Discrete Optimization 7 (2010) 149–165

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives

Anthony Przybylski [a,*], Xavier Gandibleux [a], Matthias Ehrgott [b]

[a] LINA – Laboratoire d'Informatique de Nantes Atlantique, UMR CNRS 6241, Université de Nantes, 2 Rue de la Houssinière BP 92208, 44322 Nantes Cedex 03, France

[b] Department of Engineering Science, The University of Auckland, Private Bag 92019, 70 Symonds Street, Auckland Mail Center, Auckland 1142, New Zealand

ARTICLE INFO

Article history:
Received 27 February 2009
Received in revised form 12 March 2010
Accepted 15 March 2010
Available online 18 April 2010

Keywords:
Multi-objective combinatorial optimization
Two phase method
Assignment problem
Ranking

ABSTRACT

In this paper, we present a generalization of the two phase method to solve multi-objective integer programmes with $p > 2$ objectives. We apply the method to the assignment problem with three objectives.

We have recently proposed an algorithm for the first phase, computing all supported efficient solutions. The second phase consists in the definition and the exploration of the search area inside of which nonsupported nondominated points may exist. This search area is not defined by trivial geometric constructions in the multi-objective case, and is therefore difficult to describe and to explore. The lower and upper bound sets introduced by Ehrgott and Gandibleux in 2001 are used as a basis for this description.

Experimental results on the three-objective assignment problem where we use a ranking algorithm to explore the search area show the efficiency of the method.

1. Introduction

The two phase method was first introduced by Ulungu and Teghem [1]. It is a general method for solving multi-objective combinatorial optimization (MOCO) problems, although it has never been applied for problems with more than two objectives.

In Phase 1 supported efficient solutions are found. Phase 1 is based on Geoffrion's theorem [2] that all (properly) efficient solutions of convex multi-objective programmes are precisely the optimal solutions of weighted sum problems with positive weights.

In Phase 2 nonsupported efficient solutions are found, usually using enumerative methods. To that end, information from supported nondominated points generated in Phase 1 is used to determine a search area in objective space that is guaranteed to contain all nonsupported nondominated points. In the bi-objective case, the search area consists of triangles defined by two consecutive supported efficient solutions. To search for nonsupported efficient solutions in an effective manner, lower bounds, upper bounds, reduced costs, etc. can be employed. The method has been applied to a number of bi-objective problems, e.g. assignment [3,1], network flow [4,5], knapsack [6], and spanning tree [7].

Most often the method uses efficient algorithms for single objective versions of the problem. Because such algorithms are problem specific it is necessary to preserve constraint structure of the problem throughout the solution procedure. It is therefore, for example, not possible to add constraints on objective function values as is done in other multi-objective integer programming methods, e.g., [8,9], see also [10] for a discussion of the difficulty of solving scalarized MOIPs.

* Corresponding author. Tel.: +33 0 251125825; fax: +33 0 251125812.
E-mail addresses: Anthony.Przybylski@univ-nantes.fr (A. Przybylski), Xavier.Gandibleux@univ-nantes.fr (X. Gandibleux), m.ehrgott@auckland.ac.nz (M. Ehrgott).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2010.03.005

Despite the fact that most of the exact methods which are able to generate a complete set are some modifications of the two phase method, no generalization to more than two objectives which follows the original idea has been proposed. The main reasons are that neither the computation of the supported efficient solutions nor the reduction of the search area using the supported nondominated points are trivial.

In this paper we develop such an extension of the two phase method to solve multi-objective integer linear programmes

$$\min\{(z_1(x), \dots, z_p(x)) = Cx : x \in X\}, \tag{MOIP}$$

where $p \geqq 2$ and $C \in \mathbb{R}^{p\times n}$. X denotes the set of feasible solutions of the problem and is defined by

$$X = \{x \in \mathbb{Z}^n : Ax \leqq b,\ x \geqq 0\},$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. We call $\mathbb{R}^n$ decision space ($X \subset \mathbb{R}^n$) and $\mathbb{R}^p$ objective space. $Y := \{Cx : x \in X\} \subset \mathbb{R}^p$ is called the feasible set in objective space or the outcome set.

To compare vectors we adopt the following notation. Let $y^1, y^2 \in \mathbb{R}^p$. We write $y^1 \leqq y^2$ if $y^1_k \leqq y^2_k$ for $k = 1, \dots, p$, $y^1 \leq y^2$ if $y^1 \leqq y^2$ and $y^1 \neq y^2$, and $y^1 < y^2$ if $y^1_k < y^2_k$ for $k = 1, \dots, p$. We define $\mathbb{R}^p_{\geqq} := \{x \in \mathbb{R}^p : x \geqq 0\}$ and analogously $\mathbb{R}^p_{\geq}$ and $\mathbb{R}^p_{>}$.

For $S \subset \mathbb{R}^n$ we denote by cl$S$ the closure of S and by $S^c = \mathbb{R}^p \setminus S$ the complement of S. S is called $\mathbb{R}^p_{\geqq}$-closed if $S + \mathbb{R}^p_{\geqq}$ is closed and $\mathbb{R}^p_{\geqq}$-bounded if there exists $s^0 \in \mathbb{R}^p$ such that $S \subset s^0 + \mathbb{R}^p_{\geqq}$. We denote $S_N := \{s \in S : (s - \mathbb{R}^p_{\geq}) \cap S = \{s\}\}$. For $k \in \{1, \dots, p\}$ we denote by e^k the kth unit vector.

We make the general assumption that there is no feasible solution $x \in X$ which minimizes all p objectives simultaneously.

Definition 1. A feasible solution $\hat{x} \in X$ is *efficient* if there does not exist any other feasible solution $x \in X$ such that $z(x) \leq z(\hat{x})$. $z(\hat{x})$ is then called a *nondominated point*. If $x, x' \in X$ are such that $z(x) \leq z(x')$ we say that x dominates x' and $z(x)$ dominates $z(x')$. If $z(x) = z(x')$, x and x' are *equivalent*. Y_N denotes the set of all nondominated points of Y and X_E denotes the set of efficient solutions (see Definition 2).

It is necessary to distinguish two types of efficient solutions.

- *Supported* efficient solutions are optimal solutions of a weighted sum single-objective problem

$$\min\left\{\sum_{k=1}^{p} \lambda_k z_k(x) : x \in X\right\} \tag{P$_\lambda$}$$

  for some $\lambda \in \mathbb{R}^p_{>}$. The set of supported efficient solutions is denoted X_{SE}. The images of supported efficient solutions, the supported nondominated points Y_{SN}, are located on the boundary of the convex hull conv Y of Y, i.e., they are nondominated points of $(\text{conv}\,Y) + \mathbb{R}^p_{\geqq}$.
- *Nonsupported* efficient solutions X_{NE} are efficient solutions that are not optimal solutions of (P$_\lambda$) for any $\lambda \in \mathbb{R}^p_{>}$. Nonsupported nondominated points Y_{NN} are located in the interior of conv Y. No theoretical characterisation which leads directly to a procedure for the computation of the nonsupported efficient solutions is known.

In addition we can distinguish two classes of supported efficient solutions.

- The objective vectors $z(x)$ of *extremal* supported efficient solutions are located on the vertex set of conv Y. The corresponding points in objective space are called *nondominated extreme points*. We use the notation X_{SE1} and Y_{SN1} in decision and objective space, respectively.
- For all other $x \in X_{SE}$, $z(x)$ is located in the relative interior of a face of conv Y. For such an x, there exist $x_1, x_2 \in X_{SE}$ and $\alpha \in]0, 1[$ such that $z(x) = \alpha z(x_1) + (1-\alpha)z(x_2)$. We refer to these solutions and points by X_{SE2} and Y_{SN2}, respectively.

Due to the existence of equivalent solutions specific subsets of X_E can be classified.

Definition 2. 1. [11] A *complete set* X_E is a set of efficient solutions such that all $x \in X \setminus X_E$ are either dominated by or equivalent to at least one $x' \in X_E$. I.e., for each nondominated point $y \in Y_N$ there exists at least one $x \in X_E$ such that $z(x) = y$.
2. [11] A *minimal complete set* X_{E_m} is a complete set without equivalent solutions. Any complete set contains a minimal complete set.
3. The *maximal complete set* X_{E_M} is the complete set including all equivalent solutions, i.e all $x \in X \setminus X_{E_M}$ are dominated.

Supported and nonsupported efficient solutions can also be classified using Definition 2. Thus we can talk about complete sets of supported and nonsupported efficient solutions as well as minimal and maximal complete sets of supported and nonsupported efficient solutions X_{SE_m}, X_{NE_m}, X_{SE_M}, and X_{NE_M}. For algorithms to solve MOIPs it is important to clearly specify what class of efficient solutions are found because the problems of finding different classes of efficient solutions are considerably different, e.g. finding X_{SE_m} is much simpler than finding X_{E_M}.

In this paper we use the assignment problem with three objectives to illustrate our methods and we will propose an algorithm for the exact solution (finding a minimal or maximal complete set) of this problem. The three-objective assignment problem (3AP) is defined as follows.

$$\begin{aligned} \min z_k(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^k x_{ij} & \qquad k = 1, 2, 3 \\ \sum_{i=1}^{n} x_{ij} = 1 & \qquad j = 1, \ldots, n \\ \sum_{j=1}^{n} x_{ij} = 1 & \qquad i = 1, \ldots, n \\ x_{ij} \in \{0, 1\} & \qquad i, j = 1, \ldots, n \end{aligned} \tag{3AP}$$

where all objective function coefficients c_{ij}^k are non-negative integers and $x = (x_{11}, \ldots, x_{nn})$ is the matrix of decision variables.

The remainder of the paper is structured as follows. Section 2 presents the main difficulties in the extension of the two phase method from two to three objectives. In Section 3, a new description of the search area in Phase 2 is proposed, and a way to explore it is next described in Section 4. General upper bound values usable for this exploration are derived from the bi-objective case in Section 5. Finally, the exploration of the search area is specialized to the three-objective assignment problem in Section 6 and experimental results are shown in Section 7.

2. The two phase method: from two to three objectives

In our multi-objective two phase method we follow the scheme of the original bi-objective method by Ulungu and Teghem [1]. In Phase 1 we compute a complete set of supported efficient solutions. We have discussed this procedure in [12]. Starting with a suitable subset of nondominated extreme points $\{y^1, \ldots, y^k\}$, our algorithm computes the subsets $W^0(y^i)$ of $W^0 = \{\lambda \in \mathbb{R}^p_> : \sum_{k=1}^p \lambda_k = 1\}$ for which y^i attains minimal values of $\langle \lambda, y\rangle$ over Y. This is done by computing the boundary of each set $W^0(y^i)$, and allows to discover new nondominated extreme points y. The procedure stops when $W^0(y)$ does not change for any y. At that stage a complete set Y_{SN_1} is known.

Finally, our algorithm not only computes all nondominated extreme points but also the corresponding partition of the weight set W^0. This fact is used to determine appropriate weight vectors to obtain the set Y_{SN} (with the maximal complete set X_{SE_M}) by enumeration as well as the faces of conv Y defined by the nondominated extreme points.

Next we illustrate the problems of generalizing the second phase to three objectives. Let us assume that at conclusion of Phase 1 for a bi-objective problem we have $Y_{SN} = \{y^1, \ldots, y^r\}$. Then $y_1^i < y_1^{i+1}$ and $y_2^i > y_2^{i+1}$ for all $i = 1, \ldots, r-1$. Such a natural order of nondominated points does not exist for $p \geqq 3$ objectives, causing a number of difficulties.

