Ann Oper Res (2016) 244:677–696
DOI 10.1007/s10479-016-2149-9

ORIGINAL PAPER

An interactive approach for biobjective integer programs under quasiconvex preference functions

Diclehan Tezcaner Öztürk[1] · Murat Köksalan[2]

Published online: 12 March 2016

Abstract We develop an interactive algorithm for biobjective integer programs that finds the most preferred solution of a decision maker whose preferences are consistent with a quasiconvex preference function to be minimized. During the algorithm, preference information is elicited from the decision maker. Based on this preference information and the properties of the underlying quasiconvex preference function, the algorithm reduces the search region and converges to the most preferred solution progressively. Finding the most preferred solution requires searching both supported and unsupported nondominated points, where the latter is harder. We develop theory to further restrict the region where unsupported nondominated points may lie. We demonstrate the algorithm on the generalized biobjective traveling salesperson problem where there are multiple efficient edges between node pairs and show its performance on a number of randomly generated instances.

Keywords Multiobjective decision making · Combinatorial optimization · Interactive method · Biobjective traveling salesperson problem

1 Introduction

In multiobjective problems nondominated points are of interest where each nondominated point is better than any other feasible point in at least one objective. In practical multiobjective integer programs, there are many nondominated points and finding all nondominated points is computationally hard. Furthermore, the decision maker (DM) typically is not interested in many of the nondominated points. Concentrating on the preferred regions is important in such situations to reduce the computational burden.

✉ Diclehan Tezcaner Öztürk
diclehan.ozturk@tedu.edu.tr

[1] Department of Industrial Engineering, TED University, 06420 Ankara, Turkey

[2] Department of Industrial Engineering, Middle East Technical University, 06800 Ankara, Turkey

In the literature, there are different approaches that find the most preferred point of a DM. One approach is to estimate the underlying preference function of the DM, which is unknown to the DM and the analyst. However, estimating such a function that sufficiently represents the DM's preferences is problematic. Interactive approaches, on the other hand, aim to converge to the most preferred point by progressively obtaining preference information from a DM. There are many interactive approaches in the literature. Zionts (1981) developed an interactive algorithm to choose between discrete alternatives for a DM whose preferences are consistent with a linear preference function. Korhonen et al. (1984) developed a version that assumes a quasiconcave preference function and Köksalan and Sagala (1995) generalized for any monotone preference function. Lokman et al. (2014) also considered a quasiconcave preference function but they addressed integer programs rather than choosing from an available set of alternatives.

In this study, we develop an interactive algorithm for biobjective integer programs where each objective is of minimization type. The algorithm finds the most preferred point of a DM who is assumed to have an underlying quasiconvex preference function to be minimized. Note that, this case is equivalent to maximizing a quasiconcave preference function (which is more common in the literature) where all objectives are of maximization type. Tezcaner and Köksalan (2011) developed an algorithm, *BestSol*, for biobjective integer programs to find the most preferred point of a DM whose preferences are consistent with a linear preference function. In this paper, we consider the more general case where the DM's preferences are consistent with a general quasiconvex function. These functions represent a wide spectrum of preference structures including linear preference functions.

We demonstrate our algorithm on the generalized biobjective traveling salesperson problem (BOTSP). In this problem there are typically multiple efficient tours. Furthermore, the generalized BOTSP is considered to have multiple efficient edges between each node pair. The existence of these efficient edges lead to a separate combinatorial optimization problem: the biobjective shortest path problem (BOSPP). A subset of these efficient edges appears in the efficient tours.

We organize the paper as follows. We give some definitions in Sect. 2. We provide the details of our interactive algorithm in Sect. 3 and discuss it on the generalized BOTSP in Sect. 4. We demonstrate the algorithm on an example and present computational results on randomly generated problems in Sect. 5. We make concluding remarks in Sect. 6.

2 Definitions

Before we present the problem in detail, we give some relevant definitions. These definitions are adapted from Tezcaner and Köksalan (2011) and they will be used throughout the paper, unless otherwise stated.

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ denote an integer-valued decision variable vector, $X \subset \mathbb{Z}^n$ the feasible set, point $\boldsymbol{z}(\boldsymbol{x}) = (z_1(\boldsymbol{x}), z_2(\boldsymbol{x}), \ldots, z_p(\boldsymbol{x}))$ the objective function vector corresponding to the decision vector $\boldsymbol{x}$, and $Z = \{\boldsymbol{z}(\boldsymbol{x}) : \boldsymbol{x} \in X\} \subset \mathbb{R}^p$ the image of the feasible set in objective space. Note that we refer to any vector in set X as a *solution* and its image in the objective space Z as a *point*. We will refer to the most preferred (efficient) solution and the most preferred (nondominated) point using the same distinction throughout the paper. We assume, without loss of generality, that all objectives are to be minimized.

Definition 2.1 A point $\boldsymbol{z}(\boldsymbol{x}) \in Z$ is said to be *nondominated* iff there does not exist $\boldsymbol{z}(\boldsymbol{x}') \in Z$ such that $z_j(\boldsymbol{x}') \leq z_j(\boldsymbol{x})$ for $j = 1, \ldots, p$ and $z_j(\boldsymbol{x}') < z_j(\boldsymbol{x})$ for at least one j. If there exists such a $z_j(\boldsymbol{x}')$, $z_j(\boldsymbol{x})$ is said to be *dominated*. The set of all nondominated points, $Z_{ND} \subset Z$, constitute the nondominated set.

Definition 2.2 If $\boldsymbol{z}(\boldsymbol{x}) \in Z$ is *nondominated*, then $\boldsymbol{x}$ is said to be *efficient*, and if $\boldsymbol{z}(\boldsymbol{x})$ is *dominated*, $\boldsymbol{x}$ is said to be *inefficient*. The set of all efficient solutions constitute the efficient set.

Definition 2.3 A nondominated point $\boldsymbol{z}(\boldsymbol{x}) \in Z_{ND}$ is a *supported nondominated point* iff there exists a positive linear combination of objectives that is minimized by $\boldsymbol{x}$. Otherwise, $\boldsymbol{z}(\boldsymbol{x})$ is an *unsupported nondominated point*.

Definition 2.4 An *extreme nondominated point* is a supported nondominated point that has the minimum possible value in at least one of the objectives.

From this point on, we will use the notation $\boldsymbol{y^i} = \left(y_1^i, y_2^i, \ldots, y_p^i\right)$ to denote vector $\boldsymbol{z}(\boldsymbol{x_i}) = (z_1(\boldsymbol{x_i}), z_2(\boldsymbol{x_i}), \ldots, z_p(\boldsymbol{x_i}))$ whenever we need to simplify the notation. Although more than one $\boldsymbol{x}$ vector may correspond to the same $\boldsymbol{y}$ vector, we are only interested in one of them since we operate in the objective space. Therefore, we assume that $\boldsymbol{y^i} \neq \boldsymbol{y^j}$ when $i \neq j$ for the sake of further simplifying the notation.

Definition 2.5 Let $T = \{t | \boldsymbol{y^t} \in Z_{ND}\}$ and $\boldsymbol{y^i} \in Z_{ND}$ be a supported nondominated point. Point $\boldsymbol{y^j} \in Z$, $j \neq i$, is adjacent to $\boldsymbol{y^i}$ iff there does not exist $\mu_t, \lambda \in \mathbb{R}$ satisfying $\sum_{t \in T, t \neq j} \mu_t = 1, 0 \leq \mu_t \leq 1, 0 < \lambda \leq 1$, and $\sum_{t \in T, t \neq j} \mu_t \boldsymbol{y^t} \leq \lambda \boldsymbol{y^j} + (1 - \lambda)\boldsymbol{y^i}$.

The above definition implies that adjacent nondominated points are supported nondominated points of the feasible solution space considered.

In a bicriteria problem, there are at most two nondominated points adjacent to a given nondominated point (Ramesh et al. 1990).

Definition 2.6 Let $\left\{\boldsymbol{y^i} \in \mathbb{R}^p, i = 1, \ldots, m\right\}$ be a set of points. $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is a quasiconvex function if and only if $f\left(\sum_{i=1}^m \mu_i \boldsymbol{y^i}\right) \leq \max_i \left\{f(\boldsymbol{y^i})\right\}$ for $\mu_i \in \mathbb{R}, i = 1, \ldots, m$ such that $\sum_{i=1}^m \mu_i = 1, \mu_i \geq 0, i = 1, \ldots, m$.

3 An interactive algorithm

As mentioned above, there are many nondominated points in multiobjective problems in practice and it is neither practical nor necessary to generate all of them. In our interactive approach, we aim to generate a small subset of the nondominated set while converging the most preferred point. We select and provide the DM a pair of nondominated points and ask for the preferred one. This procedure is repeated and the preferences of the DM lead our search to the most preferred point of the DM.

We assume that the DM's preferences are consistent with a quasiconvex preference function to be minimized. In the literature, quasiconcave preference functions (with objectives to be maximized) are widely used. If all objectives are minimization type, the same theory directly applies. We use Lemma 1 and Theorem 1 [adapted from Lemma 2 and Theorem 1 of Korhonen et al. (1984), respectively] to reduce the objective space corresponding to the implied inferior regions based on the expressed preferences of the DM.

