, on the one hand, the knapsack solver requires integer values as input data, but on the other hand the coefficients of the weighted sum problems that are solved in the dichotomic algorithm may become too large if expanded to values in $\mathbb{N}$. Thus, the weight coefficients may have to be rounded. However, the results are still very clear. Since the goal is to test the preprocessing algorithm, which is exact, the solver of Pisinger [21] is nevertheless applied for the second phase.

The results show that most of the extreme supported points belong to the feasible part of the arrangement of hyperplanes and can be computed by the preprocessing (PP), even for instances with a slackness $c = 0.25$. In contrast, most of the CPU-time is spent for the dichotomic search (DS), which confirms that it is useful to apply the preprocessing algorithm.

		DS with PP		DS without PP	
c	n	time in s	$\lvert\mathcal{X}_{sE}\rvert$	time in s	$\lvert\mathcal{X}_{sE}\rvert$
0.25	100	0.077	657.5	0.252	657.5
	200	0.783	2 554.7	3.216	2 554.5
	300	2.917	5 258.1	13.099	5 256.1
	400	8.397	9 691.0	49.426	9 684.7
	500	18.844	14 575.0	136.275	14 561.4
	750	100.089	32 272.8	809.173	32 184.2
	1000	414.929	57 588.3	2 723.480	57 356.0
0.50	100	0.073	1 224.3	0.777	1 224.0
	200	0.954	4 969.0	11.477	4 968.2
	300	4.449	10 633.2	62.492	10 626.7
	400	13.254	19 058.7	255.941	19 038.0
	500	31.126	29 233.6	659.510	29 190.0
	750	194.792	66 361.0	3 617.104	66 084.5
	1000	879.887	117 892.7	11 501.604	117 176.4
0.75	100	0.061	1 552.0	1.203	1 551.7
	200	0.870	6 443.2	19.406	6 441.9
	300	4.413	13 951.2	123.193	13 941.3
	400	14.494	25 113.5	474.698	25 081.2
	500	34.734	38 529.5	1 189.565	38 461.1
	750	209.799	87 614.3	6 404.522	87 186.5
	1000	925.996	155 924.2	20 088.693	154 820.2

Table 3 CPU-times and number of supported points for instances of TOKP with 100 up to 1000 items (always averaged over 30 instances) computed by a dichotomic search including the preprocessing as described above (DS with PP) and by a pure dichotomic search algorithm (DS without PP).

For comparison, the set of extreme supported points was also computed for all instances using solely the dichotomic search algorithm (DS without PP) of Przybylski et al [24]. The results are presented in Table 3 and are compared to those achieved by including the preprocessing (DS with PP). It can be observed that the number of extreme supported points differs depending on the applied approach. This can be explained by the numerical instabilities, which of course have more influence when solely using the dichotomic search algorithm. The results again clearly show that the preprocessing considerably speeds up the process.

6 Conclusion and further ideas

In this article the interrelation between the multi-objective unconstrained combinatorial optimization problem MUCO, zonotopes, arrangements of hyperplanes, and weight space decompositions was analyzed. It was shown that the convex hull of feasible points $\text{conv}(\mathcal{Y})$ in the objective space can be defined by a zonotope and that the corresponding weight space decomposition is built by an arrangement of hyperplanes. As a consequence, each extreme supported point is generated by exactly one extreme supported solution and that the number of extreme supported solutions is bounded by $2\sum_{i=0}^{m-1}\binom{n-1}{i}$. Hence, for a fixed number of objectives m, MUCO has at most $\mathcal{O}(n^{m-1})$ extreme supported solutions which is polynomial in the number of items. This is particularly interesting since MUCO problems are intractable, i.e., the number of *all* efficient solutions may still grow exponentially with the problem size. Nevertheless, the results in this paper show that a meaningful representation can be obtained in polynomial time.

It was shown that the structure of the arrangement of hyperplanes in the weight space allows an efficient computation of the extreme supported solutions of MUCO. Computational results for tri-objective problems with positive coefficients in the first and negative coefficients in the second and third objective function were presented. Furthermore, it was demonstrated that the computation of extreme supported solutions for MUCO can be used as a preprocessing for dichotomic search algorithms for MOKP. Our numerical study for TOKP reveals that this approach considerably speeds up the process of computing all extreme supported points of MOKP. Future research should address an extension to multi-objective multidimensional knapsack problems, i. e., multi-objective knapsack problems with two or more constraints, with positive coefficients in the constraints and, in a second step, including arbitrary integer coefficients in the constraints. This extension would naturally continue our studies.

It is an interesting open question whether the results obtained for MUCO can be transferred to other classes of multi-objective combinatorial optimization problems. As an example, Seipp [27] shows that arrangements of hyperplanes also appear in the context of multi-objective minimum spanning tree problems. Whether similar relations occur for other MOCO problems is subject of future research.

Acknowledgement

This work was partially supported by the bilateral cooperation project Multiobjective Combinatorial Optimization: Beyond the Biobjective Case funded by the Deutscher Akademischer Austausch Dienst (DAAD, Project-ID 57212018).