- In the bi-objective case all maximal nondominated faces of conv Y have dimension 1. The normals to these facets define search directions for nonsupported nondominated points or weights for (P_λ) problems when ranking methods are used in Phase 2 as in [3]. For $p > 2$ these maximal nondominated faces can have any dimension between 1 and $p-1$ and the normals are not well defined.
- It is clear that $Y_N \subset y^N - \mathbb{R}^p_{\geqq}$, where y^N is the nadir point defined by

$$y_k^N := \max_{y \in Y_N} y_k; \quad k = 1, \ldots, p.$$

In the bi-objective case $y^N = (y_2^1, y_1^r)$, i.e., it is defined by the two lexicographically optimal points and known after Phase 1. In problems with $p \geqq 3$ objectives this is no longer the case [13]. Indeed, it is not even determined by Y_{SN}, and therefore in general not known after Phase 1.

Example 1. Consider the following instance of the assignment problem with three objectives

$$C^1 = \begin{pmatrix} 6 & 3 & 12 \\ 13 & 17 & 10 \\ 9 & 14 & 16 \end{pmatrix}, \qquad C^2 = \begin{pmatrix} 10 & 18 & 15 \\ 19 & 7 & 12 \\ 11 & 16 & 14 \end{pmatrix}, \quad \text{and} \quad C^3 = \begin{pmatrix} 12 & 8 & 7 \\ 19 & 18 & 15 \\ 2 & 10 & 0 \end{pmatrix}.$$

The set of efficient solutions is
– x^1 with $x_{12} = x_{23} = x_{31} = 1$ and objective point $y^1 = (22, 41, 25)$,
– x^2 with $x_{13} = x_{22} = x_{31} = 1$ and objective point $y^2 = (38, 33, 27)$,
– x^3 with $x_{11} = x_{22} = x_{33} = 1$ and objective point $y^3 = (39, 31, 30)$,
– x^4 with $x_{11} = x_{23} = x_{32} = 1$ and objective point $y^4 = (30, 38, 37)$.
Points y^1, y^2 and y^3 are supported, and y^4 is nonsupported. Only one maximal face of conv Y of dimension 2 is defined by y^1, y^2 and y^3. The maximal entries of all supported nondominated points yield $y^{N'} = (39, 41, 30)$ and not the nadir point $y^N = (39, 41, 37)$.

Other heuristics to determine the nadir point may underestimate y^N and are not useful to limit the search area in an exact two phase method.

- The nadir point is usually "far away" from the nondominated points. To limit the exploration in the bi-objective case, it is usual to use local nadir points. For any set $Y_P \subset Y$ a local nadir point y^N_P is defined by $y^N_{Pk} := \max\{y_k : y \in Y_P\}$ for $k = 1, \ldots, p$. In the bi-objective case using $Y_P = \{y^i, y^{i+1}\}$ for $i = 1, \ldots, r-1$ yields a search area because

$$Y_N \subset \left(\operatorname{conv} Y_{SN} + \mathbb{R}^2_{\geqq}\right) \cap \bigcup_{i=1}^{r-1} \left((y_1^{i+1}, y_2^i) - \mathbb{R}^2_{\geqq}\right). \tag{1}$$

As Example 1 shows, we cannot use extreme points of facets of conv Y_{SN} for $p \geqq 3$ analogously, even if conv Y_{SN} has only a single nondominated facet.

3. The search area

Since we cannot use local nadir points to limit the search area we use a set of points defined globally.

Definition 3 (*[14]*).

1. A lower bound set L for Y_N is an $\mathbb{R}^p_{\geq}$-closed and $\mathbb{R}^p_{\geqq}$-bounded set $L \subset \mathbb{R}^p$ such that $Y_N \subset L + \mathbb{R}^p_{\geqq}$ and $L \subset (L + \mathbb{R}^p_{\geqq})_N$.
2. An upper bound set U for Y_N is an $\mathbb{R}^p_{\geq}$-closed and $\mathbb{R}^p_{\geqq}$-bounded set $U \subset \mathbb{R}^p$ such that $Y_N \subset cl[(U+\mathbb{R}^p_{\geqq})^c]$ and $U \subset (U+\mathbb{R}^p_{\geqq})_N$.

Ehrgott and Gandibleux [14] show that $L = (\operatorname{conv} Y_{SN})_N$ is a lower bound set for Y_N. Any set of feasible points U that does not contain y^i, y^j with $y^i \leq y^j$ or $y^j \leq y^i$ is an upper bound set for Y_N, in particular $U = Y_{SN}$ is an upper bound set. Both L and U are obtained in Phase 1. Consequently, the search area can be defined by

$$\left(L + \mathbb{R}^p_{\geqq}\right) \setminus \left(U + \mathbb{R}^p_{>}\right). \tag{2}$$

However, this description is difficult to use in algorithms. In the bi-objective case the search area (2) consists of triangles with corner points y^i, y^{i+1} and (y_1^{i+1}, y_2^i), $i = 1, \ldots, r-1$, i.e. it is equivalently described by (1). This description of the search area using local nadir points is more convenient than (2) for two reasons. The search area is not explored globally but with several local explorations (of a triangle), and the partition of the search area clearly appears in (1). Moreover, the local nadir points are also used to compute upper bound values in each local exploration.

Next we prove an equivalent description of the search area (2) in a multi-objective context. To do this, we determine a set of points $D(U)$ such that $(L + \mathbb{R}^p_{\geqq}) \setminus (U + \mathbb{R}^p_{>}) = (L + \mathbb{R}^p_{\geqq}) \cap (D(U) - \mathbb{R}^p_{\geqq})$. The set $D(U)$ plays the same role in a multi-objective context as the local nadir points in the bi-objective context.

Definition 4. Let $U \subset Y$ be a set of feasible points that does not contain y^1, y^2 with $y^1 \leq y^2$ or $y^2 \leq y^1$. Let $D(U) \subset \mathbb{R}^p$ be the set of points of maximal cardinality that satisfy

(i) for all $\epsilon \in \mathbb{R}^p_{>}$, $u - \epsilon$ is not dominated by any point in U and
(ii) there does not exist $v \in \mathbb{R}^p$ satisfying condition (i) with $u \leq v$.

An illustration of conditions (i)–(ii) is given in Fig. 1. Obviously, in the bi-objective case the set $D(U)$ is the set of local nadir points. Proposition 1 shows that $D(U)$ provides the desired description of the search area.

Proposition 1. $\left(L + \mathbb{R}^p_{\geqq}\right) \setminus \left(U + \mathbb{R}^p_{>}\right) = \left(L + \mathbb{R}^p_{\geqq}\right) \cap \bigcup_{u \in D(U)} \left(u - \mathbb{R}^p_{\geqq}\right)$.

Proof. Suppose $y \notin U + \mathbb{R}^p_{>}$. Then for all $\epsilon \in \mathbb{R}^p_{>}$, $y - \epsilon$ is not dominated by any point in U, i.e. y satisfies condition (i) of Definition 4. Thus, either $y \in D(U)$ or there exists $v \in D(U)$ with $y \leq v$. In both cases $y \in \bigcup_{u \in D(U)} (u - \mathbb{R}^p_{\geqq})$.

Suppose there exists $y \in \bigcup_{u \in D(U)} (u - \mathbb{R}^p_{\geqq})$ and that $u < y$ for some $u \in U$. Then y does not satisfy condition (i) of Definition 4. This implies no $v \in \mathbb{R}^p$ with $y \leq v$ satisfies this condition. In particular, there exists $u \in D(U)$ with $y \in (u - \mathbb{R}^p_{\geqq})$. This gives a contradiction because $u \in D(U)$ and u does not satisfy condition (i) of Definition 4. Therefore the assumption that there exists $u \in U$ with $u < y$ is false and $y \notin U + \mathbb{R}^p_{>}$. □

In the following, we propose a procedure to compute $D(U)$. Suppose $D(Q)$ is known for some $Q \subset U$. For $y \in U \setminus Q$ we need to determine $D(Q \cup \{y\})$. Initially, $D(\emptyset) = \{(\infty, \ldots, \infty)\}$ or $D(\emptyset) = \{y^N\}$.

Lemma 1. *Let $Q \subset U \subset Y$, where U does not contain two points y^1, y^2 with $y^1 \leq y^2$ or $y^2 \leq y^1$. Let $y \in U \setminus Q$ and let $W = \{u \in D(Q) : y < u\}$. Then*

(i) *for all $u \in D(Q) \setminus W$, $u \in D(Q \cup \{y\})$;*
(ii) *for all $u \in W$, $u \notin D(Q \cup \{y\})$;*
(iii) *for all $u \in W$ let $u^k = (u_1, \ldots, u_{k-1}, y_k, u_{k+1}, \ldots, u_p)$, $k = 1, \ldots, p$. Then for all $k \in \{1, \ldots, p\}$, u^k satisfies condition (i) of Definition 4 and there does not exist $v \in (u - \mathbb{R}^p_{\geqq})$ satisfying condition (i) of Definition 4 with $u^i \leq v$ (see Fig. 2 for an illustration with three objectives).*

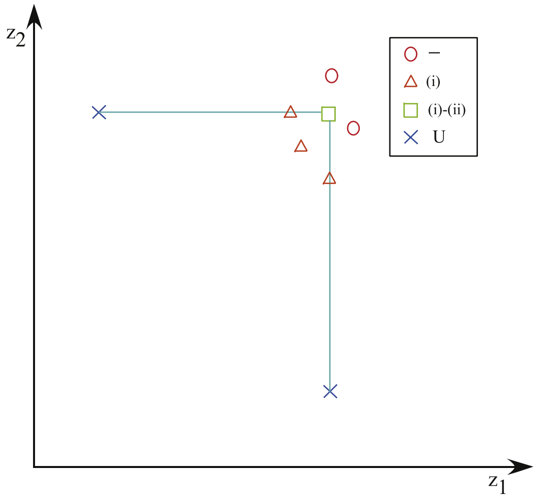

Fig. 1. Illustration of Conditions (i)–(ii) of Definition 4.

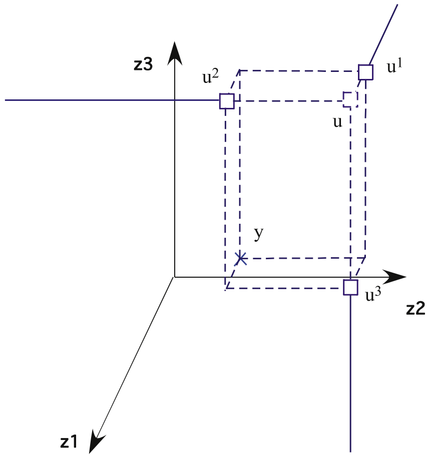

Fig. 2. $y < u$ and u^1, u^2 and u^3 are the only points that satisfy condition (i) and that may verify condition (ii) of Definition 4 in $(u - \mathbb{R}^3_{\geqq})$.

Proof. (i) Let $u \in D(Q) \setminus W$. The result follows directly from the fact that $u - \epsilon$ is not dominated by y, for any $\epsilon \in \mathbb{R}^p_{>}$.

(ii) Let $u \in W$ and $\epsilon = \frac{1}{2}(\min_{k\in\{1,\dots,p\}}(u_k - y_k))\sum_{k=1}^{p} e^k$. Then $u - \epsilon$ is dominated by y, so $u \notin D(Q \cup \{y\})$.