Lemma 1 *Consider a quasiconvex function $f : \mathbb{R}^p \to \mathbb{R}$, and points $\boldsymbol{y^i} \in \mathbb{R}^p$, $i = 1, 2, \ldots, m$. Let $f\left(\boldsymbol{y^k}\right) > f\left(\boldsymbol{y^i}\right), i \neq k$ and $Y = \left\{\boldsymbol{y} | \boldsymbol{y} = \boldsymbol{y^k} + \sum_{i=1; i\neq k}^{m} \mu_i(\boldsymbol{y^k} - \boldsymbol{y^i}), \mu_i \geq 0\right\}$. If $\boldsymbol{y} \in Y, \boldsymbol{y} \neq \boldsymbol{y^k}$ then $f(\boldsymbol{y}) \geq f(\boldsymbol{y^k})$.*

Proof The proof directly follows from the proof of Lemma 2 in Korhonen et al. (1984). □

Theorem 1 *Let $f : \mathbb{R}^p \to \mathbb{R}$, be a nondecreasing quasiconvex function, $\boldsymbol{y^i} \in \mathbb{R}^p, i = 1, 2, \ldots, m$, and $\boldsymbol{y} \in \mathbb{R}^p$ be any point, and $f\left(\boldsymbol{y^k}\right) > f\left(\boldsymbol{y^i}\right), i \neq k$. If there is a feasible solution to*

$$\boldsymbol{y^k} + \sum_{i=1; i\neq k}^{m} \mu_i(\boldsymbol{y^k} - \boldsymbol{y^i}) \leq \boldsymbol{y}, \mu_i \geq 0, \quad i = 1, \ldots, m,$$

then $f(\boldsymbol{y}) \geq f\left(\boldsymbol{y^k}\right)$.

Proof The proof directly follows from the proof of Theorem 1 in Korhonen et al. (1984). □

Remark The set Y in Lemma 1 defines a cone. We will refer to the cone and the region dominated by the cone as the *cone-inferior region*. If a DM's preferences are consistent with f, then any point $\boldsymbol{y}$ that is in the cone-inferior region is less preferred than $\boldsymbol{y^k}$ according to Theorem 1.

We present the following lemma to find the regions in the objective space that do not contain any nondominated points using the information that two nondominated points are adjacent.

Lemma 2 *Let nondominated point $\boldsymbol{y^k}$ be adjacent to a nondominated point $\boldsymbol{y^r}$, $r \neq k$. Then, there does not exist any $\boldsymbol{y^i}$, $i \neq k, r$, that dominates some affine combination of $\boldsymbol{y^k}$ and $\boldsymbol{y^r}$.*

Proof Recall $T = \left\{t | \boldsymbol{y^t} \in Z_{ND}\right\}$. For $\boldsymbol{y^r}$ to be adjacent to $\boldsymbol{y^k}$, the following inequality does not hold for any $\mu_t, \lambda \in \mathbb{R}$ satisfying $\sum_{t\in T, t\neq r} \mu_t = 1, 0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$.

$$\sum_{t\in T, t\neq r} \mu_t \boldsymbol{y^t} \leq \lambda \boldsymbol{y^r} + (1 - \lambda)\boldsymbol{y^k} \tag{1}$$

Or equivalently,

$$\sum_{t\in T, t\neq r} \mu_t \boldsymbol{y^t} - \boldsymbol{y^k} \leq \lambda \left(\boldsymbol{y^r} - \boldsymbol{y^k}\right) \tag{2}$$

Similarly, for $\boldsymbol{y^k}$ to be adjacent to $\boldsymbol{y^r}$, the following inequality does not hold for any $\mu_t, \lambda \in \mathbb{R}$ satisfying $\sum_{t\in T, t\neq k} \mu_t = 1, 0 \leq \mu_t \leq 1$ and $0 < \lambda \leq 1$.

$$\sum_{t\in T, t\neq k} \mu_t \boldsymbol{y^t} - \boldsymbol{y^r} \leq \lambda \left(\boldsymbol{y^k} - \boldsymbol{y^r}\right) \tag{3}$$

Suppose there exists a nondominated point $\boldsymbol{y^i}$ dominating some affine combination of $\boldsymbol{y^k}$ and $\boldsymbol{y^r}$.

$$\boldsymbol{y^i} \leq \lambda_r \boldsymbol{y^r} + \lambda_k \boldsymbol{y^k} \quad \text{for some } \lambda_k, \lambda_r \in \mathbb{R} \text{ and } \lambda_k + \lambda_r = 1. \tag{4}$$

This inequality is equivalent to the following two inequalities.

$$\boldsymbol{y^i} - \boldsymbol{y^k} \leq \lambda_r \left(\boldsymbol{y^r} - \boldsymbol{y^k}\right) \quad \text{for some } \lambda_r \in \mathbb{R}. \tag{5}$$

$$\boldsymbol{y^i} - \boldsymbol{y^r} \leq \lambda_k \left(\boldsymbol{y^k} - \boldsymbol{y^r}\right) \quad \text{for some } \lambda_k \in \mathbb{R}. \tag{6}$$

Setting $\mu_k = 1 - \mu_i$ for $\mu_i > 0$ and $\mu_t = 0$ for $t \neq k, i$ in (2), we obtain the following inequality.

$$\boldsymbol{y^i} - \boldsymbol{y^k} \leq \frac{\lambda}{\mu_i} \left(\boldsymbol{y^r} - \boldsymbol{y^k}\right) \quad \text{for } 0 < \lambda, \mu_i \leq 1. \tag{7}$$

Similarly, setting $\mu_r = 1 - \mu_i$ for $\mu_i > 0$ and $\mu_t = 0$ for $t \neq r, i$ in (3), we obtain the following inequality.

$$\boldsymbol{y^i} - \boldsymbol{y^r} \leq \frac{\lambda}{\mu_i} \left(\boldsymbol{y^k} - \boldsymbol{y^r}\right) \quad \text{for } 0 < \lambda, \mu_i \leq 1. \tag{8}$$

In (7) and (8), $\frac{\lambda}{\mu_i} \in \mathbb{R}_+$ for $0 < \lambda, \mu_i \leq 1$. λ_k or λ_r is positive to satisfy $\lambda_k + \lambda_r = 1$ in (4). If $\lambda_r > 0$, the inequality (5) becomes equivalent to (7) for $\frac{\lambda}{\mu_i} = \lambda_r$ that contradicts with $\boldsymbol{y^r}$ being adjacent to $\boldsymbol{y^k}$. If $\lambda_k > 0$, the inequality (6) becomes equivalent to (8) for some $\frac{\lambda}{\mu_i} = \lambda_k$ that contradicts with $\boldsymbol{y^k}$ being adjacent to $\boldsymbol{y^r}$. □

With Lemma 3 we provide the theory showing the region that does not contain the most preferred point and hence can be reduced, around a nondominated point that is preferred to both its adjacent points.

Lemma 3 *Let $f : \mathbb{R}^2 \to \mathbb{R}$, be a quasiconvex function (to be minimized) that represents the preferences of the DM, $\boldsymbol{y^i} = \left(y_1^i, y_2^i\right)$, and $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$ be the left and right adjacent points of the supported nondominated point $\boldsymbol{y^b}$. Let $f\left(\boldsymbol{y^b}\right) < f(\boldsymbol{y^a})$ and $f\left(\boldsymbol{y^b}\right) < f(\boldsymbol{y^c})$ and $f(\boldsymbol{y^*}) = \min_{\boldsymbol{y} \in Z_{ND}}\{f(\boldsymbol{y})\}$. Then*

$$\begin{aligned}\boldsymbol{y^*} \in Y^* = &\left\{\boldsymbol{y} | \boldsymbol{y} = \mu_a \boldsymbol{y^a} + \mu_b \boldsymbol{y^b} + \mu_{ab}\eta\left(\boldsymbol{y^a}, \boldsymbol{y^b}\right), \mu_a + \mu_b + \mu_{ab} = 1, \mu_a, \mu_b, \mu_{ab} \geq 0\right\}\\ &\cup \left\{\boldsymbol{y} | \boldsymbol{y} = \mu_c \boldsymbol{y^c} + \mu_b \boldsymbol{y^b} + \mu_{cb}\eta\left(\boldsymbol{y^c}, \boldsymbol{y^b}\right), \mu_c + \mu_b + \mu_{cb} = 1, \mu_c, \mu_b, \mu_{cb} \geq 0\right\}\\ &\textit{where } \eta\left(\boldsymbol{y^i}, \boldsymbol{y^j}\right) = \left(\max\left(y_1^i, y_1^j\right), \max\left(y_2^i, y_2^j\right)\right).\end{aligned}$$

