J Glob Optim (2013) 57:347–365
DOI 10.1007/s10898-012-9955-7

Finding all nondominated points of multi-objective integer programs

Banu Lokman · Murat Köksalan

Received: 8 December 2011 / Accepted: 28 June 2012 / Published online: 25 July 2012

Abstract We develop exact algorithms for multi-objective integer programming (MIP) problems. The algorithms iteratively generate nondominated points and exclude the regions that are dominated by the previously-generated nondominated points. One algorithm generates new points by solving models with additional binary variables and constraints. The other algorithm employs a search procedure and solves a number of models to find the next point avoiding any additional binary variables. Both algorithms guarantee to find all nondominated points for any MIP problem. We test the performance of the algorithms on randomly-generated instances of the multi-objective knapsack, multi-objective shortest path and multi-objective spanning tree problems. The computational results show that the algorithms work well.

Keywords Multiple criteria · Combinatorial optimization · Nondominated point

1 Introduction

Multi-objective integer programming (MIP) problems are hard to solve in general. There are special cases of MIP problems, widely referred as Multi-objective combinatorial optimization (MOCO) problems. MOCO has been a growing research area during the last decade. Single objective combinatorial problems have been widely studied in the past. The decision makers (DMs) usually have to deal with multiple conflicting objectives but generalizing the results of single objective problems to multiple objectives is not straightforward. Typically, the computational complexity increases substantially. The number of nondominated points

B. Lokman (✉)
Industrial Engineering Department, TED University, 06420, Ankara, Turkey
e-mail: banu.lokman@tedu.edu.tr

M. Köksalan
Industrial Engineering Department, Middle East Technical University,
06531 Ankara, Turkey
e-mail: koksalan@ie.metu.edu.tr

may be exponential in the problem size and identifying all nondominated points becomes intractable in this case. Heuristic approaches have been developed to find desirable solutions (see for example, Köksalan 1999; Ulungu et al. 1999; Phelps and Köksalan 2003; Ehrgott and Gandibleux 2004).

Ehrgott and Gandibleux (2000) review exact and approximation methods developed for MOCO problems. Early papers in MOCO mostly focused on finding supported nondominated points. Using weighted linear combinations of objectives, all supported nondominated points can be generated by systematically varying the weights. Typically, finding unsupported nondominated points is computationally harder for MOCO problems. These solutions cannot be reached by using a weighted linear combination of objectives. Generating only the supported nondominated points may also be a difficult task especially for large-sized MOCO problems.

Some authors separate the generation of the nondominated points into two phases. In the first phase, all supported nondominated points are generated using the weighted sum scalarization. In the second phase, all unsupported nondominated points are obtained by employing problem specific techniques. This approach has been applied to several biobjective combinatorial problems. Visée et al. (1998) proposed a two phase method and branch and bound procedures for the biobjective knapsack problem. Ramos et al. (1998) and Steiner and Radzik (2008) developed a two phase method to generate all nondominated trees for the biobjective spanning tree problem. Przybylski et al. (2010) worked on the two-phase method for MIP problems and experimented with three-objective assignment problems.

Mavrotas and Diakoulaki (2005) developed a branch and bound algorithm to find the extreme nondominated points for multiple objective linear programming (MOLP) having binary variables in addition to real-valued variables. They conducted experiments with two, three and four-objective problems.

There exist exact algorithms, especially for the multi-objective shortest path (MOSP) problem, adapted from the single objective methods. Martins (1984) proposed an algorithm based on the label setting method to generate all efficient paths of MOSP problem. He tested the performance of the algorithm on MOSP problem with two and four objectives. Tung and Chew (1992) developed an exact algorithm for MOSP problem which is a generalization of the label correcting method for the classical shortest path problem. They applied their algorithm on MOSP problem with three objectives. Guerriero and Musmanno (2001) also developed a label correcting method to generate the entire set of efficient paths and implemented the algorithm on MOSP problem with two, three and four objectives. Corley (1985) proposed an algorithm for the multi-objective spanning tree (MOST) problem which is a generalization of Prim (1957)'s algorithm developed for solving the single-objective spanning tree problem. Gomes da Silva et al. (2008) and Mavrotas et al. (2011) developed approaches for the special case of two-objective knapsack problems. These algorithms are problem specific and cannot be directly generalized to other problems.