(iii) Let $u \in W$ and u^k, $k = 1, \dots, p$, defined as above. Then for all $\epsilon \in \mathbb{R}^p_{>}$, $u^k - \epsilon = u - (u_k - y_k)e^k - \epsilon$ is not dominated by y, so it is not dominated by any point of $Q \cup \{y\}$. Therefore u^k satisfies condition (i) of Definition 4. Let $k, j \in \{1, \dots, p\}$ and $\alpha \in \mathbb{R}_{>}$, let $v^j = u^k + \alpha e^j$ then if $j = k$, $v^j - \sum_{i=1}^{p} \frac{\alpha}{2} e^i$ is dominated by y, and if $j \neq k$, $v^j \notin (u - \mathbb{R}^p_{\geqq})$. □

We note that for all $u \in D(Q)$ such that $y \leq u$ without having $y < u$ it holds that $u \in D(Q \cup \{y\})$. Indeed, for all $\epsilon \in \mathbb{R}^p_{>}$, $u - \epsilon$ is not dominated by y and u still verifies condition (i) of Definition 4.

Proposition 2. *Let $Q \subset U$, where U is a set of feasible points that does not contain two points y^1, y^2 with $y^1 \leq y^2$ or $y^2 \leq y^1$. Let $y \in U \setminus Q$ and let W defined as in Lemma 1. Then*

$$D(Q \cup \{y\}) = (D(Q) \setminus W) \cup \bigcup_{u\in W} \bigcup_{k=1}^{p} \{u^k : \text{ there is no } v \neq u \in D(Q) \text{ with } u^k \leq v\}$$

where $u^k = (u_1, \dots, u_{k-1}, y_k, u_{k+1}, \dots, u_p)$ for $k = 1, \dots, p$.

Proof. Using (i) of Lemma 1 implies that $(D(Q) \setminus W) \subset D(Q \cup \{y\})$. Using (ii)–(iii) of Lemma 1 implies that $(D(Q \cup \{y\}) \setminus (D(Q) \setminus W)) \subseteq \bigcup_{u\in W} \bigcup_{k=1}^{p} \{u^k\}$. It remains to determine which points of $\bigcup_{u\in W} \bigcup_{k=1}^{p} \{u^k\}$ satisfy condition (ii) of Definition 4.

Let $u \in W$ and $k \in \{1, \dots, p\}$. Suppose that there exists $v \in D(Q)$ such that $u^k \leq v$. There are two cases because either $v \in D(Q \cup \{y\})$ ($v \notin W$) or not ($v \in W$). In the former case it is obvious that u^k does not satisfy condition (ii) of Definition 4. In the latter case v yields new points v^j, $j = 1, \dots, p$ and Lemma 1 implies that these points also verify condition (i) of

Definition 4. Since $u^k \leq v^k$, u^k does not verify condition (ii) of Definition 4. We suppose now that there does not exists $v \in D(Q)$ such that $u^k \leq v$. Then there are no points $v \in D(Q \cup \{y\})$ such that $u^k \leq v$, and u^k verifies condition (ii) of Definition 4. □

Note that if we know the ideal point $y^I = (y^I_1, \ldots, y^I_p)$ it is clear that for any point of $D(Q)$ with an entry identical to y^I the only feasible points in $(u - \mathbb{R}^p_{\geqq})$ are points minimizing one objective and are obtained in Phase 1. Such points are not useful in the description of the search area. In particular, the ideal point is given by the supported nondominated points [13].

Proposition 2 provides the update procedure of Algorithm 2. Considering the points $u \in D(Q)$ one by one, we test if $y < u$. This implies that $u \in W$, and that we must consider the points $\{u^1, \ldots, u^p\}$. If u^k has no identical entry with the ideal point and there is no $v \neq u \in D(Q)$ with $u^k \leq v$ then u^k is added to N, the set of new points of $D(Q \cup \{y\})$. By application of Proposition 2, we have $D(Q \cup \{y\}) = (D(Q) \setminus W) \cup N$.

Algorithm 1 procedure `computeUBS`

Parameters ↓ : A set of feasible points U, the nadir point y^N, the ideal point y^I
Parameters ↑ : The set $D(U)$

```
--| Initialization of the upper bound set
--| Q is a subset of U
Q ← ∅
D(Q) ← {y^N}
--| Computation of D(U) by considering the points of U one by one
for all y in U do
  updateUBS(y ↓, y^I ↓, D(Q) ↓, D(Q ∪ {y}) ↑)
  Q ← Q ∪ {y}
end for
```

Comment. In the algorithms, the symbols ↓, ↑ and ↕ specify the transmission mode of a parameter to a procedure; they correspond respectively to the mode IN, OUT and IN OUT. The symbol --| marks the beginning of a comment line.

Algorithm 2 procedure `updateUBS`

Parameters ↓ : A feasible point y, a set $D(Q)$, the ideal point y^I
Parameters ↑ : Updated set $D(Q \cup \{y\})$

```
--| In the following, W is the set of points in D(Q) \ D(Q ∪ {y})
--| N is the set of points in D(Q ∪ {y}) \ D(Q)
W ← ∅
N ← ∅
--| Comparison of y with each point u in D(Q)
for all u in D(Q) do
  --| Application of Proposition 2
  if y < u then
    W ← W ∪ {u}
    --| In the following u^k = (u_1, ..., u_{k-1}, y_k, u_{k+1}, ..., u_p)
    for k = 1 to p do
      if (y_k = y^I_k) or (∃v ≠ u ∈ D(Q) such that u^k ≤ v) then
        --| nothing to do here
      else
        N ← N ∪ {u^k}
      end if
    end for
  end if
end for
D(Q ∪ {y}) ← (D(Q) \ W) ∪ N
```

After Phase 1 we know all supported nondominated points. Since Y_{SN} is an upper bound set, we can compute $D(Y_{SN})$ to define the search area in objective space. Note that if we do not know the nadir point some points of $D(Y_{SN})$ will necessary have an undetermined entry ∞. In fact, it is possible to initialize the procedure with $D(\emptyset) = \{(\infty, \ldots, \infty)\}$ instead of $D(\emptyset) = \{y^N\}$.

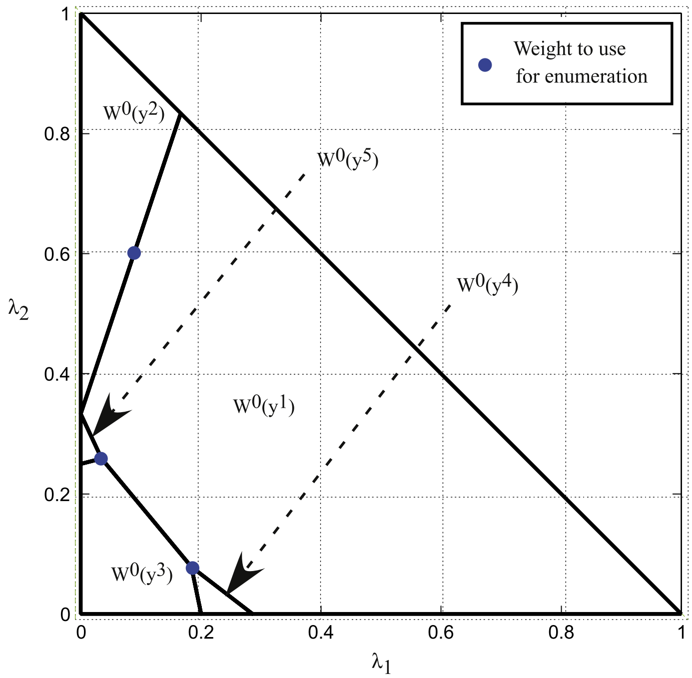

Fig. 3. Partition of W^0 defined by nondominated extreme points.

Example 2. Let us consider the three-objective assignment problem with the cost matrices

$$C^1 = \begin{pmatrix} 2 & 5 & 4 & 7 \\ 3 & 3 & 5 & 7 \\ 3 & 8 & 4 & 2 \\ 6 & 5 & 2 & 5 \end{pmatrix}, \quad C^2 = \begin{pmatrix} 3 & 3 & 6 & 2 \\ 5 & 3 & 7 & 3 \\ 5 & 2 & 7 & 4 \\ 4 & 6 & 3 & 5 \end{pmatrix} \text{ and } C^3 = \begin{pmatrix} 4 & 2 & 5 & 3 \\ 5 & 3 & 4 & 3 \\ 4 & 3 & 5 & 2 \\ 6 & 4 & 7 & 3 \end{pmatrix}.$$

With the algorithm of Przybylski et al. [12] we compute the set of nondominated extreme points and the faces of conv Y they define using the partition of the weight set. The supported efficient solutions are

- x^1 with $x_{11} = x_{22} = x_{34} = x_{43} = 1$ and objective point $y^1 = (9, 13, 16)$.
- x^2 with $x_{11} = x_{24} = x_{32} = x_{43} = 1$ and objective point $y^2 = (19, 11, 17)$.
- x^3 with $x_{12} = x_{23} = x_{31} = x_{44} = 1$ and objective point $y^3 = (18, 20, 13)$.
- x^4 with $x_{11} = x_{23} = x_{34} = x_{42} = 1$ and objective point $y^4 = (14, 20, 14)$.
- x^5 with $x_{11} = x_{23} = x_{32} = x_{44} = 1$ and objective point $y^5 = (20, 17, 14)$.

Fig. 3 shows the partition of the weight set obtained with these points.

The weight set decomposition allows to find appropriate weights for the enumeration of non-extremal supported nondominated points [12]. It also determines faces of conv Y where these points may exist. In this example no non-extremal supported nondominated point is found. The set conv Y has three nondominated faces.

- The face of dimension 2 given by the face $\{(\frac{1}{29}, \frac{15}{58})\}$ of dimension 0 in the weight set and defined by the extreme points $(9, 13, 16)$, $(20, 17, 14)$ and $(18, 20, 13)$ in the outcome set.
- The face of dimension 2 given by the face $\{(\frac{7}{38}, \frac{3}{38})\}$ of dimension 0 in the weight set and defined by the extreme points $(9, 13, 16)$, $(18, 20, 13)$ and $(14, 20, 14)$ in the outcome set.
- The face of dimension 1 given by the face of dimension 1 in the weight set the middle of which is $(\frac{1}{12}, \frac{7}{12})$ and which is defined by the extreme points $(9, 13, 16)$ and $(19, 11, 17)$ in the outcome set.

By complete enumeration of all feasible solutions we can obtain all nonsupported efficient solutions for this small example. They are

- x^6 with $x_{11} = x_{22} = x_{33} = x_{44} = 1$ and objective point $y^6 = (14, 18, 15)$.
- x^7 with $x_{12} = x_{23} = x_{34} = x_{41} = 1$ and objective point $y^7 = (18, 18, 14)$.

Thus, the nadir point is $y^N = (20, 20, 17)$, however, in practice, we do not know it.

Initially, $D(Q)$ is given by $\{(\infty, \infty, \infty)\}$ and $Q = \emptyset$. To compute $D(Y_{SN})$, we will add the points of Y_{SN} to Q one by one. From Y_{SN} we get the ideal point $y^I = (9, 11, 13)$.