Proof If point $\boldsymbol{y^b}$ and points $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$ are adjacent points, there does not exist any point dominating any affine combinations of the adjacent point pairs due to Lemma 2. This eliminates region $Y_1 = \left\{\boldsymbol{y} | \boldsymbol{y} < \lambda_a \boldsymbol{y^a} + \lambda_b \boldsymbol{y^b} \text{ for all } \lambda_a, \lambda_b \in \mathbb{R} \text{ such that } \lambda_a + \lambda_b = 1\right\} \cup \left\{\boldsymbol{y} | \boldsymbol{y} < \lambda_c \boldsymbol{y^c} + \lambda_b \boldsymbol{y^b} \text{ for all } \lambda_c, \lambda_b \in \mathbb{R} \text{ such that } \lambda_c + \lambda_b = 1\right\}$. We eliminate the cone-inferior regions $Y_2 = \left\{\boldsymbol{y} | \boldsymbol{y} \geq \boldsymbol{y^a} + \mu_b\left(\boldsymbol{y^a} - \boldsymbol{y^b}\right), \mu_b \geq 0\right\}$ and $Y_3 = \left\{\boldsymbol{y} | \boldsymbol{y} \geq \boldsymbol{y^c} + \mu_b\left(\boldsymbol{y^c} - \boldsymbol{y^b}\right),\right.$ $\left.\mu_b \geq 0\right\}$ since $\boldsymbol{y^*} \notin \{Y_2 \cup Y_3\}$ due to Lemma 1. For $\boldsymbol{y^d} \in Y_4 = \{\boldsymbol{y} | y_j \geq y_j^i, j = 1, 2,$ and $y_j > y_j^i$ for at least one j for $i = a, b, c\}$, $f\left(\boldsymbol{y^d}\right) \geq f(\boldsymbol{y^a})$ or $f\left(\boldsymbol{y^d}\right) \geq f\left(\boldsymbol{y^b}\right)$ or $f\left(\boldsymbol{y^d}\right) \geq f(\boldsymbol{y^c})$. Therefore, $\boldsymbol{y^*} \in Y^* = \left\{\boldsymbol{y} | \boldsymbol{y} = \mu_a \boldsymbol{y^a} + \mu_b \boldsymbol{y^b} + \mu_{ab}\eta\left(\boldsymbol{y^a}, \boldsymbol{y^b}\right), \mu_a + \mu_b + \right.$ $\left.\mu_{ab} = 1, \mu_a, \mu_b, \mu_{ab} \geq 0\right\} \cup \left\{\boldsymbol{y} | \boldsymbol{y} = \mu_c \boldsymbol{y^c} + \mu_b \boldsymbol{y^b} + \mu_{cb}\eta\left(\boldsymbol{y^c}, \boldsymbol{y^b}\right), \mu_c + \mu_b + \mu_{cb} = 1,\right.$ $\left.\mu_c, \mu_b, \mu_{cb} \geq 0\right\}$. □

We demonstrate Lemma 3 in Fig. 1. Here, we have three nondominated points $\boldsymbol{y^a}$, $\boldsymbol{y^b}$ and $\boldsymbol{y^c}$. We eliminate the regions in which the most preferred point of the DM cannot lie. Given that point $\boldsymbol{y^b}$ is preferred to its two adjacent points, $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$, there cannot be any points in Region 1 (which is the union of the regions dominating the affine combinations of $\boldsymbol{y^a}$ and

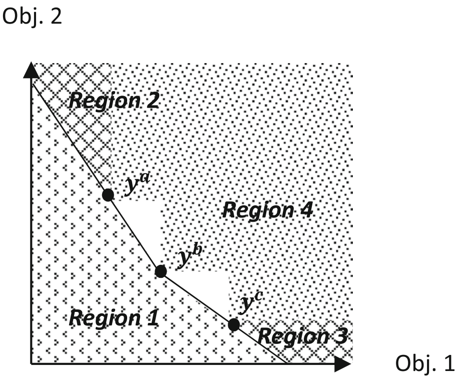

Fig. 1 Admissible regions for the most preferred point

$\boldsymbol{y^b}$, and $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$) due to Lemma 2. The most preferred point of the DM cannot lie in the cone-inferior regions 2 and 3 due to Theorem 1. Region 4 is composed of points dominated by $\boldsymbol{y^a}, \boldsymbol{y^b}$ or $\boldsymbol{y^c}$. Therefore, the only regions in which the most preferred point of the DM can lie are the triangular regions between the current most preferred point, $\boldsymbol{y^b}$, and its two adjacent points, $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$.

We do not impose any additional constraints on the objective space before finding the supported nondominated point that is preferred to all its adjacent points. We refer to the objective space that has no additional constraints as the *original objective space*.

The three additional constraints; an upper bound on the first objective ($z_1(\boldsymbol{x}) \leq UB^1$), an upper bound on the second objective ($z_2(\boldsymbol{x}) \leq UB^2$), and a lower bound on the weighted combination of these objectives ($wz_1(\boldsymbol{x}) + (1-w)\,z_2(\boldsymbol{x}) \geq LB$) define a region (triangle) in which the most preferred point of the DM may lie. We refer to the region defined by these triangles as the *reduced objective space*. The nondominated points in the reduced objective space are, by definition, unsupported nondominated points. In Fig. 1, we have two regions; the left triangle and the right triangle. The left and right triangles' bounds on the two objectives are $z_1(\boldsymbol{x}) \leq y_1^b$ and $z_2(\boldsymbol{x}) \leq y_2^a - \delta$, and $z_1(\boldsymbol{x}) \leq y_1^c - \delta$ and $z_2(\boldsymbol{x}) \leq y_2^b$, respectively, where δ is a sufficiently small positive constant. By subtracting δ from y_2^a and y_1^c, we exclude the inferior points $\boldsymbol{y^a}$ and $\boldsymbol{y^c}$ from the triangles. The lower bound on the weighted combination of the two objectives is a redundant constraint since there are no solutions that violate this constraint given the upper bounds on the objectives. Therefore, we will not use the lower bound on the weighted combination of the two objectives.

We next summarize the steps of the interactive algorithm to find the most preferred point of a DM whose preferences are consistent with a quasiconvex preference function for a biobjective integer program. In this algorithm, we generalize the concepts used in Tezcaner Köksalan (2011)'s approach for the quasiconvex case. We use the ideas in Lemmas 1, 2, 3, and Theorem 1 above to narrow down the objective space.

3.1 The steps of the interactive algorithm (IA)

Let $w_{LE} = 1 - \rho$, and $w_{RE} = \rho$. Throughout the algorithm, we use ρ as a sufficiently small positive constant to prevent obtaining dominated points (see Steuer 1986, pp. 422–430 for further discussion on how to choose ρ).

Step 1. Find the two extreme nondominated points of the problem minimizing $U(\boldsymbol{x}) = wz_1(\boldsymbol{x}) + (1-w)z_2(\boldsymbol{x})$ for $w = w_{LE}$ and w_{RE}, respectively, and denote them by $\boldsymbol{y^{LE}}$

and $\boldsymbol{y}^{\boldsymbol{RE}}$, respectively. If $\boldsymbol{y}^{\boldsymbol{LE}} = \boldsymbol{y}^{\boldsymbol{RE}}$, the problem has only one nondominated point. Let $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{LE}} = \boldsymbol{y}^{\boldsymbol{RE}}$ and go to Step 10. Otherwise, go to Step 2.

Step 2. Let $w' = \frac{y_2^{LE} - y_2^{RE}}{(y_2^{LE} - y_2^{RE}) + (y_1^{RE} - y_1^{LE})}$

Find the solution that minimizes the function, $U(\boldsymbol{x}) = wz_1(\boldsymbol{x}) + (1 - w)z_2(\boldsymbol{x})$ formed with weight $w = w'$. Let the corresponding point be $\boldsymbol{y}'$. If $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{LE}}$ or $\boldsymbol{y}^{\boldsymbol{RE}}$, set $\boldsymbol{y}^{\boldsymbol{L}} = \boldsymbol{y}^{\boldsymbol{LE}}$ and $\boldsymbol{y}^{\boldsymbol{R}} = \boldsymbol{y}^{\boldsymbol{RE}}$, and go to Step 3. Otherwise, go to Step 4.

Step 3. Ask the DM to compare $\boldsymbol{y}^{\boldsymbol{L}}$ with $\boldsymbol{y}^{\boldsymbol{R}}$.

- If $\boldsymbol{y}^{\boldsymbol{L}}$ is preferred to $\boldsymbol{y}^{\boldsymbol{R}}$, set the two bounds of the right triangle as $z_2(\boldsymbol{x}) \leq y_2^L$, $z_1(\boldsymbol{x}) \leq y_1^R - \delta$. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{L}}$. Go to Step 8.
- If $\boldsymbol{y}^{\boldsymbol{R}}$ is preferred to $\boldsymbol{y}^{\boldsymbol{L}}$, set the two bounds of the left triangle as $z_2(\boldsymbol{x}) \leq y_2^L - \delta$, $z_1(\boldsymbol{x}) \leq y_1^R$. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{R}}$. Go to Step 6.

Step 4. Find the left adjacent point, $\boldsymbol{y}^{\boldsymbol{L}}$, of $\boldsymbol{y}'$. Ask the DM to compare $\boldsymbol{y}'$ and $\boldsymbol{y}^{\boldsymbol{L}}$.

- If $\boldsymbol{y}'$ is preferred to $\boldsymbol{y}^{\boldsymbol{L}}$, set the two bounds for the left triangle as $z_2(\boldsymbol{x}) \leq y_2^L - \delta$, $z_1(\boldsymbol{x}) \leq y_1'$. If no bounds for the right triangle has been defined yet, go to Step 5; otherwise, go to Step 6.
- If $\boldsymbol{y}^{\boldsymbol{L}}$ is preferred to $\boldsymbol{y}'$, set the two bounds for the right triangle as $z_2(\boldsymbol{x}) \leq y_2^L$, $z_1(\boldsymbol{x}) \leq y_1' - \delta$. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{L}}$. If $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{LE}}$, go to Step 8 and if $\boldsymbol{y}' \neq \boldsymbol{y}^{\boldsymbol{LE}}$, go to Step 4.

Step 5. Find the right adjacent point, $\boldsymbol{y}^{\boldsymbol{R}}$, of $\boldsymbol{y}'$. Ask the DM to compare $\boldsymbol{y}'$ with $\boldsymbol{y}^{\boldsymbol{R}}$.

- If $\boldsymbol{y}'$ is preferred to $\boldsymbol{y}^{\boldsymbol{R}}$, set the two bounds for the right triangle as $z_1(\boldsymbol{x}) \leq y_1^R - \delta$, $z_2(\boldsymbol{x}) \leq y_2'$. Go to Step 6.
- If $\boldsymbol{y}^{\boldsymbol{R}}$ is preferred to $\boldsymbol{y}'$, set the two bounds for the left triangle as $z_2(\boldsymbol{x}) \leq y_2' - \delta$, $z_1(\boldsymbol{x}) \leq y_1^R$. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{R}}$. If $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{RE}}$, go to Step 6 and if $\boldsymbol{y}' \neq \boldsymbol{y}^{\boldsymbol{RE}}$, go to Step 5.