References

1. Aissi H, Mahjoub AR, McCormick ST, Queyranne M (2015) Strongly polynomial bounds for multiobjective and parametric global minimum cuts in graphs and hypergraphs. Mathematical Programming 154(1):3–28
2. Aneja YP, Nair KPK (1979) Bicriteria transportation problem. Management Science 25(1):73–78
3. Benson HP, Sun E (2000) Outcome space partition of the weight set in multiobjective linear programming. Journal of Optimization Theory and Applications 105(1):17–36
4. Benson HP, Sun E (2002) A weight set decomposition algorithm for finding all efficient extreme points in the outcome set of a multiple objective linear program. European Journal of Operational Research 139:26–41
5. Bentley JL, Ottmann TA (1979) Algorithms for reporting and counting geometric intersections. IEEE Transactions on Computers C-28(9):643–647
6. Bökler F, Mutzel P (2015) Output-sensitive algorithms for enumerating the extreme nondominated points of multiobjective combinatorial optimization problems. In: Bansal N, Finocchi I (eds) Algorithms - ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, Springer, Berlin, pp 288–299

7. Bökler F, Ehrgott M, Morris C, Mutzel P (2017) Output-sensitive complexity of multiobjective combinatorial optimization. Journal of Multicriteria Decision Analysis 24(1–2):25–36
8. Buck R (1943) Partition of space. The American Mathematical Monthly 50(9):541–544
9. Edelsbrunner H (1987) Algorithms in Combinatorial Geometry. Springer, Berlin Heidelberg
10. Ehrgott M (2005) Multicriteria Optimization. Springer
11. Ehrgott M, Gandibleux X (2000) A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum 22:425–460
12. Ehrgott M, Löhne A, Shao L (2012) A dual variant of Benson's "outer approximation algorithm" for multiple objective linear programming. Journal of Global Optimization 52:757–778
13. Ferrez JA, Fukuda K, Liebling TM (2005) Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. European Journal of Operational Research 166:35–50
14. Figueira JR, Fonseca CM, Halffmann P, Klamroth K, Paquete L, Ruzika S, Schulze B, Stiglmayr M, Willems D (2017) Easy to say they are hard, but hard to see they are easy - towards a categorization of tractable multiobjective combinatorial optimization problems. Journal of Multi-Criteria Decision Analysis 24:82–98
15. Gaas S, Saaty T (1955) The computational algorithm for the parametric objective function. Naval Research Logistics Quarterly 2:39–45
16. Geoffrion AM (1968) Proper eficiency and the theory of vector maximization. Journal of mathematical analysis and applications 22:618–630
17. Heyde F, Löhne A (2008) Geometric duality in multiple objective linear programming. SIAM Journal of Optimization 19(2):836–845
18. Karmarkar N (1984) A new polynomial-time algorithm for linear programming. Combinatorica 4(4):373–395
19. Lacour R, Klamroth K, Fonseca CM (2015) A box decomposition algorithm to compute the hypervolume indicator. CoRR abs/1510.01963, URL http://arxiv.org/abs/1510.01963, 1510.01963
20. Özpeynirci Ö, Köksalan M (2010) An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. Management Science 56(12):2302–2315
21. Pisinger D (1997) A minimal algorithm for the 0–1 knapsack problem. Operations Research 46(5):758–767
22. Pisinger D (2015) Minknap algorithm. URL http://www.diku.dk/∼pisinger/codes.html
23. Przybylski A, Gandibleux X, Ehrgott M (2010) A recursive algorithm for finding all nondominated extreme points in the outcome set of a multiobjective integer programme. INFORMS Journal on Computing 22(3):371–386
24. Przybylski A, Klamroth K, Lacour R (2017) A simple and efficient dichotomic search algorithm for multi-objective integer linear programmes, submitted manuscript
25. Schulze B (2017) New Perspectives on Multi-Objective Knapsack Problems. Ph.D. Thesis. Shaker Verlag, Aachen
26. Schulze B, Paquete L, Klamroth K, Figueira JR (2017) Bi-dimensional knapsack problems with one soft constraint. Computers & Operations Research

78:15–26

27. Seipp F (2013) On Adjacency, Cardinality, and Partial Dominance in Discrete Multiple Objective Optimization. Dr. Hut Verlag, München
28. Gomes da Silva C, Clímaco G, Figueira JR (2004) Geometrical configuration of the Pareto frontier of bi-criteria $\{0,1\}$-knapsack problems. Research Report 16-2004, INESC-Coimbra, Portugal
29. Visée M, Teghem J, Pirlot M, Ulungu EL (1998) Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. Journal of Global Optimization 12:139–155
30. Wolsey LA (1998) Integer Programming. Series in discrete mathematics and optimization, Wiley-Interscience
31. Zaslavsky T (1975) Facing up to arrangements: face-count formulas for partitions of space by hyperplanes. Memoirs of the American Mathematical Society 1(154)
32. Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben AE, Bäck T, Schoenauer M, Schwefel HP (eds) Parallel Problem Solving from Nature – PPSN V, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 292–301
33. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Grunert da Fonseca V (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation 7(2):117–132