- We consider the supported point $\begin{pmatrix}19\\11\\17\end{pmatrix}$. Since $\begin{pmatrix}19\\11\\17\end{pmatrix} < \begin{pmatrix}\infty\\\infty\\\infty\end{pmatrix}$ we get $W = \left\{\begin{pmatrix}\infty\\\infty\\\infty\end{pmatrix}\right\}$. Considering the points $\begin{pmatrix}19\\\infty\\\infty\end{pmatrix}$, $\begin{pmatrix}\infty\\11\\\infty\end{pmatrix}$, $\begin{pmatrix}\infty\\\infty\\17\end{pmatrix}$ we see that $\begin{pmatrix}\infty\\11\\\infty\end{pmatrix}$ has an identical entry with the ideal point, so
$$D\left(Q \cup \left\{\begin{pmatrix}19\\11\\17\end{pmatrix}\right\}\right) = \left(D(Q) \setminus \left\{\begin{pmatrix}\infty\\\infty\\\infty\end{pmatrix}\right\}\right) \cup \left\{\begin{pmatrix}19\\\infty\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}\right\}.$$
We set $Q \leftarrow Q \cup \left\{\begin{pmatrix}19\\11\\17\end{pmatrix}\right\}$ and obtain $D(Q) = \left\{\begin{pmatrix}19\\\infty\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}\right\}$.
- We consider the supported point $\begin{pmatrix}9\\13\\16\end{pmatrix}$. Since $\begin{pmatrix}9\\13\\16\end{pmatrix} < \begin{pmatrix}19\\\infty\\\infty\end{pmatrix}$ and $\begin{pmatrix}9\\13\\16\end{pmatrix} < \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}$ we have $W = \left\{\begin{pmatrix}19\\\infty\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}\right\}$. Thus we consider the points $\begin{pmatrix}9\\\infty\\\infty\end{pmatrix}$, $\begin{pmatrix}19\\13\\\infty\end{pmatrix}$, $\begin{pmatrix}19\\\infty\\16\end{pmatrix}$, and $\begin{pmatrix}9\\\infty\\17\end{pmatrix}$, $\begin{pmatrix}\infty\\13\\17\end{pmatrix}$, $\begin{pmatrix}\infty\\\infty\\16\end{pmatrix}$. Because $\begin{pmatrix}19\\\infty\\16\end{pmatrix} \le \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}$, $\begin{pmatrix}9\\\infty\\\infty\end{pmatrix}$ and $\begin{pmatrix}9\\\infty\\17\end{pmatrix}$ have an identical entry with the ideal point we obtain
$$D\left(Q \cup \left\{\begin{pmatrix}9\\13\\16\end{pmatrix}\right\}\right) = \left(D(Q) \setminus \left\{\begin{pmatrix}19\\\infty\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\17\end{pmatrix}\right\}\right) \cup \left\{\begin{pmatrix}19\\13\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\13\\17\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\16\end{pmatrix}\right\}.$$
We set $Q \leftarrow Q \cup \left\{\begin{pmatrix}9\\13\\16\end{pmatrix}\right\}$ so that $D(Q) = \left\{\begin{pmatrix}19\\13\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\13\\17\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\16\end{pmatrix}\right\}$.
- Considering the supported point $\begin{pmatrix}20\\17\\14\end{pmatrix}$ we have $\begin{pmatrix}20\\17\\14\end{pmatrix} < \begin{pmatrix}\infty\\\infty\\16\end{pmatrix}$ and $W = \left\{\begin{pmatrix}\infty\\\infty\\16\end{pmatrix}\right\}$. We consider the points $\begin{pmatrix}20\\\infty\\16\end{pmatrix}$, $\begin{pmatrix}\infty\\17\\16\end{pmatrix}$, $\begin{pmatrix}\infty\\\infty\\14\end{pmatrix}$ which leads to no deletion. Thus
$$D\left(Q \cup \left\{\begin{pmatrix}20\\17\\14\end{pmatrix}\right\}\right) = \left(D(Q) \setminus \left\{\begin{pmatrix}\infty\\\infty\\16\end{pmatrix}\right\}\right) \cup \left\{\begin{pmatrix}20\\\infty\\16\end{pmatrix}, \begin{pmatrix}\infty\\17\\16\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\14\end{pmatrix}\right\}.$$
With $Q \leftarrow Q \cup \left\{\begin{pmatrix}20\\17\\14\end{pmatrix}\right\}$ we get $D(Q) = \left\{\begin{pmatrix}19\\13\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\13\\17\end{pmatrix}, \begin{pmatrix}20\\\infty\\16\end{pmatrix}, \begin{pmatrix}\infty\\17\\16\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\14\end{pmatrix}\right\}$.
- Considering the supported point $\begin{pmatrix}18\\20\\13\end{pmatrix}$ we see that $\begin{pmatrix}18\\20\\13\end{pmatrix} < \begin{pmatrix}20\\\infty\\16\end{pmatrix}$ and $\begin{pmatrix}18\\20\\13\end{pmatrix} < \begin{pmatrix}\infty\\\infty\\14\end{pmatrix}$. Thus $W \leftarrow \left\{\begin{pmatrix}20\\\infty\\16\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\14\end{pmatrix}\right\}$. Next we consider the points $\begin{pmatrix}18\\\infty\\16\end{pmatrix}$, $\begin{pmatrix}20\\20\\16\end{pmatrix}$, $\begin{pmatrix}20\\\infty\\13\end{pmatrix}$, and $\begin{pmatrix}18\\\infty\\14\end{pmatrix}$, $\begin{pmatrix}\infty\\20\\14\end{pmatrix}$, $\begin{pmatrix}\infty\\\infty\\13\end{pmatrix}$. Since $\begin{pmatrix}18\\\infty\\14\end{pmatrix} \le \begin{pmatrix}20\\\infty\\16\end{pmatrix}$, $\begin{pmatrix}20\\\infty\\13\end{pmatrix}$ and $\begin{pmatrix}\infty\\\infty\\13\end{pmatrix}$ have an identical entry with the ideal point
$$D\left(Q \cup \left\{\begin{pmatrix}18\\20\\13\end{pmatrix}\right\}\right) = \left(D(Q) \setminus \left\{\begin{pmatrix}20\\\infty\\16\end{pmatrix}, \begin{pmatrix}\infty\\\infty\\14\end{pmatrix}\right\}\right) \cup \left\{\begin{pmatrix}18\\\infty\\16\end{pmatrix}, \begin{pmatrix}20\\20\\16\end{pmatrix}, \begin{pmatrix}\infty\\20\\14\end{pmatrix}\right\}$$
$Q \leftarrow Q \cup \left\{\begin{pmatrix}18\\20\\13\end{pmatrix}\right\}$ yields $D(Q) = \left\{\begin{pmatrix}19\\13\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\13\\17\end{pmatrix}, \begin{pmatrix}\infty\\17\\16\end{pmatrix}, \begin{pmatrix}18\\\infty\\16\end{pmatrix}, \begin{pmatrix}20\\20\\16\end{pmatrix}, \begin{pmatrix}\infty\\20\\14\end{pmatrix}\right\}$.
- For the supported point $\begin{pmatrix}14\\20\\14\end{pmatrix}$, we have $\begin{pmatrix}14\\20\\14\end{pmatrix} < \begin{pmatrix}18\\\infty\\16\end{pmatrix}$ so that $W = \left\{\begin{pmatrix}18\\\infty\\16\end{pmatrix}\right\}$. Thus, we consider $\begin{pmatrix}14\\\infty\\16\end{pmatrix}$, $\begin{pmatrix}18\\20\\16\end{pmatrix}$, $\begin{pmatrix}18\\\infty\\14\end{pmatrix}$. Since $\begin{pmatrix}18\\20\\16\end{pmatrix} \le \begin{pmatrix}20\\20\\16\end{pmatrix}$ we obtain
$$D\left(Q \cup \left\{\begin{pmatrix}14\\20\\14\end{pmatrix}\right\}\right) = \left(D(Q) \setminus \left\{\begin{pmatrix}18\\\infty\\16\end{pmatrix}\right\}\right) \cup \left\{\begin{pmatrix}14\\\infty\\16\end{pmatrix}, \begin{pmatrix}18\\\infty\\14\end{pmatrix}\right\}.$$
For $Q \leftarrow Q \cup \left\{\begin{pmatrix}14\\20\\14\end{pmatrix}\right\}$ we finally have $Q = Y_{SN}$ and
$$D(Y_{SN}) = \left\{\begin{pmatrix}19\\13\\\infty\end{pmatrix}, \begin{pmatrix}\infty\\13\\17\end{pmatrix}, \begin{pmatrix}\infty\\17\\16\end{pmatrix}, \begin{pmatrix}20\\20\\16\end{pmatrix}, \begin{pmatrix}\infty\\20\\14\end{pmatrix}, \begin{pmatrix}14\\\infty\\16\end{pmatrix}, \begin{pmatrix}18\\\infty\\14\end{pmatrix}\right\}.$$

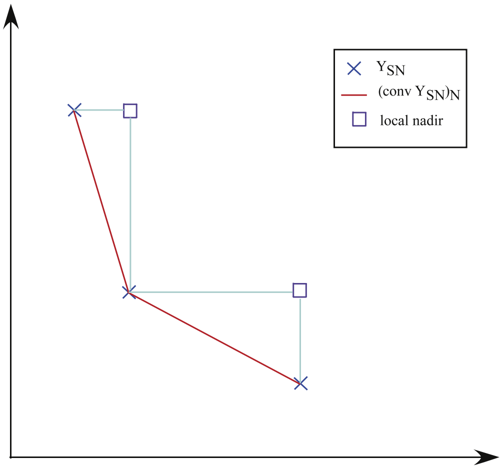

Fig. 4. The search area is naturally partitioned in the bi-objective case.

In case the nadir point is known the procedure of computing $D(Y_{SN})$ can be started with $D(\emptyset) = \{y^N\} = \left\{\begin{pmatrix}20\\20\\17\end{pmatrix}\right\}$. Then we get $D(Y_{SN}) = \left\{\begin{pmatrix}20\\13\\17\end{pmatrix}, \begin{pmatrix}20\\20\\16\end{pmatrix}\right\}$.

In the bi-objective case the search area (1) is the union of disjoint triangles. We shall use Example 2 to see that there is no analogy to this for problems with $p \geqq 3$ objectives. To simplify the discussion we shall assume that the nadir point is known.

Example 2 (*Continued*)**.** According to Proposition 1 the search area is

$$\left((\text{conv } Y_{SN})_N + \mathbb{R}^3_{\geqq}\right) \cap \left(D(Y_{SN}) - \mathbb{R}^3_{\geqq}\right).$$

Let F_1 be the facet of conv Y_{SN} defined by nondominated extreme points $(9, 13, 16)$, $(20, 17, 14)$ and $(18, 20, 13)$ and let F_2 be the facet defined by $(9, 13, 16)$, $(18, 20, 13)$ and $(14, 20, 14)$.

The intersection of $((20, 13, 17) - \mathbb{R}^3_{\geqq})$ with $((\text{conv } Y_{SN})_N + \mathbb{R}^3_{\geqq})$ has extreme points $(9, 13, 17)$, $(20, 11, 17)$, $(20, 13, \frac{634}{41})$. The first and the second point are not located on any face of $(\text{conv } Y_{SN})_N$ and the third point is located on face F_2.

The intersection of $((20, 20, 16) - \mathbb{R}^3_{\geqq})$ with $((\text{conv } Y_{SN})_N + \mathbb{R}^3_{\geqq})$ has extreme points $(9, 20, 16)$, $(20, \frac{173}{15}, 16)$ and $(20, 20, 13)$. The first and the third point are not located on any face of $(\text{conv } Y_{SN})_N$ and the second point is located on face F_2.