Step 6. Search for the left adjacent point of $\boldsymbol{y}'$ in the left triangle.

- If there is no new nondominated point:
    - If the bounds for the right triangle have been defined, go to Step 8.
    - If no bounds for the right triangle has been defined yet, go to Step 10.
- If a new nondominated point is found, denote it by $\boldsymbol{y}^{\boldsymbol{L}}$ and go to Step 7.

Step 7. Ask the DM to compare $\boldsymbol{y}'$ with $\boldsymbol{y}^{\boldsymbol{L}}$.

- If $\boldsymbol{y}'$ is preferred to $\boldsymbol{y}^{\boldsymbol{L}}$, set $z_2(\boldsymbol{x}) \leq y_2^L - \delta$ for the left triangle and go to Step 6.
- If $\boldsymbol{y}^{\boldsymbol{L}}$ is preferred to $\boldsymbol{y}'$, set $z_1(\boldsymbol{x}) \leq y_1^L$ for the left triangle, and $z_2(\boldsymbol{x}) \leq y_2^L$ and $z_1(\boldsymbol{x}) \leq y_1' - \delta$ for the right triangle. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{L}}$ and go to Step 6.

Step 8. Search for the right adjacent point of $\boldsymbol{y}'$ in the right triangle. If there is no new nondominated point, go to Step 10. If a new nondominated point is found, denote it by $\boldsymbol{y}^{\boldsymbol{R}}$ and go to Step 9.

Step 9. Ask the DM to compare $\boldsymbol{y}'$ with $\boldsymbol{y}^{\boldsymbol{R}}$.

- If $\boldsymbol{y}'$ is preferred to $\boldsymbol{y}^{\boldsymbol{R}}$, set $z_1(\boldsymbol{x}) \leq y_1^R - \delta$ for the right triangle and go to Step 8.
- If $\boldsymbol{y}^{\boldsymbol{R}}$ is preferred to $\boldsymbol{y}'$, set $z_2(\boldsymbol{x}) \leq y_2' - \delta$ and $z_1(\boldsymbol{x}) \leq y_1^R$ for the left triangle, $z_2(\boldsymbol{x}) \leq y_2^R$ for the right triangle. Set $\boldsymbol{y}' = \boldsymbol{y}^{\boldsymbol{R}}$ and go to Step 6.

Step 10. The most preferred point $\boldsymbol{y}^*$ is $\boldsymbol{y}'$.

To demonstrate, consider nondominated points $\boldsymbol{y}^{\boldsymbol{a}}, \boldsymbol{y}^{\boldsymbol{b}}, \boldsymbol{y}^{\boldsymbol{c}}, \boldsymbol{y}^{\boldsymbol{d}}$ and $\boldsymbol{y}^{\boldsymbol{e}}$ in Fig. 2. The most preferred point of the DM that minimizes the underlying preference function shown in Fig. 2a

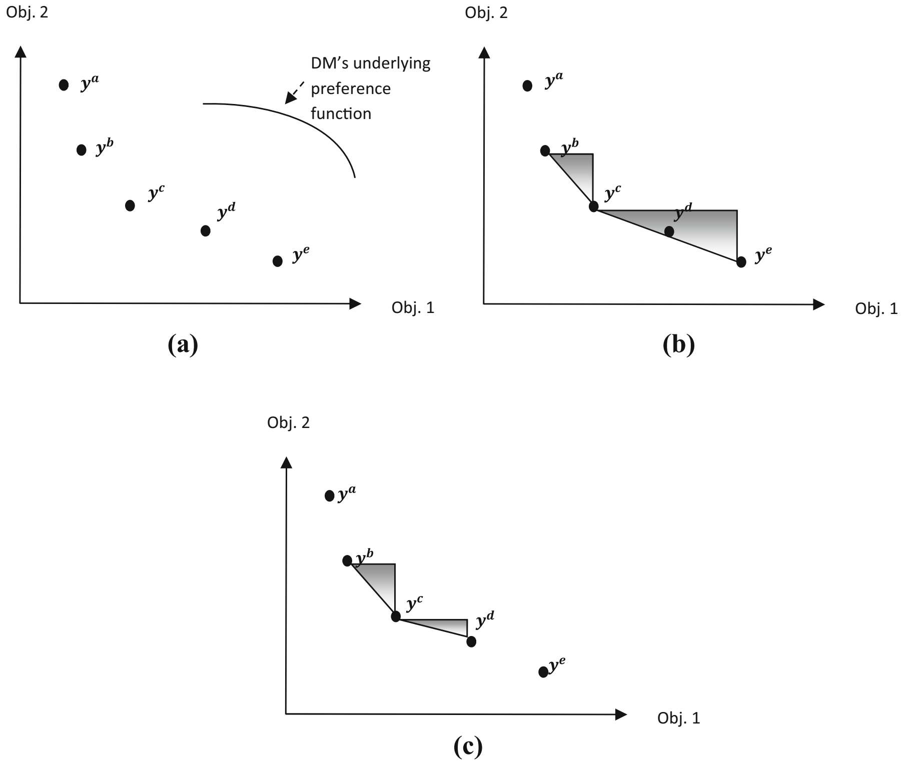

Fig. 2 Demonstration of the interactive algorithm. **a** Nondominated set. **b** Reducing the objective space around y^c. **c** Search within the reduced objective space

is $\boldsymbol{y^c}$. We find the two extreme nondominated points; $\boldsymbol{y^a}$ and $\boldsymbol{y^e}$ (step 1), and we find $\boldsymbol{y^c}$ minimizing the function passing through $\boldsymbol{y^a}$ and $\boldsymbol{y^e}$ (step 2). Since $\boldsymbol{y^c}$ is preferred to both its adjacent points $\boldsymbol{y^b}$ and $\boldsymbol{y^e}$, we reduce the objective space to the two triangles shown in Fig. 2b (steps 4 and 5). We discover that there are no new points in the left triangle (step 6). We next find $\boldsymbol{y^d}$ in the right triangle (step 8). The DM prefers $\boldsymbol{y^c}$ to $\boldsymbol{y^d}$ and we further reduce the objective space accordingly as shown in Fig. 2c (step 9). Since there are no new points in any of the two triangles, we conclude that $\boldsymbol{y^c}$ is the most preferred point (step 10).

Note that there might be several efficient solutions that correspond to the same most preferred point obtained. Our approach terminates with one of those most preferred solutions.

3.2 Finding adjacent points

During IA, we find adjacent points of $\boldsymbol{y}'$ both in the original (in steps 4 and 5) and the reduced objective spaces (in steps 6 and 8). We modify the method presented in Tezcaner and Köksalan (2011) to find the adjacent points. Our modification is in the initialization step to avoid searching for previously-found nondominated points.

We first explain the steps to find the left adjacent point of $\boldsymbol{y}'$ in the original objective space. Let S be a set that stores all nondominated points found in IA so far.

Step A.0. Let A $= \left\{\boldsymbol{y^i} | \boldsymbol{y^i} \in S, y_1^i < y_1'\right\}$. If $A = \emptyset$, find the left extreme nondominated point, $\boldsymbol{y^{LE}}$ minimizing $U(\boldsymbol{x}) = (1-\rho)z_1(\boldsymbol{x}) + \rho z_2(\boldsymbol{x})$ where ρ is as before. Set $\boldsymbol{y^L} = \boldsymbol{y^{LE}}$

and $w' = \frac{y_2^L - y_2'}{(y_2^L - y_2') + (y_1' - y_1^L)}$. If $A \neq \emptyset$, set $w' = \min_{\boldsymbol{y^i} \in A} \left\{ \frac{y_2^i - y_2'}{(y_2^i - y_2') + (y_1' - y_1^i)} \right\}$ and $\boldsymbol{y^L} = \operatorname{argmin}_{\boldsymbol{y^i} \in A} \left\{ \frac{y_2^i - y_2'}{(y_2^i - y_2') + (y_1' - y_1^i)} \right\}$.

Step A.1. Find the solution that minimizes $U(\boldsymbol{x}) = w' z_1(\boldsymbol{x}) + (1 - w') z_2(\boldsymbol{x})$ and denote the corresponding objective function vector as $\boldsymbol{y^{min}}$. If $\boldsymbol{y^{min}} = \boldsymbol{y^L}$ or $\boldsymbol{y}'$, go to Step A.2. Otherwise, let $\boldsymbol{y^L} = \boldsymbol{y^{min}}$, $w' = \frac{y_2^L - y_2'}{(y_2^L - y_2') + (y_1' - y_1^L)}$ and go to Step A.1.

Step A.2. The left adjacent point of $\boldsymbol{y}'$ is $\boldsymbol{y^L}$.

In Step A.0, we set the current left adjacent point of $\boldsymbol{y}'$ to the left extreme nondominated point if there are no other nondominated points in the region searched. In case there are other points in the searched region, we examine the slopes of the lines joining each such point to $\boldsymbol{y}'$ and select the point that corresponds to the line that has the least-negative slope. We set w' to the absolute value of the slope of the line joining $\boldsymbol{y}'$ with its current left adjacent point. In Step A.1, we keep updating the current left adjacent point of $\boldsymbol{y}'$, each time minimizing $U(\boldsymbol{x})$ formed with a different w'. We terminate the algorithm when no new point exists.

A similar algorithm is used to find the right adjacent point of $\boldsymbol{y}'$ in the original objective space. The main difference is that the right adjacent point of $\boldsymbol{y}'$ is the nondominated point that is associated with the line having the most-negative slope, considering all lines joining candidate points to $\boldsymbol{y}'$.

To find the left adjacent point of $\boldsymbol{y}'$ inside the left triangle defined by bounds UB^1 and UB^2, we also follow steps A.0 to A.2. In this case A $= \left\{ \boldsymbol{y^i} | \boldsymbol{y^i} \in S, y_1^i < UB^1, y_2^i \leq UB^2 \right\}$ and we use the additional constraints $z_k(\boldsymbol{x}) \leq UB^k, k = 1, 2$ each time we search for a solution minimizing $U(\boldsymbol{x})$.

Finding the right adjacent point of $\boldsymbol{y}'$ inside the right triangle is similar.

3.3 Reducing the objective space further

During IA, we search for both supported and unsupported nondominated points. Supported nondominated points can be found by optimizing an objective function that is a linear combination of the two objectives. Thus, the problem reduces to its single objective version, making the use of problem-specific algorithms possible, if available. On the other hand, searching for unsupported nondominated points may turn out to be computationally hard because of the constraints $y_k^i \leq UB^k, k = 1, 2$, that need to be added to the original problem. These additional constraints may cause additional computational difficulty. To improve the computational effort for finding unsupported nondominated points, Tuyttens et al. (2000) develop further constraints to reduce the search region. In this section, we present Tuyttens et al.'s approach and improve it further to reduce a larger area in the search region.

Tuyttens et al. study the so called *two phase algorithm* on the biobjective assignment problem. In this algorithm, the nondominated frontier is generated in two phases. In the first phase, the supported nondominated points are generated (see Fig. 3a). This phase is computationally relatively easy because each supported nondominated point can be obtained by minimizing a suitable linear combination of the two objectives. In the second phase, unsupported nondominated points are found. To find the unsupported points, the nondominated region between each adjacent supported nondominated point pair (see Fig. 3b) is searched.

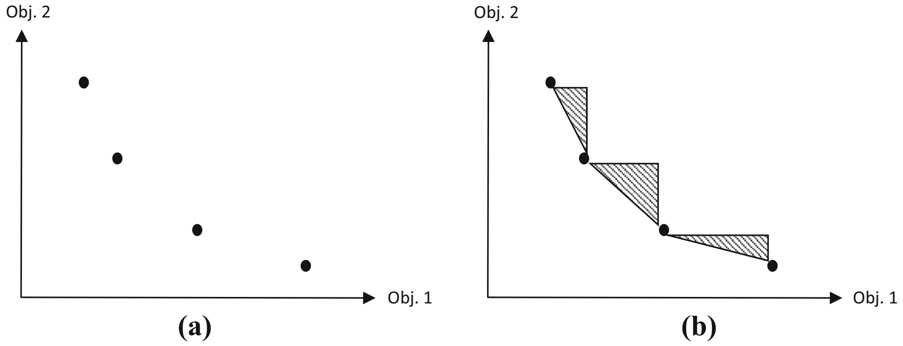

Fig. 3 The two phase algorithm. **a** Phase one. **b** Phase two

Tuyttens et al. propose an additional constraint that can be used to narrow down the search space in the second phase. This constraint reduces the dominated part of the triangles between supported nondominated points as illustrated in Fig. 4. Suppose two new unsupported nondominated points, $\boldsymbol{y}^{\mathbf{3}}$ and $\boldsymbol{y}^{\mathbf{4}}$, are found during the search between the two supported nondominated points, $\boldsymbol{y}^{\mathbf{1}}$ and $\boldsymbol{y}^{\mathbf{2}}$. From the definition of dominance, we know that the shaded region is dominated by the unsupported nondominated points. Tuyttens et al. construct a line joining the two supported nondominated points, $\boldsymbol{y}^{\mathbf{1}}$ and $\boldsymbol{y}^{\mathbf{2}}$, and move it in the direction of dominated points. They search for the first point where the line has no intersection with the nondominated region inside the triangle. The idea is to cover as big a portion of the shaded region as possible using a linear constraint. In the example in Fig. 4, there are three possibilities for this line to pass through. These are the points that have the worst objective values of the two neighboring nondominated points. Przybylski et al. (2008) refer to these points as local nadir points. We demonstrate the local nadir points, $\boldsymbol{\eta}^{\mathbf{1}}$, $\boldsymbol{\eta}^{\mathbf{2}}$ and $\boldsymbol{\eta}^{\mathbf{3}}$, with values $\left(\eta_1^1, \eta_2^1\right) = \left(y_1^3, y_2^1\right)$, $\left(\eta_1^2, \eta_2^2\right) = \left(y_1^4, y_2^3\right)$, and $\left(\eta_1^3, \eta_2^3\right) = \left(y_1^2, y_2^4\right)$, and the lines that pass through them in Fig. 4b.

Among these three possibilities, only the line through $\boldsymbol{\eta}^{\mathbf{2}}$ does not have an intersection with the nondominated regions. Therefore, the value of this line is set as an upper bound $\left(wz_1(\boldsymbol{x}) + (1-w)\, z_2(\boldsymbol{x}) \leq wy_1^4 + (1-w)\, y_2^3 = UB^c\right)$ as shown in Fig. 4c, where $w = \frac{y_2^1 - y_2^2}{\left(y_2^1 - y_2^2\right) + \left(y_1^2 - y_1^1\right)}$. Przybylski et al. (2008) modify this bound exploiting the fact that all objective coefficients are integer valued.

Although constraint $wz_1(\boldsymbol{x}) + (1-w)\, z_2(\boldsymbol{x}) \leq UB^c$ reduces a part of the dominated set, a dominated region still remains unaccounted for. We improve this bound in order to further reduce the unaccounted dominated region with linear constraints. We next develop an algorithm to determine a number of upper bounds, which we refer to as *weighted upper bounds*, in the triangles. Throughout the procedure below, we search within the triangle defined by the two nondominated points $\boldsymbol{y}^{\mathbf{1}}$, $\boldsymbol{y}^{\mathbf{2}}$, and their nadir point, where we assume $y_1^1 < y_1^2$ and $y_2^1 > y_2^2$, without loss of generality. Let w be as above.

3.3.1 W-UB Procedure

Step U.0. Find local nadir points of all neighboring nondominated points within the triangle and place them in set L. Denote the nadir point l as $\boldsymbol{\eta}^{\boldsymbol{l}}$.
Step U.1. Find the local nadir point that gives the largest combined objective value, $\boldsymbol{\eta}^* = \operatorname*{argmax}_{\boldsymbol{\eta}^l \in L} \left\{w\eta_1^l + (1-w)\eta_2^l\right\}$. Set $\boldsymbol{\eta}^{\mathbf{0}} = \boldsymbol{\eta}^*$.

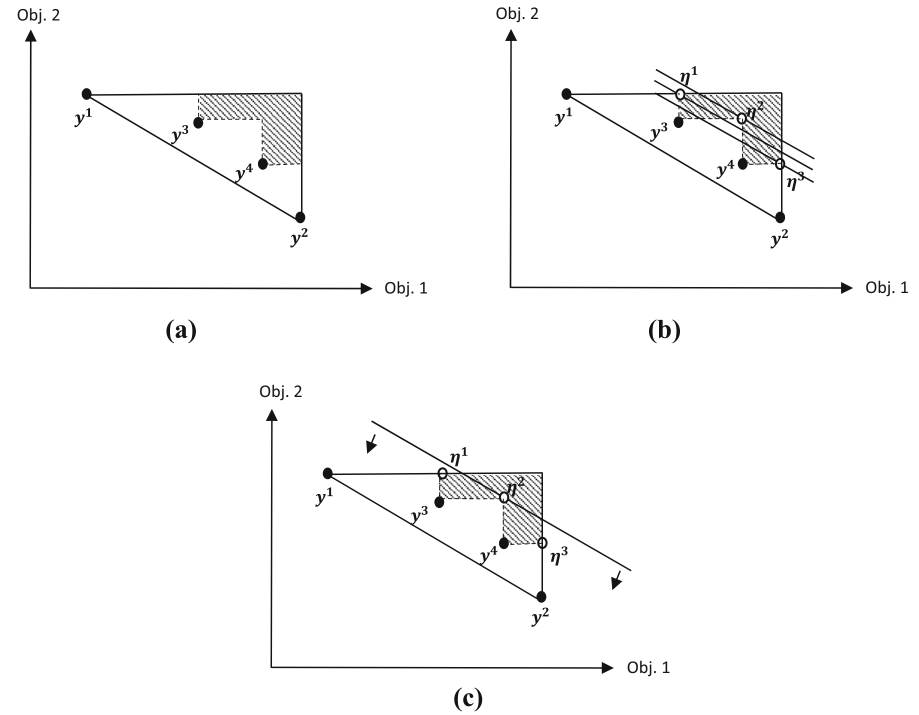

Fig. 4 The additional constraint in phase two. **a** Dominated regions. **b** Local nadir points. **c** Additional constraint

Step U.2. Find the left adjacent nadir point, $\boldsymbol{\eta}^{\mathbf{L0}}$, of $\boldsymbol{\eta}^{\mathbf{0}}$, treating the problem as a maximization problem. If there are no left adjacent nadir points in the search space, set $\boldsymbol{\eta}^{\mathbf{0}} = \boldsymbol{\eta}^*$ and go to Step U.3. Otherwise, define a new weighted upper bound, $w'z_1(\boldsymbol{x}) + \left(1 - w'\right) z_2(\boldsymbol{x}) \leq UB\left(w'\right) = w'\eta_1^0 + \left(1 - w'\right)\eta_2^0$ where $w' = \frac{\eta_2^{L0}-\eta_2^0}{(\eta_2^{L0}-\eta_2^0)+(\eta_1^0-\eta_1^{L0})}$. If $\eta_2^{L0} = y_2^1$, set $\boldsymbol{\eta}^{\mathbf{0}} = \boldsymbol{\eta}^*$ and go to Step U.3. If $\eta_2^{L0} \neq y_2^1$, set $\boldsymbol{\eta}^{\mathbf{0}} = \boldsymbol{\eta}^{\mathbf{L0}}$ and go to Step U.2.