This shows that the easy description of the search area by triangles in the bi-objective case does not carry over to $p \geqq 3$ objectives. More importantly, $((20, 20, 16) - \mathbb{R}^3_{\geqq}) \cap ((\text{conv } Y_{SN})_N + \mathbb{R}^3_{\geqq})$ and $((20, 13, 17) - \mathbb{R}^3_{\geqq}) \cap ((\text{conv } Y_{SN})_N + \mathbb{R}^3_{\geqq})$ have a non-empty intersection. Therefore, redundant exploration is difficult to avoid.

Intuitively, the search area is naturally partitioned in the bi-objective case because of the natural order of nondominated points (Fig. 4). This is no longer true with more than two objectives because local nadir points cannot be used to describe the search area, which is larger (Fig. 5).

In the general case, where the nadir point is unknown, Example 2 suggests that most of the points of $D(Y_{SN})$ have infinite entries. However, experiments have shown that this is only true for small instances. We show that no point in $D(U)$ has more than $p - 2$ infinite entries.

Proposition 3. *Suppose that for each objective U contains one point which minimizes it. Then there is no point in $D(U)$ with more than $p - 2$ infinite entries.*

Proof. Suppose that $D(U)$ contains a point with only one entry that is different from ∞. Without loss of generality this point is $(u_1, \infty, \ldots, \infty)$. Let $y^1 \in U$ be a point that minimizes the first objective. Either $y^1 < (u_1, \infty, \ldots, \infty)$ or not. In the former case Proposition 2 implies that $(u_1, \infty, \ldots, \infty)$ is replaced by one of the points $(u_1, y^1_2, \infty, \ldots, \infty), \ldots, (u_1, \infty, \ldots, \infty, y^1_p)$. In the latter case u_1 is also the first entry in the ideal point and $(u_1, \infty, \ldots, \infty)$ is deleted from $D(U)$. □

Of course, Proposition 3 implies that for a problem with three objectives $D(Y_{SN})$ does not contain a point with more than one infinite entry. Next, we analyse the cause of the infinite entries.

Let us consider a problem P with p objectives and let $U \subset Y$ be a set of feasible points that does not contain two points y^1, y^2 with $y^1 \leq y^2$ or $y^2 \leq y^1$. Let $I \subset \{1, \ldots, p\}$ with $|I| \geqq 2$ and let P_I be the problem defined by the objectives indexed by I. We define the following sets:

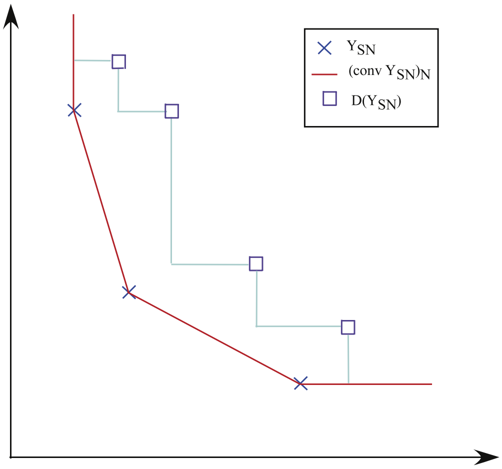

Fig. 5. The search area with more than two objectives.

- $U_I \subset \mathbb{R}^{|I|}$ is the set of points obtained by projecting U to $\mathbb{R}^{|I|}$, i.e. entries the index of which are not in I are deleted.
- $\bar{U}_I$ is the nondominated subset of U_I in $\mathbb{R}^{|I|}$.
- $U_I^P \subset U$ is the set of points obtained from the points of $\bar{U}_I$ by restoring the original entries for the objectives in $\{1, \dots, p\} \setminus I$.
- $D^P(\bar{U}_I) \subset \mathbb{R}^p$ is the set defined by the points of $D(\bar{U}_I) \subset \mathbb{R}^{|I|}$ with infinite entries for all objectives in $\{1, \dots, p\} \setminus I$.

Lemma 2 below shows the relationship between $D^P(\bar{U}_I)$ and $D(U)$.

Lemma 2. $D^P(\bar{U}_I) \subset D(U)$.

Proof. Let $u \in D^P(\bar{U}_I)$. Then for all $\epsilon \in \mathbb{R}^p_>$, $u - \epsilon$ is not dominated by any point of U_I^P. Suppose that there exists $\epsilon \in \mathbb{R}^p_>$ with $u - \epsilon$ dominated by a point v in $U \setminus U_I^P$. Then by deleting the entries of $u - \epsilon$ and v with index not in I we obtain points $(u - \epsilon)_I$ and v_I such that $v_I \leqq (u - \epsilon)_I$. This implies that v_I is not dominated by any point of $\bar{U}_I$ and thus $v_I \in \bar{U}_I$, which is a contradiction. So u satisfies condition (i) for $D(U)$. Condition (ii) is naturally satisfied since u is constructed from a subset of U and with infinite entries. $\square$

Lemma 2 implies the following propositions.

Proposition 4. *For all $I \subset \{1, \dots, p\}$ with $|I| \geqq 2$, there exists at least one point in $D(U)$ with finite entries in the components indexed by I and infinite entries otherwise.*

Proposition 5. *The objective points of problem P corresponding to all efficient solutions of P_I that are also efficient solutions of P are located in $\bigcup_{u \in D_I(U)} (u - \mathbb{R}^p_{\geqq})$ where $D_I(U) = \{u \in D(U) : u_i = \infty$ for all $i \in \{1, \dots, p\} \setminus I\}$.*

Example 2 (*Continued*)**.** We illustrate Propositions 4 and 5 on Example 2. There are three bi-objective problems with $I = \{1, 2\}$, $I = \{1, 3\}$ and $I = \{2, 3\}$, respectively.

- The objective points for problem P of the efficient solutions of problem $P_{\{1,2\}}$ are $(9, 13, 16)$ and $(19, 11, 17)$. Both points are located in $((19, 13, \infty) - \mathbb{R}^3_{\geqq})$.
- The objective points for problem P of the efficient solutions of problem $P_{\{1,3\}}$ are $(9, 13, 16)$, $(14, 20, 14)$ and $(18, 20, 13)$. All these points are located in $((14, \infty, 16) - \mathbb{R}^p_{\geqq}) \cup ((18, \infty, 14) - \mathbb{R}^p_{\geqq})$.
- The objective points for problem P of the efficient solutions of problem $P_{\{2,3\}}$ are $(9, 13, 16)$, $(20, 17, 14)$, $(19, 11, 17)$ and $(18, 20, 13)$. All these points are located in $((\infty, 13, 17) - \mathbb{R}^p_{\geqq}) \cup ((\infty, 17, 16) - \mathbb{R}^p_{\geqq}) \cup ((\infty, 20, 14) - \mathbb{R}^p_{\geqq})$.

In the next section, we will show that in order to explore the part of search area defined by points of $D(Y_{SN})$ with infinite entries, the solution of the p subproblems with $p - 1$ objectives is necessary. In that case, we can use the following fundamental result.

Theorem 1 ([13])**.** *Let Y_N^{p-1} denote the set of nondominated points of P given by all efficient solutions of the p problems $P_{\{1,\dots,p\}\setminus\{k\}}$ with $p - 1$ objectives. Then the nadir point $y^N = (y_1^N, \dots, y_p^N)$ is given by*

$$y_k^N = \max\{y_k : y \in Y_N^{p-1}\}, \quad k \in \{1, \dots, p\}.$$

4. Exploration of the search area

Let $\bar{Y} = Y_{SN}$ such that $D(\bar{Y})$ describes the search area as in Proposition 1. In particular $Y_N \subset \bigcup_{u \in D(\bar{Y})}(u - \mathbb{R}^p_{\geqq})$. In the following let UP be the set of points delimiting the part of the search area that remains to be explored, i.e., $\bigcup_{u \in UP}(u - \mathbb{R}^p_{\geqq})$. Initially $UP = D(\bar{Y})$. Next we explain how to use the facets of $(\operatorname{conv} Y_{SN})_N$ to reduce the search area.

We compute the distance between all points of UP that do not have an infinite entry and all hyperplanes corresponding to facets of $(\operatorname{conv} Y_{SN})_N$. The points with an infinite entry will be dealt with separately. Recall that the distance between a point u and a hyperplane h is

$$\operatorname{dist}(u, h) = \frac{\lambda_1 u_1 + \lambda_2 u_2 + \cdots + \lambda_p u_p + \alpha}{\sqrt{\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_p^2}},$$

where $u = (u_1, u_2, \ldots, u_p)$ and $\lambda_1 z_1 + \lambda_2 z_2 + \cdots + \lambda_p z_p + \alpha = 0$ is the equation of the hyperplane h.

For each point u of UP we must explore $(u - \mathbb{R}^p_{\geqq})$. Analogous to the bi-objective case described in [3] we propose to explore a band between a facet of $(\operatorname{conv} Y_{SN})_N$ and the point u. However, the association of u and a facet of $(\operatorname{conv} Y_{SN})_N$ is not as obvious as in the bi-objective case. Since in the worst case it will be necessary to explore the complete band we propose to use the slimmest band, i.e. we associate u with the facet of $(\operatorname{conv} Y_{SN})_N$ which is closest to the point u. The band is defined by the hyperplane containing this facet and the parallel hyperplane containing u.

Let H be the set of hyperplanes given by the facets of $(\operatorname{conv} Y_{SN})_N$. For each $u \in UP$ let $h(u)$ be the closest hyperplane to u. Let $H_p = \{h \in H : h = h(u) \text{ for some } u \in UP\}$ be the set of hyperplanes selected for potential exploration. It is not necessarily true that $H = H_p$ and there may be $u \neq v$ with $h(u) = h(v)$. For all $h \in H_p$ let $UP(h) = \{u \in UP : h(u) = h\}$ and $Val(h) = \max\{dist(u, h) : u \in UP(h)\}$. The hyperplane $h^* \in H_p$ with $\min_{h \in H_p} Val(h) = Val(h^*)$ is chosen for the exploration of $\bigcup_{u \in UP(h^*)}(u - \mathbb{R}^p_{\geqq})$, using the band defined by h^* and the parallel hyperplane containing the farthest point from h^* in $UP(h^*)$. Once the exploration is complete we update UP to $UP \setminus UP(h)$ and h^* is deleted from the list of hyperplanes H. The procedure of assigning points in UP to hyperplanes in H is summarized in Algorithm 3.

Algorithm 3 procedure `ChooseWeightAndPoint`

Parameters $\downarrow : H, UP$
Parameters $\uparrow : h^*, UP(h^*)$

$H_p \leftarrow \emptyset$
for all $h \in H$ **do**
  $UP(h) \leftarrow \emptyset$
end for
for all u in UP **do**
  for all h in H **do**
    `ComputeDistPointHyperplane`$(u \downarrow, h \downarrow, dist(u, h) \uparrow)$
  end for
  $h' \leftarrow \arg(\min_{h \in H} dist(u, h))$
  $UP(h') \leftarrow UP(h') \cup \{u\}$
  $H_p \leftarrow H_p \cup \{h'\}$
end for
$h^* \leftarrow \arg(\min_{h \in Hp} \max_{u \in UP(h)} dist(u, h))$

During the exploration some new nondominated points Y_{new} can be found. This implies an update $\bar{Y} \leftarrow \bar{Y} \cup Y_{new}$ of $\bar{Y}$ and consequently an update of UP using Y_{new} (Algorithm 2). This exploration and update process is iterated until $UP = \emptyset$ implying that $\bar{Y} = Y_N$.