Step U.3. Find the right adjacent nadir point, $\boldsymbol{\eta}^{\mathbf{R0}}$, of $\boldsymbol{\eta}^{\mathbf{0}}$, treating the problem as a maximization problem. If there are no right adjacent nadir points in the search space, terminate the algorithm. Otherwise, define a new weighted upper bound $w'z_1(\boldsymbol{x}) + \left(1 - w'\right) z_2(\boldsymbol{x}) \leq UB\left(w'\right) = w'\eta_1^0 + \left(1 - w'\right)\eta_2^0$ where $w' = \frac{\eta_2^0-\eta_2^{R0}}{(\eta_2^0-\eta_2^{R0})+(\eta_1^{R0}-\eta_1^0)}$. If $\eta_1^{R0} = y_1^2$, terminate the algorithm. If $\eta_1^{R0} \neq y_1^2$, set $\boldsymbol{\eta}^{\mathbf{0}} = \boldsymbol{\eta}^{\mathbf{R0}}$ and go to Step U.3.

We may visualize the set of local nadir points, L, as nondominated points of a problem where more is better (maximization) in each criterion. With this analogy, steps U.2 and U.3 of the *W-UB procedure* finds the left and right adjacent points of a given nondominated (local nadir) point where all points are readily available. To find the left adjacent point of $\boldsymbol{\eta}^{\mathbf{0}}$, we check the line that passes through $\boldsymbol{\eta}^{\mathbf{0}}$ and any other point that has the first objective value smaller than that of $\boldsymbol{\eta}^{\mathbf{0}}$. The (local nadir) point corresponding to the line with the steepest slope (most negative) is the left adjacent point. The right adjacent point of $\boldsymbol{\eta}^{\mathbf{0}}$ is found in a similar manner on the right side of $\boldsymbol{\eta}^{\mathbf{0}}$ using the line that has the flattest slope (least negative) among the candidate lines. Specifically, we find the left and right adjacent points of $\boldsymbol{\eta}^{\mathbf{0}}$ as follows:

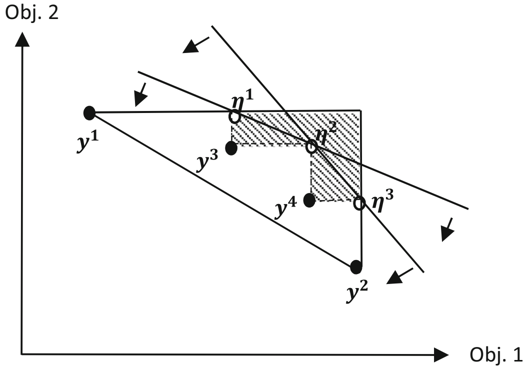

Fig. 5 Additional constraints in phase two

$$\eta^{L0} = \operatorname{argmax}_{\eta^l \in Q}\left\{\frac{\eta_2^l - \eta_2^0}{\left(\eta_2^l - \eta_2^0\right) + \left(\eta_1^0 - \eta_1^l\right)}\right\} \quad \text{where } \mathrm{Q} = \left\{\eta^l | \eta^l \in L, \eta_1^l < \eta_1^0\right\}.$$

$$\eta^{R0} = \operatorname{argmin}_{\eta^l \in Q}\left\{\frac{\eta_2^0 - \eta_2^l}{\left(\eta_2^0 - \eta_2^l\right) + \left(\eta_1^l - \eta_1^0\right)}\right\} \quad \text{where } \mathrm{Q} = \left\{\eta^l | \eta^l \in L, \eta_1^l > \eta_1^0\right\}.$$

These weighted upper bounds impose cuts in the feasible space eliminating a larger portion of the dominated space. Using this procedure, we replace the bound given in the example of Fig. 4c with the two bounds shown in Fig. 5. The procedure to generate such bounds may be used to reduce the solution space in any biobjective integer program to improve computational efficiency.

4 Generalized biobjective traveling salesperson problem

We developed IA to find the most preferred point of a DM for any biobjective integer program. In this section, we demonstrate it on the generalized BOTSP.

Most of the literature on multiobjective traveling salesperson problem (MOTSP) assumes that each node pair is connected with a single edge that is naturally efficient. This is not a realistic assumption, since there would typically be a number of efficient edges each better than another efficient edge in at least one objective, under the presence of multiple objectives. We refer to the case where each node is connected by a number of efficient edges as the generalized MOTSP. The MOTSP with a single edge between nodes is a special case of the generalized MOTSP. More information on the generalized MOTSP can be found in Tezcaner and Köksalan (2011).

MOTSP with a single edge between nodes is NP-hard (Ehrgott 2005, p. 279). Mostly, heuristics have been developed for the solution of this problem. Lust and Teghem (2010) classify these solution approaches and develop a new method, two-phase Pareto local search, for MOTSP. Paquete and Stützle (2003) develop a two-phase local search method and Jaszkiewicz and Zielniewicz (2009) consider a Pareto memetic algorithm using path relinking and Pareto local search. Jozefowiez et al. (2008), Karademir (2008) and Bérubé et al. (2009) consider a special traveling salesperson problem (TSP), TSP with profits. Jozefowiez et al. approximate the nondominated frontier by an evolutionary algorithm (EA), Karademir proposes a genetic algorithm and Bérubé generates the nondominated frontier using the ε-constraint method (see Chankong and Haimes 1983). Hansen (2000) uses a scalarizing function to solve MOTSP.

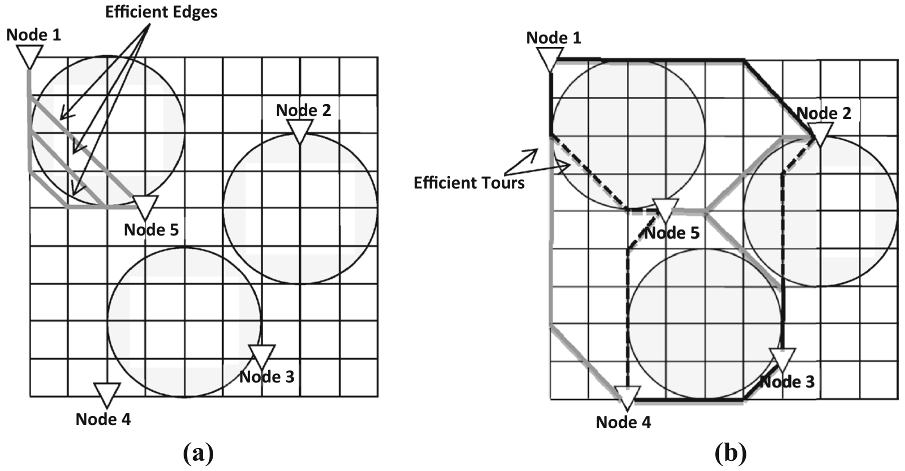

Fig. 6 Generalized BOTSP. **a** Efficient edges. **b** Efficient tours

Special cases of MOTSP are studied in Özpeynirci and Köksalan (2009, 2010). To the best of our knowledge, Tezcaner and Köksalan's (2011) study is the only one that defines and addresses the generalized MOTSP. It develops a general interactive algorithm to find the most preferred point for a biobjective integer program and implements it for the generalized BOTSP.

We demonstrate a special generalized BOTSP example in Fig. 6. Here we consider the three circular areas as regions to be avoided (as one of the objectives); central parts more important to be avoided than those towards the circumferences. The other objective we consider is to minimize total distance traveled. We have in total 5 nodes to visit in a terrain divided into grids (equal-length intervals in both x and y axes). The vehicle is assumed to move horizontally, vertically, or diagonally through a grid. There are three efficient edges between nodes 1 and 5 as shown in Fig. 6a. These efficient edges form due to the conflict between the two objectives. The efficient edge that totally avoids the threat region has the largest distance, whereas the edge with the shortest distance exposes the vehicle to the biggest threat. Figure 6b shows two efficient tours that pass through all nodes. The dashed lines correspond to the first tour (1-5-4-3-2-1 or equivalently 1-2-3-4-5-1 since the graph is assumed to be undirected) and the light solid lines correspond to the second tour (1-4-3-5-2-1 or 1-2-5-3-4-1). The dark solid lines are common to both tours.

In a realistic setting, an edge between a pair of nodes could correspond to a path that passes through a set of subnodes. For example, subnodes would correspond to the corner points of grids (grid points) in Fig. 6 and to the intersection points of highways in a road network. Therefore, an edge corresponds to a path between a pair of nodes passing through a subset of subnodes. Then, finding the efficient edges between a node pair requires solving a BOSPP between that node pair. Although the BOSPP is NP-complete (Ehrgott 2005, p. 222), for practical size problems the number of efficient edges is typically small (see Müller-Hannemann and Weihe 2006) and finding them is manageable in general. Once the efficient edges are formed, they can be used as input and the generalized BOTSP can be formulated as below.

Let $G = (N, E)$ be an undirected graph with node set $N = \{1, 2, \ldots\}$, e_{ijr} be efficient edge r connecting node pair (i, j), R_{ij} be the index set of efficient edges between node pair

(i, j), and $E = \left\{e_{121}, e_{122}, \ldots, e_{|N|,|N|-1,\left|R_{|N|,|N|-1}\right|}\right\}$ be the set of all efficient edges. Let x_{ijr} be the binary decision variable that indicates whether efficient edge $r \in R_{ij}$ is used in the tour or not, c^k_{ijr} denote the value of objective function k of efficient edge $r \in R_{ij}$, and $P = \{(i, j) | i \in N, j \in N, i \neq j\}$ be the set of all node pairs. The problem can be formulated as:

(G-BOTSP)

$$\text{Min } z_1(\boldsymbol{x}) = \sum_{(i,j)\in P} \sum_{r\in R_{ij}} c^1_{ijr} x_{ijr} \tag{9}$$

$$\text{Min } z_2(\boldsymbol{x}) = \sum_{(i,j)\in P} \sum_{r\in R_{ij}} c^2_{ijr} x_{ijr} \tag{10}$$

Subject to:

$$\sum_{j\in N} \sum_{r\in R_{ij}} x_{ijr} = 1 \quad i \in N \tag{11}$$

$$\sum_{i\in N} \sum_{r\in R_{ij}} x_{ijr} = 1 \quad j \in N \tag{12}$$

$$u_i - u_j + \sum_{r\in R_{ij}} (|N| - 1) x_{ijr} + \sum_{r\in R_{ji}} (|N| - 3) x_{jir} \leq |N| - 2 \quad i, j \in N \backslash \{1\}, i \neq j \tag{13}$$

$$1 \leq u_i \leq |N| - 1 \quad i \in N \backslash \{1\} \tag{14}$$

$$x_{ijr} \in \{0, 1\} \quad (i, j) \in P, r \in R_{ij} \tag{15}$$

The two objectives, $z_1(\boldsymbol{x})$ and $z_2(\boldsymbol{x})$, are any two sum-type objectives. They are minimized in (9) and (10), respectively. We assume that $c^k_{ijr} \geq 0$ for all $(i, j) \in P, k = 1, 2$, and $r \in R_{ij}$. Equations (11) and (12) ensure departure from and arrival to each node, respectively. There are different possible formulations for subtour elimination constraints. Laporte (1992) reviews the algorithms developed for TSP. He covers some subtour elimination constraints including the formulation introduced by Miller et al. (1960), which was strengthened later. Constraints (13) and (14) are the strengthened version of Miller et al.'s formulation. This formulation requires a smaller number of additional constraints compared to other subtour elimination constraints. However it requires including $|N| - 1$ positive decision variables in the formulation.

In order to solve (G-BOTSP), we first need to find all efficient edges between all node pairs. An algorithm developed for solving multiobjective shortest path problems can be used for this purpose. Alternatively, we may employ the ε-constraint method (see Chankong and Haimes 1983) where one objective is optimized treating the other as a constraint to satisfy an aspiration level, ε. Changing ε systematically all efficient edges can be found.

During IA, we find nondominated points both in the original and reduced objective spaces. We next discuss these two cases.

4.1 Finding nondominated points in the original objective space

In Steps 1 through 5 of IA, we find tours for the combined objective $U(\boldsymbol{x}) = wz_1(\boldsymbol{x}) + (1 - w)z_2(\boldsymbol{x})$ without imposing additional constraints in the objective space. The only difference of this formulation from the single-objective TSP is the existence of multiple efficient edges between node pairs. Note that there is a unique efficient edge between each node pair (i, j) that minimizes $U(\boldsymbol{x})$. Recall that c_{ijr}^k denotes the k^{th} objective value of efficient edge $r \in R_{ij}, k = 1, 2$. Using the weights of $U(\boldsymbol{x})$, we create a dummy edge for each node pair (i, j) with the following edge cost:

$$c_{ij}(w) = \min_{r \in R_{ij}} \left\{ wc_{ijr}^1 + (1 - w)\, c_{ijr}^2 \right\} \qquad (i, j) \in P$$

We then solve the TSP using the following objective function, where h_{ij} is a binary variable to indicate whether or not the edge connecting node pair (i, j) is used.

$$\text{Min } z(\boldsymbol{h}) = \sum_{(i,j) \in P} c_{ij}(w) h_{ij}$$

We solve the resulting single-objective TSP using the Concorde software (see http://www.math.uwaterloo.ca/tsp/concorde.html) inputting the dummy edge cost for each node pair (i, j).

4.2 Finding nondominated points in the reduced objective space

In Steps 6–9 of IA, we have the following two constraints in addition to constraints (11)–(15).

$$\sum_{(i,j) \in P} \sum_{r \in R_{ij}} c_{ijr}^1 x_{ijr} \leq UB^1 \tag{16}$$

$$\sum_{(i,j) \in P} \sum_{r \in R_{ij}} c_{ijr}^2 x_{ijr} \leq UB^2 \tag{17}$$

As mentioned before, the nondominated points in the reduced objective space are unsupported nondominated points. As we find new unsupported points, we can introduce cuts of the form below using *W-UB Procedure* to further reduce the objective space:

$$w' z_1(\boldsymbol{x}) + \left(1 - w'\right) z_2(\boldsymbol{x}) \leq UB\left(w'\right) \tag{18}$$

The introduction of (16)–(18) destroys the input structure required by Concorde, making it inapplicable. In the absence of Concorde, we may use any integer programming solver to find unsupported nondominated points minimizing $U(\boldsymbol{x}) = w' z_1(\boldsymbol{x}) + (1 - w') z_2(\boldsymbol{x})$ subject to constraints (11)–(18). In our application, we use the CPLEX solver.

5 Demonstration and computational results

In this section, we first provide an example problem to demonstrate our algorithm. The problem is a special case of the generalized BOTSP, the unmanned air vehicle (UAV) route planning problem. We then present computational results for randomly generated instances of the generalized BOTSP.

5.1 An example

In the UAV route planning problem, the path the vehicle follows through a terrain visiting all predetermined targets is searched for. This problem is a special case of the generalized BOTSP. We consider a number of nodes (targets) to visit where each node is connected by

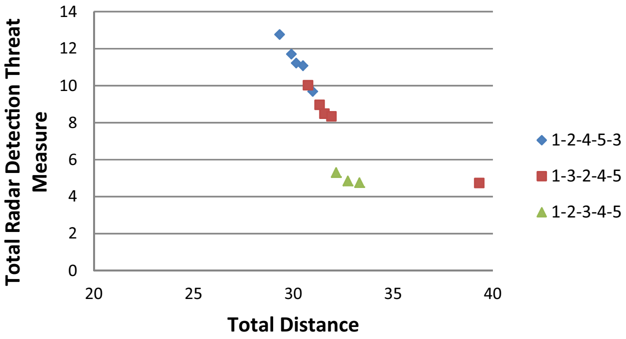

Fig. 7 Nondominated set of the UAV route planning problem

a number of efficient edges. We assume there are two minimization-type objectives: total distance travelled and total radar detection threat.

The approaches that have been developed for the route planning problem of UAVs are mostly heuristic in nature (see, for example, Zheng et al. 2003; Foo et al. 2009; Wu et al. 2009; Waldock and Corne 2012). Many of them linearly combine the objectives forming a single objective and search for an "optimal" solution rather than addressing the tradeoffs between multiple efficient solutions. To the best of our knowledge, Tezcaner and Köksalan's (2011) approach is the only implementation of an interactive approach to biobjective UAV route planning.

We demonstrate our algorithm using the UAV route planning problem introduced by Tezcaner and Köksalan (2011), which they solved assuming an underlying linear preference function. In this problem, there are 5 nodes and 3 radars. Here we assume that the DM has an underlying Tchebycheff preference function $f(\boldsymbol{y}) = \max[0.33\left(y_1 - y_1^{**}\right), 0.67\left(y_2 - y_2^{**}\right)]$ to be minimized where $\boldsymbol{y}^{**} = (y_1^{**}, y_2^{**})$ is a utopia point having values that are less than or equal to the best possible values in each objective. We set $\boldsymbol{y}^{**} = (0{,}0)$. We pretend that we do not know f and use it only to simulate the preferences of the DM in responding to pairwise comparisons. For demonstration purposes, we provide the nondominated set of the problem in the objective space in Fig. 7. There are three efficient tours and a set of nondominated points for each tour. The efficient solutions of different tours are demonstrated by different shapes in Fig. 7. The objective function values of each nondominated point are given in Table 1. We see that the most preferred point of the DM is $\boldsymbol{y}^* = (29.312, 12.764)$ when we plug its objective values in f. In our algorithm, we pretend that we do not know the nondominated set, f, or the most preferred point.

Before running the algorithm, we search for all efficient edges between all node pairs using the ε-constraint method and find 44 of them.

In the first step of IA, we find the two extreme nondominated points, $\boldsymbol{y}^{LE} = (29.312, 12.764)$ and $\boldsymbol{y}^{RE} = (39.312, 4.744)$. We then go to Step 2 and find the nondominated point, $\boldsymbol{y}' = (32.140, 5.305)$, corresponding to the solution minimizing $U(\boldsymbol{x})$ for $w' = 0.445$. In Step 4, we find the left adjacent point of $\boldsymbol{y}'$, $\boldsymbol{y}^L = (29.312, 12.764)$, and ask the DM to compare $\boldsymbol{y}'$ with $\boldsymbol{y}^L$. f implies that the DM prefers $\boldsymbol{y}^L$. Thus, we set the two bounds for the right triangle as $z_2(\boldsymbol{x}) \leq 12.764$, $z_1(\boldsymbol{x}) \leq 32.140\text{-}\delta$. The current most preferred point is set to $\boldsymbol{y}' = (29.312, 12.764)$ and since $\boldsymbol{y}' = \boldsymbol{y}^{LE}$, we go to Step 8. We then search for the right adjacent point of $\boldsymbol{y}' = (29.312, 12.764)$ within the triangle and we find the three unsupported nondominated points, (31.898, 8.344), (31.554, 8.491), and (30.726, 10.029), successively.