The proposed process for the exploration is a generalization of the exploration by triangles in the bi-objective case: the area under each local nadir point is explored using the nearest line, which is the hypotenuse of the associated triangle. The main difference for $p > 2$ is that due to the lack of a natural order of nondominated points an exploration can reveal feasible points that are useful to reduce the search area. This does not happen in the bi-objective case, because a feasible point of a triangle is never a feasible point of another triangle.

The above procedure is not applicable to points of UP with an infinite entry since $dist(u, h)$ is not defined for such a u. However, Proposition 4 implies that there exist points in UP with 1 to $p - 2$ infinite entries. To deal with this, we use all subproblems with 2 to $p - 1$ objectives.

Lemma 3. *Let P be a problem with p objectives and let $P(k) = P_{\{1,\ldots,p\}\setminus\{k\}}$. Let x be an extremal supported efficient solution of $P(k)$. Then if x is an efficient solution of P it is also an extremal supported efficient solution of P.*

Proof. Let x be simultaneously extremal supported efficient solution of $P(k)$ and an efficient solution of P. Assume that x is not an extremal supported efficient solution of P, then there exist $a, b \in Y_N$ and $\alpha \in]0, 1[$ such that $y = z(x) = \alpha a + (1-\alpha)b$. Since x is an extremal supported efficient solution of $P(k)$, we have $y_i = a_i = b_i$ for all $i \in \{1, \ldots, p\} \setminus \{k\}$. Finally, a and b can differ only in the kth entry. Hence y is dominated, the assumption that x is not an extremal supported solution of P is thus false. □

By induction, we obtain the following proposition.

Proposition 6. *Let P be a problem with p objectives and let P_I be a subproblem given by objectives indexed by $I \subset \{1, \ldots, p\}$. Let x be an extremal supported efficient solution of P_I. Then if x is an efficient solution of P it is also an extremal supported efficient solution of P.*

Note that if x is a non-extremal supported efficient solution of P_I and an efficient solution of P then x is not necessarily a supported efficient solution of P.

Example 3. We consider an instance of the three-objective assignment problem defined by the cost matrices

$$C^1 = \begin{pmatrix} 1 & 50 & 1 \\ 50 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}, \qquad C^2 = \begin{pmatrix} 1 & 50 & 1 \\ 50 & 2 & 2 \\ 3 & 2 & 1 \end{pmatrix} \quad \text{and} \quad C^3 = \begin{pmatrix} 0 & 50 & 0 \\ 50 & 0 & 50 \\ 0 & 50 & 0 \end{pmatrix}.$$

If we only consider the bi-objective problem defined by C^1 and C^2, we have 3 supported solutions with the objective points (6, 4), (5, 5), (4, 6). By adding the third objective defined by C^3, these points become (6, 4, 0), (5, 5, 100), (4, 6, 0). It is easy to see that the point (5, 5, 100) is nonsupported.

Since initially $\bar{Y} = Y_{SN}$ Proposition 6 implies that all nondominated extreme points of all subproblems of P with 2 to $p-1$ objectives are known. Since the facets of $(\operatorname{conv} Y_{SN})_N$ of any multi-objective problem are defined by its nondominated extreme points, the facets of $(\operatorname{conv} Y_{SN})_N$ of all subproblems with 2 to $p-1$ objectives can be used. Thus, for points $u \in UP$ with infinite entries we can compute the shortest distance between u and the hyperplanes of appropriate subproblems in the same way as above. In particular for $p = 2$, we compute the shortest distance between the lines defined by "adjacent" nondominated extreme points of a bi-objective problem to determine a band defined by local nadir points. Consequently, the solution of all subproblems with 2 to $p-1$ objectives is required. In particular, the solution of the p subproblems with $p-1$ objectives is a necessary step of the algorithm. The total number of subproblems solved is hence $\sum_{q=2}^{p-1} \binom{p}{q}$, a numberthat grows quickly with p.

In order to avoid useless different procedures, we can therefore compute directly the nadir point using Theorem 1. After Phase 1, having obtained all supported nondominated points and hyperplanes describing $\operatorname{conv} Y + \mathbb{R}^p_{\geqq}$, we solve the p problems with $(p-1)$ objectives (in the sense of computing the maximal complete set). This yields some new efficient solutions and, using Theorem 1, the nadir point. The points of UP with infinite entries can now be eliminated and therefore Algorithm 3 can be used for assigning all points in UP to hyperplanes in H.

5. Upper bound values in the exploration

It is usual for Phase 2 algorithms in the bi-objective case to explore the complete band between the line defined by a facet of $(\operatorname{conv} Y_{SN})_N$ and a parallel line defined by $\{y \in \mathbb{R}^2 : \lambda^T y = \alpha\}$ where $\lambda \in \mathbb{R}^2_>$ is the normal vector of the facet and α is an upper bound value. We generalize this kind of upper bound value for the multi-objective case. The set UP defined in Section 4 will be used. An improvement of that upper bound value in the case where the costs are integer is also given, analogous to the bi-objective case [3].

Let $UP(h) = \{u^i : i \in I\}$, where I is an index set, denote the subset of points obtained by Algorithm 3 and h is the hyperplane (with normal vector λ) defined by the facet chosen for the exploration. Let h_α be the hyperplane defined by $\{y \in \mathbb{R}^p : \lambda^T y = \alpha\}$ for a given value $\alpha \in \mathbb{R}$.

A first upper bound value is directly given by the points of $UP(h)$. The goal of the considered exploration is to find all nondominated points in $\bigcup_{i \in I}(u^i - \mathbb{R}^p_{\geqq})$ so α must be chosen such that the band we explore between h and h_α is just large enough to contain all feasible points in $\bigcup_{i \in I}(u^i - \mathbb{R}^p_{\geqq})$. By definition of $\{u^i : i \in I\}$

$$\beta_0 := \max\{\lambda^T u^i : i \in I\}$$

is a valid upper bound value. Because all points $y \in Y$ contained in $\bigcup_{i \in I}(u^i - \mathbb{R}^p_{\geqq})$ with $\lambda^T u^i \geq \beta_0$ are dominated, enumeration of all $y \in Y$ with y between h and h_{β_0} yields all nonsupported nondominated points in $\bigcup_{i \in I}(u^i - \mathbb{R}^p_{\geqq})$. After all explorations, using this upper bound value we will find the maximum complete set X_{E_M}.

Let us now assume that all costs are integer, i.e. all feasible points $y \in Y$ have integer components. We first restrict the discussion to the three-objective case.

By parallel translation of h_{β_0} towards h until an integer point is reached the value of β_0 can be reduced. This is analogous to the bi-objective case [3]. Since not all points in the facets of $(u - \mathbb{R}^3_{\geqq})$ for $u \in U$ are dominated (Definition 4 implies that the points of $UP(h)$ are not strictly dominated), we cannot directly use a translation by $(1, 1, 1)$. We must also consider all facets of $(u - \mathbb{R}^3_{\geqq})$ to check which part of the facet is dominated. To do this for all facets of $(u - \mathbb{R}^3_{\geqq})$ we compute an upper bound value like in the bi-objective case.

The facets of $(u - \mathbb{R}^3_{\geqq})$ are $F_j(u) = \{y \in (u - \mathbb{R}^3_{\geqq}) : y_j = u_j\}$ for $j = 1, 2, 3$. Let $\{y^l : l \in \{1, \dots, s\}\} \subset F_j(u)$ be a set of feasible points that does not contain two points y^1, y^2 with $y^1 \leq y^2$ or $y^2 \leq y^1$. Without loss of generality we consider the facet $F_3(u)$. To compute the upper bound value for this facet let $\{y^l = (y^l_1, y^l_2, u_3) : l \in \{1, \dots, s\}\}$ be sorted by increasing order of the first objective. Local nadir points are defined by

$$n^0 = (y^1_1, u_2, u_3)$$
$$n^l = (y^{l+1}_1, y^l_2, u_3) \quad \text{for all } l \in \{1, \dots, s-1\}$$
$$n^s = (u_1, y^k_2, u_3)$$

and the upper bound of this face is computed like in the bi-objective case [3]. With

$$\gamma_1 = \max\{\lambda^T y^l : l = 1, \dots, s\}$$
$$\gamma_2 = \max\{\lambda^T (n^l - (1, 1, 0)) : l = 0, \dots, s\}$$

the distance to the farthest potentially nondominated point in the face $F_3(u)$ is given by

$$\rho(F_3(u)) = \max\{\gamma_1, \gamma_2\}.$$

If there is no feasible point in the facet $F_3(u)$ then $\rho(F_3(u)) = \lambda^T(u_1 - 1, u_2 - 1, u_3)$.

The distance to the farthest potentially nondominated point in $(u - \mathbb{R}^3_{\geqq})$ is therefore determined by

$$\alpha(u) = \max\{\rho(F_1(u)), \rho(F_2(u)), \rho(F_3(u)), \lambda^T(u - (1, 1, 1))\}.$$

The new upper bound value corresponding to the farthest potentially nondominated point of $\bigcup_{i \in I}(u^i - \mathbb{R}^3_{\geqq})$ is then

$$\beta_1 = \max\{\alpha(u^i) : i \in I\}. \tag{3}$$

All solutions $x \in X$, $z(x) \in \bigcup_{i \in I}(u^i - \mathbb{R}^3_{\geqq})$ with $\lambda^T u^i > \beta_1$ are dominated and enumeration of all $x \in X$ with $z(x)$ between h and h_{β_1} (included) yields all nonsupported efficient solutions with objective points in $\bigcup_{i \in I}(u^i - \mathbb{R}^3_{\geqq})$, including the equivalent ones. After all explorations, using this upper bound value, we will find the maximum complete set X_{E_M}.

Again, as in the bi-objective case upper bound β_1 can be improved if a complete set of efficient solutions is to be found rather than the maximal complete set. We can ignore the potentially nondominated points when computing the bound because we have already found some solutions for each of these points. The formula is the same as (3) except that $\rho(F_j(u)) = \gamma_2$. This changes the value of $\alpha(u)$. We denote this bound by β_2.

All solutions $x \in X$, $z(x) \in \bigcup_{i \in I}(u^i - \mathbb{R}^3_{\geqq})$ with $\lambda^T z > \beta_2$ are dominated by or equivalent to an efficient (or potentially efficient) solution we have already found. We find all efficient solutions in the band between h and h_{β_2} inclusive plus some others outside the band. After all explorations we will find a complete set.

For $p > 3$ the generalization of the upper bound value β_1 and β_2 is done recursively. In the same way as described above, we consider the facets of $(u^i - \mathbb{R}^p_{\geqq})$ for all $i \in I$. There are p facets to consider in which we compute an upper bound value like for a problem with $(p - 1)$ objectives. Since the procedure to compute β_1 and β_2 for the bi-objective case is known this procedure to compute upper bounds for the p objective case is well defined.

We are now able to present a general statement of our multi-objective two phase method (see Algorithm 4).

Step1 Compute all nondominated extreme points using the recursive algorithm of Przybylski et al. [12] (procedure `ComputeSupportedAndWeight`). Using the partition of the weight set find all supported efficient solutions and the facets of conv Y defined by the nondominated extreme points as well as the hyperplanes containing them. Set $\bar{Y} \leftarrow Y_{SN}$ and let H be the set of hyperplanes obtained.

Step2 Solve the p problems with $p-1$ objectives in the sense of finding the maximal complete set. Add the newly discovered nondominated points to $\bar{Y}$. The nadir point y^N is now also known.

Step3 Compute $D(\bar{Y})$ using the procedure described in Section 3 and summarized in Algorithms 1 and 2. Let UP be the set of points delimiting the remaining search area, i.e., $\bigcup_{u \in UP}(u - \mathbb{R}^p_{\geqq})$. Initialize $UP \leftarrow D(\bar{Y})$.

Step4 If $UP \neq \emptyset$ associate a hyperplane $h \in H$ with a set of points $\{u_1, \dots, u_q\} \subseteq UP$ to explore $(h + \mathbb{R}^3_{\geqq}) \cap (\bigcup_{i=1}^q (u_i - \mathbb{R}^3_{\geqq}))$ using the procedure described in Section 4 and summarized in Algorithm 3. Otherwise stop, $\bar{Y}$ is the set of nondominated points.

Step5 Problem specific exploration of $(h + \mathbb{R}^3_{\geqq}) \cap (\bigcup_{i=1}^q (u_i - \mathbb{R}^3_{\geqq}))$ (see Algorithm 5 described in Section 6 for the case of (3AP)) yielding a new subset of nondominated points Y_{new}.

Step6 Update $UP \leftarrow UP \setminus \{u^1, \dots, u^q\}$ using Y_{new} with Algorithm 2. Update $\bar{Y} \leftarrow \bar{Y} \cup Y_{new}$ and $H \leftarrow H \setminus \{h\}$. Goto Step 4.

As in the bi-objective case the exploration in Step 5 should be problem specific. It is easily possible to adapt a procedure designed for the bi-objective case.

Algorithm 4 procedure `entryPoint`

Parameters $\downarrow$: Cost Matrix $C^1, \ldots, C^p$
Parameters $\uparrow$: Complete Set S

--| Phase 1: Computation of all supported solutions and hyperplanes to be used in Phase 2
`ComputeSupportedAndWeight`$(C^1 \downarrow, \ldots, C^p \downarrow, S \uparrow, H \uparrow, y^I \uparrow)$
--| Solve the p $(p-1)$-objective problems to complete S and find the nadir point y^N
--| (we assume here that $p > 2$)
for $k := 1$ to p **do**
  `entrypoint`$(C^1 \downarrow, \ldots, C^{k-1} \downarrow, C^{k+1} \downarrow, \ldots, C^p \downarrow, S \updownarrow)$
end for
for $k := 1$ to p **do**
  $y^N_k \leftarrow \max\{y_k : y \in S\}$
end for
--| Computation of UP the initial set of points delimiting the search area
`computeUBS`$(z(S) \downarrow, y^N \downarrow, y^I \downarrow, UP \uparrow)$
--| Main Loop of the algorithm
while $UP \neq \emptyset$ **do**
  --| choose weight and points delimiting an exploration
  `ChooseWeigthAndPoints`$(H \downarrow, UP \downarrow, h^* \uparrow, UP(h^*) \uparrow)$
  `explor3AP`$(C^1 \downarrow, \ldots, C^p \downarrow, h^* \downarrow, UP(h^*) \downarrow, S \updownarrow, S_{new} \uparrow)$
  --| With the new feasible points, update UP
  for all y in $z(S_{new})$ **do**
    `updateUPS`$(y \downarrow, UP \updownarrow)$
  end for
  $H \leftarrow H \setminus \{h^*\}$
  $UP \leftarrow UP \setminus U(h^*)$
end while

6. Exploration for the assignment problem with three objectives

In this section we describe an exploration procedure for the three-objective assignment problem. In [3] we have demonstrated that a ranking algorithm outperforms all other known strategies. Ranking algorithms compute K-best solution of a single objective problem and have been used for bi-objective assignment problems in [15,3]. The best known complexity for a ranking algorithm for the assignment problem is $\mathcal{O}(Kn^3)$ and there are four algorithms in the literature with this complexity: [16–19]. However, the performance of these algorithms in practice is not identical. As shown in Pedersen et al. [19,20], the algorithms of Pascoal et al. [18], Pedersen et al. [19] and Miller et al. [17] perform a lot faster than the algorithm by Chegireddy and Hamacher [16], but their memory requirement is higher.

All methods store a set of candidate solutions for the next solution in the ranking. After each iteration, the element with lowest cost is picked from the candidate set as the next solution in the ranking. It is replaced by some other solutions. The algorithms by Pascoal et al. [18], Pedersen et al. [19] and Miller et al. [17] are derived from the general algorithm by Murty [21], in particular each solution picked from the candidate set is replaced by at most $n-1$ new solutions. The algorithm by Chegireddy and Hamacher [16] is an application of the binary search tree algorithm by Hamacher and Queyranne [22] and each solution picked from the candidates is replaced by 2 new solutions.

Algorithm 3 returns a hyperplane h (with associated normal vector λ) and a subset of points $UP(h) = \{u_i : i \in I\}$ delimiting the exploration to be done. The area we must explore is $B = (h + \mathbb{R}^3_{\geqq}) \cap (\bigcup_{u \in UPR}(u - \mathbb{R}^3_{\geqq}))$ where initially $UPR = UP(h)$. During the exploration we use information from the explored points as soon as they become available to reduce the necessary exploration in B by updating UPR and by updating upper bound values.

To explore B we search solutions with increasing value of $z^\lambda = \sum_{i=1}^{3} \lambda_i z^i$ until one of the upper bound values β_i (see Section 5) is reached. Any one of the existing ranking algorithms can be used.

Before the enumeration, we compute an initial upper bound value α as described in Section 5 using $\{u_i : i \in I\}$. During the execution of the ranking algorithm new solutions which are not dominated by a solution of S are added to S, the set of efficient solutions obtained in the exploration. Each time S is updated it is possible to update UPR with Algorithm 2 and/or the upper bound value α. The ranking algorithm stops as soon as a solution with $\lambda^T y = \beta_i$ is found.

Note that during the enumeration we may also find solutions with objective points located outside of B. These solutions are also added to S if they are not dominated but they do not trigger an update of the upper bound value or the set of points UPR.

From the CPU time requirement the ranking algorithm of choice is either Pascoal et al. [18], Pedersen et al. [19] or Miller et al. [17]. Since B is the slimmest possible band to explore the initial upper bound value is already rather small and will moreover be reduced after few iterations. Therefore, each time a new solution is added in the ranking we can test each new candidate solution whether $\lambda^T z(x) > \alpha$. If so, it is not necessary to add this solution. In practice this happens for most of the new candidate solutions. Consequently, the required memory of the ranking algorithm remains small.

Algorithm 5 procedure `explor3AP`

Parameters ↓: $C^1, C^2, C^3, h, UP(h), S$
Parameters ↑: S, S_{new}

```
--| Compute one optimal solution x̃ of (3AP_λ) with the cost matrix
--|    C^λ = λ^T C where λ is given by h
solveAP ( C^λ ↓, x̃ ↑)

--| Let S_new, the list of efficient solutions corresponding to B
S_new ← ∅
--| Let vUB be the initial value for the upper bound according to β1 or β2
UPR ← UP(h)
computeUpperBoundValue(S_new ↓, UPR ↓, vUB ↑)
if vUB > z^λ(x̃) then

  --| begin the ranking
  K ← 0
  while (z^λ(x^K) ≤ vUB) do
    K ← K + 1
    ComputeNextK_bestSolution(K ↓, C^λ ↓, x^K ↑)
    if not( isDominated(x^K ↓, S_new ↓) ) then
      S_new ← S_new ∪ {x^K}
      updateUBS(z(x^K) ↓, y^I ↓, UPR ↕)
      computeUpperBoundValue(S_new ↓, UPR ↓, vUB ↕)         --| according to β1 or β2
    end if
  end while
end if
S ← S ∪ S_new
```

We continue Example 2 to illustrate the exploration of the search area using bound β_2.

Example 2 (*Continued*). After Phase 1 $Y_{SN} = \left\{ \begin{pmatrix} 9 \\ 13 \\ 16 \end{pmatrix}, \begin{pmatrix} 19 \\ 11 \\ 17 \end{pmatrix}, \begin{pmatrix} 18 \\ 20 \\ 13 \end{pmatrix}, \begin{pmatrix} 14 \\ 20 \\ 14 \end{pmatrix}, \begin{pmatrix} 20 \\ 17 \\ 14 \end{pmatrix} \right\}$. We find the maximal complete sets of the three bi-objective problems. No new solutions are found. Thus $\bar{Y} = Y_{SN}$ and the nadir point is $y^N = (20, 20, 17)$.

Next we compute $D(\bar{Y}) = \left\{ u^1 = \begin{pmatrix} 20 \\ 13 \\ 17 \end{pmatrix}, u^2 = \begin{pmatrix} 20 \\ 20 \\ 16 \end{pmatrix} \right\}$ and initialize $UP = D(\bar{Y})$.

Phase 1 also yields two facets and the corresponding hyperplanes

$$h_1 = \{z \in \mathbb{R}^3 : 7z_1 + 3z_2 + 28z_3 - 550 = 0\}$$
$$h_2 = \{z \in \mathbb{R}^3 : 2z_1 + 15z_2 + 41z_3 - 869 = 0\}.$$

The distances between the upper bound points and the hyperplanes are approximately $d(u^1, h_1) = 3.618539$, $d(u^1, h_2) = 1.441531$, $d(u^2, h_1) = 3.377303$, and $d(u^2, h_2) = 2.905943$. Thus, $UP(h_1) = \emptyset$, $UP(h_2) = \{u^1, u^2\}$ and $H_p = \{h_2\}$. Only one hyperplane is selected for an exploration. Since there is no choice we will use the weight $\lambda = (2, 15, 41)$ given by h_2 to explore $(u^1 - \mathbb{R}^3_{\geqq}) \cup (u^2 - \mathbb{R}^3_{\geqq})$.

Let Y_{new} be the set of nondominated points obtained during the exploration. Initially $Y_{new} = \emptyset$. Let UPR be the set of points defining the search area of the exploration. Initially $UPR = \{u^1, u^2\}$.

The initial upper bound value is given by $\beta_2 = \max\{\alpha(u^1), \alpha(u^2)\}$ where for $i = 1, 2$, $\alpha(u^i) = \max\{\rho(F_1(u^i)), \rho(F_2(u^i)), \rho(F_3(u^i)), \lambda^T(u^i - (1, 1, 1))\}$. We have $\rho(F_1(u^1)) = (20, 12, 16)\lambda$, $\rho(F_2(u^1)) = (19, 13, 16)\lambda$, $\rho(F_3(u^1)) = (19, 12, 17)\lambda$, $\rho(F_1(u^2)) = (20, 19, 15)\lambda$, $\rho(F_2(u^2)) = (19, 20, 15)\lambda$, $\rho(F_3(u^3)) = (19, 19, 16)\lambda$. Therefore $\alpha(u^1) = \max\{876, 889, 915, 874\} = 915$ and $\alpha(u^2) = \max\{940, 953, 979, 938\} = 979$ and $\beta_2 = \max\{915, 979\} = 979$.

Starting the ranking we obtain a solution with objective point (18, 20, 13) that we add to Y_{new}. This point is on a face of $(u^2 - \mathbb{R}^3_{\geqq})$, therefore there is no update of UPR. The recomputation of β_2 gives no change.