Table 1 Nondominated points

Nondominated point	Distance	Radar detection threat
1	29.312	12.764
2	29.898	11.704
3	30.140	11.226
4	30.484	11.079
5	30.726	10.029
6	30.968	9.688
7	31.312	8.969
8	31.554	8.491
9	31.898	8.344
10	32.140	5.305
11	32.726	4.851
12	33.312	4.757
13	39.312	4.744

In this small example, we end up finding 9 of the 13 nondominated points and ask the DM to compare 4 nondominated point pairs in order to find the most preferred point. We next give some results for larger instances.

5.2 Computational results

To further demonstrate its performance, we test IA on several randomly generated instances. We consider 5, 8, and 15-node generalized BOTSPs, in which we generate 20 efficient edges between each node pair. For each problem type, we create 10 different instances randomly generating the objective values of the edges from a uniform distribution with range [0, 100].

We solve all the problems assuming that the DM has an underlying Tchebycheff preference function $f(y) = max[\lambda\left(y_1 - y_1^{**}\right), (1-\lambda)\left(y_2 - y_2^{**}\right)]$ to be minimized, as above. We solve the instances for three different weight values, $\lambda = 0.2$, 0.5, and 0.8. We implement the algorithm in C++ and run the tests on a PC with Intel Core i7-3770 CPU, 8 GB RAM.

In order to report the size of the full nondominated set, we solve the whole problem using the ε-constraint method. We report these in Table 2 in order to provide a sense of the magnitude of the problem. We also present the averages and the ranges of the number of nondominated points found, as well as the number of comparisons made by our algorithm in Table 2. There is a large deviation in the total number of nondominated points between the 10 instances of each problem size as seen from the ranges in Table 2. This in turn leads to large variations in the number of nondominated points found and the number of comparisons made. The algorithm finds only a small percentage (2–15 %) of the nondominated points in all problems and eliminates roughly a third of the generated nondominated points directly, without asking the DM.

Although there are variations in solution times, all problems are solved fast. For 5, 8, and 15-node problems, the average solution durations are 17.429, 31.546 and 95.997 s, respectively. As expected, the larger the problem size, the longer the execution times.

Table 2 Results for 5, 8, and 15-node problems

	Average [min., max.] number of total nondominated points	λ	Average number of nondominated points found	[Min., max.] number of nondominated points found	Average number of comparisons	[Min., max.] number of comparisons
5-Node problems	291.1 [228, 347]	0.2	31.0	[20, 46]	20.1	[12, 31]
		0.5	41.8	[14, 82]	28.1	[6, 58]
		0.8	40.9	[23, 74]	30.8	[16, 60]
8-Node problems	777.5 [571, 972]	0.2	33.7	[21, 48]	20.7	[12, 35]
		0.5	47.6	[25, 81]	30.6	[15, 60]
		0.8	49.4	[22, 75]	33.5	[14, 56]
15-Node problems	2286.7 [1567, 3064]	0.2	60.3	[37, 95]	38.9	[23, 62]
		0.5	51.4	[19, 84]	30.1	[11, 55]
		0.8	67.7	[29, 158]	47.8	[19, 111]

6 Conclusions

In this study, we develop an interactive algorithm for bicriteria integer programs that finds the most preferred point of a DM whose preferences are consistent with a quasiconvex function. The algorithm gradually reduces the search region around the most preferred point of the DM considering his/her preferences. We ask for comparison between adjacent points, which can be supported or unsupported depending on the search region. To reduce the computational effort in finding unsupported nondominated points, we improve the bound developed by Tuyttens et al. (2000) and eliminate more of the search region.

We apply the interactive algorithm to the generalized BOTSP that has multiple efficient edges between nodes and test the algorithm on randomly generated instances with 5, 8, and 15 nodes. The results show fast convergence to the most preferred point of the DM avoiding unnecessary preference comparisons.

During the interactive algorithm, the majority of the computational effort is devoted to the search for unsupported nondominated points in reduced objective spaces. In its current version, the interactive algorithm can be applied to any biobjective integer program. When the algorithm is applied to different problem types, problem specific approaches can be used to improve the computational efficiency of the search for unsupported nondominated points. Alternatively, for larger problem instances that are computationally prohibitive even in the single-objective case, heuristics can be developed for the search for unsupported nondominated points. Another possibility for handling the computational complexity could be to develop an approximation algorithm by defining a reasonable worst-case performance bound. For example, a larger than warranted portion of the objective space can be reduced each time a point is preferred to its adjacent points, allowing for a small deviation from the most preferred point. Development of such heuristics is a promising area for future research.

Currently, our algorithm is applicable to any biobjective integer program. Generalizing for mixed integer programs is another idea for future research. One main difficulty would be the existence of continuous nondominated frontiers. In this case, the selection of the point pairs to present to the DM should be reconsidered. Additionally, the space reduction after the DM's preferences might be modified considering the shape of the frontier. We should also readdress the stopping condition due to the continuous frontier.

References

Bérubé, J. F., Gendreau, M., & Potvin, J. Y. (2009). An exact ε-constraint method for Bi-objective combinatorial optimization problems—Application to the traveling salesman problem with profits. *European Journal of Operational Research*, *194*(1), 39–50.

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective decision making: Theory and methodology*. New York: North-Holland.

Ehrgott, M. (2005). *Multicriteria optimization* (Vol. 2). Berlin: Springer.

Foo, J. L., Knutzon, J., Kalivarapu, V., Olver, J., & Winer, E. (2009). Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization. *Journal of Aerospace Computing, Information and Communication*, *6*(4), 271–290.

Hansen, M. P. (2000). Use of substitute scalarizing functions to guide a local search based heuristic: The case of moTSP. *Journal of Heuristics*, *6*, 419–431.

Jaszkiewicz, A., & Zielniewicz, P. (2009). Pareto memetic algorithm with path relinking for bi-objective traveling salesperson problem. *European Journal of Operational Research*, *193*(3), 885–890.

Jozefowiez, N., Glover, F., & Laguna, M. (2008). Multi-objective meta-heuristics for the traveling salesman problem with profits. *Journal of Mathematical Modelling and Algorithms*, *7*(2), 177–195.

Karademir, S. (2008). *A genetic algorithm for the biobjective traveling salesman problem with profits*. M.S. thesis, Department of Industrial Engineering, Middle East Technical University.

Korhonen, P., Wallenius, J., & Zionts, S. (1984). Solving the discrete multiple criteria problem using convex cones. *Management Science*, *30*(1), 1336–1345.

Köksalan, M. M., & Sagala, P. N. S. (1995). Interactive approaches for discrete alternative multiple criteria decision making with monotone utility functions. *Management Science*, *41*, 1158–1171.

Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(2), 231–247.

Lokman, B., Köksalan, M., Korhonen, P. J., & Wallenius, J. (2014). An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Annals of operations research*. doi:10.1007/s10479-014-1545-2

Lust, T., & Teghem, J. (2010). The multiobjective traveling salesman problem: A survey and a new approach. *Advances in Multi-Objective Nature Inspired Computing Studies in Computational Intelligence*, *272*, 119–141.

Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, *7*(4), 326–329.

Müller-Hannemann, M., & Weihe, K. (2006). On the cardinality of the Pareto set in bicriteria shortest path problems. *Annals of Operations Research*, *147*, 269–286.

Özpeynirci, Ö., & Köksalan, M. (2009). Multiobjective traveling salesperson problem on halin graphs. *European Journal of Operational Research*, *196*, 155–161.

Özpeynirci, Ö., & Köksalan, M. (2010). Pyramidal tours and multiple objectives. *Journal of Global Optimization*, *48*(4), 569–582.

Paquete, L., & Stützle, T. (2003). A two-phase local search for the biobjective traveling salesman problem. *Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO 2003),* LNCS 2632, (pp. 479–493). New York: Springer.

Przybylski, A., Gandibleux, X., & Ehrgott, M. (2008). Two phase algorithms for the Bi-objective assignment problem. *European Journal of Operations Research*, *185*, 509–533.

Ramesh, R., Karwan, M. H., & Zionts, S. (1990). An interactive method for bicriteria integer programming. *IEEE Transactions on Systems, Man, and Cybernetics*, *20*, 395–403.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application*. New York: Wiley.

Tezcaner, D., & Köksalan, M. (2011). An interactive algorithm for multi-objective route planning. *Journal of Optimization Theory and Applications*, *150*(2), 379–394.

Tuyttens, D., Teghem, J., Fortemps, Ph, & Van Nieuwenhuyse, K. (2000). Performance of the MOSA method for the bicriteria assignment problem. *Journal of Heuristics*, *6*, 295–310.

Waldock, A., & Corne, D. W. (2012). Exploiting prior information in multi-objective route planning. *Parallel Problem Solving from Nature XII Lecture Notes in Computer Science*, *7492*, 11–21.

Wu, P. P., Campbell, D., & Merz, T. (2009). On-board multi-objective mission planning for unmanned aerial vehicles. *Aerospace conference IEEE*.

Zheng, C., Ding, M., & Zhou, C. (2003). Real-time route planning for unmanned air vehicle with an evolutionary algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(1), 68–81.

Zionts, S. (1981). A multiple criteria method for choosing among discrete alternatives. *European Journal of Operational Research*, *7*(1), 143–147.