The next solution gives the point (20, 17, 14) that we add to Y_{new}. This point is located on a face of $(u^2 - \mathbb{R}^3_{\geqq})$ and there is again no update of UPR. The recomputation of β_2 gives no change.

The next solution gives the point (9, 13, 16) that we add to Y_{new}. It is located on a face of both $(u^1 - \mathbb{R}^3_{\geqq})$ and $(u^2 - \mathbb{R}^3_{\geqq})$. Therefore there is no change in UPR. The recomputation of β_2 improves the upper bound value to 957.

Table 1
Experimental results for the methods of Sylva and Crema [23], Tenfelde-Podehl [25], Laumanns et al. [24] and the two phase method on (3AP) instances with CPU time in seconds.

Size	$\lvert Y_N \rvert$	Sylva and Crema [23]	Tenfelde-Podehl [25]	Laumanns et al. [24]	Two phase β_2
5	12	0.15	0.04	0.15	0.00
10	221	99 865.00	97.30	41.70	0.08
15	483	×	544.53	172.29	0.36
20	1 942	×	×	1 607.92	4.51
25	3 750	×	×	5 218.00	30.13
30	5 195	×	×	15 579.00	55.87
35	10 498	×	×	101 751.00	109.96
40	14 733	×	×	×	229.05
45	23 941	×	×	×	471.60
50	29 193	×	×	×	802.68

The next solution gives the point (18, 18, 14) that we add to Y_{new}. This point strictly dominates $u^2 = (20, 20, 16)$ and there is an update of *UPR* using Algorithm 2 which is now given by

$$UPR = \left\{ \begin{pmatrix} 20 \\ 13 \\ 17 \end{pmatrix}, \begin{pmatrix} 18 \\ 20 \\ 16 \end{pmatrix}, \begin{pmatrix} 20 \\ 18 \\ 16 \end{pmatrix}, \begin{pmatrix} 20 \\ 20 \\ 14 \end{pmatrix} \right\}.$$

The upper bound value remains 957.

The following solutions yield the dominated point (17, 14, 16), the nondominated point (19, 11, 17) that is added to Y_{new} without update of the upper bound value, the nondominated point (14, 20, 14) that is added to Y_{new} with no update of the upper bound value, the dominated point (12, 15, 16), and then the nondominated point (14, 18, 15) that we add to Y_{new}. As $(14, 18, 15) < (18, 20, 16)$, we update *UPR*, it is now given by

$$UPR = \left\{ \begin{pmatrix} 20 \\ 13 \\ 17 \end{pmatrix}, \begin{pmatrix} 20 \\ 20 \\ 14 \end{pmatrix}, \begin{pmatrix} 20 \\ 18 \\ 16 \end{pmatrix}, \begin{pmatrix} 14 \\ 20 \\ 16 \end{pmatrix}, \begin{pmatrix} 18 \\ 20 \\ 15 \end{pmatrix} \right\}$$

The upper bound value remains 957.

All other solutions until the upper bound value is reached are dominated. This concludes the exploration and we update $\bar{Y} \leftarrow \bar{Y} \cup Y_{new}$ with the nonsupported points (18, 18, 14) and (14, 18, 15). We update $UP \leftarrow UP \setminus \{u^1, u^2\}$ so that $UP = \emptyset$. The algorithm ends and $\bar{Y}$ is the set of all nondominated points.

7. Experimental results

We have generated a series of 10 instances with a size varying from 5×5 to 50×50 with a step of 5. The objective function coefficients are generated randomly in $\{0, \ldots, 20\}$ following a uniform distribution. A computer with a P4 EE 3.73 GHz processor and 4 Gb of RAM has been used for the experiments. We have used four methods to solve these instances, namely the two phase method as described above and three general methods for multi-objective combinatorial optimization problems, the methods by Sylva and Crema [23], Laumanns et al. [24], Tenfelde-Podehl [25]. All algorithms have been implemented in C. The binaries have been obtained using the compiler gcc with optimizer option -O3. The methods by Sylva and Crema [23], Laumanns et al. [24] and Tenfelde-Podehl [25] require the solution of assignment problems with some additional constraints. We have used used CPLEX 9.1 for that purpose. In the two phase method we have used the ranking algorithm by Pascoal et al. [18] and the upper bound value β_2. Computations were aborted after 100,000 s.

Table 1 summarizes the results. The number of nondominated points increases very fast with problem size. The differences in computation time between the four algorithms is enormous. In particular, the two phase method performs orders of magnitude faster than the general methods. This is not completely surprising because the two phase method exploits the problem structure and it was already fastest in the bi-objective case. The size of the gap between the two phase method and the other methods is huge. For example the time needed to solve the 35×35 instance is 925 times longer for the fastest of these general methods than the two phase method. We also note that the gap is increasing with problem size. As in the bi-objective case the time needed by the two phase method is very small on small instances, increases fast with increasing size of the instance until a medium size and then increases relatively slowly. In Table 1 we can see that from size 25×25 the time needed by the two phase method is about multiplied by 2 when the instance size increases by 5.

The main difficulties observed in the application of the three general methods to the three-objective assignment problem are the difficulty of the subproblems to be solved for [23], their number for [24], and the memory requirement for [25]. In all of these methods, the main difficulties are caused by the number of nondominated points (see [26] for details).

Experimental results on bi-objective assignment problems in [15,3] have shown that generating objective coefficients in $\{0, \ldots, r\}$ with a larger value of r increases the number of nondominated points and the time needed by any solution method. The same observation should be done in the three-objective case.

8. Conclusion

In this paper we have proposed a generalization of the two phase method to solve multi-objective integer programmes with more than two objectives. The method has been applied to the three-objective assignment problem. Numerical experiments show that the two phase method performs several orders of magnitude faster than general methods. The difference between the methods is bigger than in the bi-objective case. This shows the increasing importance of exploiting the problem structure with the number of objectives. In future work, our new two phase algorithm can be tested on other multi-objective integer linear programmes and also compared with more exact methods, such as the multi-objective branch and bound algorithms [9,27], or the multi-objective dynamic programming algorithm Bazgan et al. [28].

References

[1] E.L. Ulungu, J. Teghem, The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems, Foundations of Computing and Decision Sciences 20 (1995) 149–165.
[2] A.M. Geoffrion, Proper efficiency and the theory of vector maximization, Journal of Mathematical Analysis and Applications 22 (1968) 618–630.
[3] A. Przybylski, X. Gandibleux, M. Ehrgott, Two phase algorithms for the bi-objective assignment problem, Europen Journal of Operational Research 185 (2008) 509–533.
[4] H. Lee, P.S. Pulat, Bicriteria network flow problems: integer case, European Journal of Operational Research 66 (1993) 148–157.
[5] A. Sedeño-Noda, C. González-Martín, An algorithm for the biobjective integer minimum cost flow problem, Computers & Operations Research 28 (2) (2001) 139–156.
[6] M. Visée, J. Teghem, M. Pirlot, E.L. Ulungu, Two-phases method and branch and bound procedures to solve the bi-obective knapsack problem, Journal of Global Optimization 12 (1998) 139–155.
[7] R.M. Ramos, S. Alonso, J. Sicilia, C. González, The problem of the optimal biobjective spanning tree, European Journal of Operational Research 111 (1998) 617–628.
[8] M. Ehrgott, D.M. Ryan, The method of elastic constraints for multiobjective combinatorial optimization and its application in airline crew scheduling, in: T. Tanino, T. Tanaka, M. Inuiguchi (Eds.), Multi-Objective Programming and Goal-Programming, Springer Verlag, Berlin, 2003, pp. 117–122.
[9] G. Mavrotas, D. Diakoulaki, A branch and bound algorithm for mixed zero-one multiple objective linear programming, European Journal of Operational Research 107 (3) (1998) 530–541.
[10] M. Ehrgott, A discussion of scalarization techniques for multiobjective integer programming, Annals of Operations Research 147 (2006) 343–360.
[11] P. Hansen, Bicriterion path problems, in: G. Fandel, T. Gal (Eds.), Multiple Criteria Decision Making Theory and Application, in: Lecture Notes in Economics and Mathematical Systems, vol. 177, Springer Verlag, Berlin, 1979, pp. 109–127.
[12] A. Przybylski, X. Gandibleux, M. Ehrgott, A recursive algorithms for finding all nondominated extreme points in the outcome set of a multiobjective integer program. INFORMS Journal on Computing (2010), in print (doi:10.1287/ijoc.1090.0342).
[13] M. Ehrgott, D. Tenfelde-Podehl, Computation of ideal and nadir values and implications for their use in MCDM methods, European Journal of Operational Research 151 (2003) 119–139.
[14] M. Ehrgott, X. Gandibleux, Bound sets for biobjective combinatorial optimization problems, Computers and Operations Research 34 (2007) 2674–2694.
[15] C.R. Pedersen, L.R. Nielsen, K.A. Andersen, The bicriterion multi modal assignment problem: introduction, analysis, and experimental results, INFORMS Journal on Computing 20 (2008) 400–411.
[16] C.R. Chegireddy, H.W. Hamacher, Algorithms for finding k-best perfect matchings, Discrete Applied Mathematics 18 (1987) 155–165.
[17] M.L. Miller, H.S. Stone, I.J. Cox, Optimizing murty's ranked assignment method, IEEE Transactions on Aerospace and Electronic Systems 33 (1997) 851–862.
[18] M. Pascoal, M.E. Captivo, J. Clìmaco, A note on new variant of Murty's ranking assignments algorithm, 4OR. A Quarterly Journal of Operations Research 1 (2003) 243–255.
[19] C.R. Pedersen, L.R. Nielsen, K.A. Andersen, An algorithm for ranking assignments using reoptimization, Computers and Operations Research 35 (2008) 3714–3726.
[20] C.R. Pedersen, L.R. Nielsen, K.A. Andersen, Erratum to "an algorithm for ranking assignments using reoptimization", Computers and Operations Research 37 (2010) 426–427.
[21] K.G. Murty, An algorithm for ranking all the assignments in order of increasing cost, Operations Research 16 (1968) 682–687.
[22] H.W. Hamacher, M. Queyranne, k best solutions to combinatorial optimization problems, Annals of Operations Research 4 (1985) 123–143.
[23] J. Sylva, A. Crema, A method for finding the set of non-dominated vectors for multiple objective integer linear programs, European Journal of Operational Research 158 (2004) 46–55.
[24] M. Laumanns, L. Thiele, E. Zitzler, An adaptive scheme to generate the pareto front based on the epsilon-constraint method, European Journal of Operational Research 169 (2005) 932–942.
[25] D. Tenfelde-Podehl, A recursive algorithm for multiobjective combinatorial optimization problems with q criteria, Technical report, Institut für Mathematik, Technische Universität Graz, 2003.
[26] A. Przybylski, X. Gandibleux, M. Ehrgott, Computational results for four exact methods to solve the three-objective assignment problem, in: V. Barichard, M. Ehrgott, X. Gandibleux, V. T'Kindt (Eds.), Multiobjective Programming and Goal Programming—Theoretical Results and Practical Applications, in: Lecture Notes in Economics and Mathematical Systems, vol. 618, Springer Verlag, Berlin, 2009, pp. 79–88.
[27] F. Sourd, O. Spanjaard, A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem, INFORMS Journal on Computing 20 (2008) 472–484.
[28] C. Bazgan, H. Hugot, D. Vanderpooten, Solving efficiently the 0–1 multi-objective knapsack problem, Computers & Operations Research 36 (2009) 260–279.