Özlen and Azizoğlu (2009) developed an algorithm to generate all nondominated points for MIPs based on the epsilon constraint method. They do not conduct computational experiments but they demonstrate their algorithm on a three-objective assignment problem. Laumanns et al. (2006) also developed an algorithm to generate nondominated points based on the epsilon constraint method. In their experiments on multi-objective knapsack problems (MOKP), they end up solving a very large number of models for even small problems and their computation times are excessive. They also developed, a heuristic version but this version is capable of generating only a small percentage of the nondominated points for the same problems.

Sylva and Crema (2004) developed an exact algorithm for generating all nondominated points for MIPs. The algorithm keeps finding new nondominated points, one at a time. After finding a new nondominated point, a new model is constructed by adding new constraints and binary variables to the previous model. Then, the new model is solved to obtain the new nondominated point. Naturally, the task becomes impractical as the number of nondominated points increases. The algorithm of Sylva and Crema (2004) includes the full enumeration of the set of nondominated points which may be impossible especially for large-sized problems. Sylva and Crema (2007) developed another algorithm in order to find a well-dispersed subset of nondominated points for mixed MIPs based on their earlier approach.

In this paper, we develop two exact algorithms to generate all nondominated points for MIP problems efficiently. Our first algorithm finds the nondominated points iteratively by solving a model with additional variables and constraints at each iteration. Our algorithm improves the algorithm of Sylva and Crema (2004) by decreasing the number of additional constraints and binary variables. However, the improved algorithm still requires substantial computational effort as the number of nondominated points increases. Our second algorithm tries to reduce the complexity by imposing bounds on the objectives rather than adding additional constraints or binary variables at each iteration. Different from many of the previous exact methods, our methods are not restricted to bicriteria problems and are applicable for any number of objectives. We conduct extensive experiments with three and four-objective instances of several MIPs and show that the algorithms work well. To the best of our knowledge, this is the first study that conducts experiments to generate all nondominated points for four-objective MIPs.

We develop exact methods in Sect. 2, and demonstrate their performances in Sect. 3. We present our conclusions in Sect. 4.

2 Exact algorithms to generate all nondominated points

We develop two exact algorithms to find all nondominated points of MIP problems. We first give some background.

2.1 Definitions and some theory

A general multi-objective problem can be defined as:

$$
\begin{aligned}
&(P)\\
&\text{“Max”}\ \left\{z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_p(\mathbf{x})\right\}\\
&\text{subject to}\\
&\mathbf{x} \in \mathbf{X}\\
&\text{where}\\
&z_i(\mathbf{x}) = i\text{th objective function}\\
&\mathbf{x}:\ \text{decision vector}\\
&\mathbf{X}:\ \text{solution space}\\
&p:\ \text{the number of objective functions}
\end{aligned}
$$

The quotation marks are used as the maximization of a vector is not a well-defined mathematical operation.

The objective vector $\mathbf{z}(\mathbf{x}') = (z_1(\mathbf{x}'), z_2(\mathbf{x}'), \ldots, z_p(\mathbf{x}'))$ is said to *dominate* $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_p(\mathbf{x}))$ if $z_i(\mathbf{x}) \le z_i(\mathbf{x}')$ for all $i = 1, 2, \ldots, p$ and $\mathbf{z}(\mathbf{x}') \ne \mathbf{z}(\mathbf{x})$. If there does not exist such an $\mathbf{x}'$, then the decision vector, $\mathbf{x}$, is said to be *efficient* and the corresponding objective vector, $\mathbf{z}(\mathbf{x})$, is said to be *nondominated*. *Efficient (nondominated) frontier* is defined as the entire set of efficient solutions (nondominated points). It is well-known that maximizing a positive linear combination of the objectives yields an efficient solution. Such solutions are referred to as supported efficient solutions and their images in the objective space are called supported nondominated points.

2.2 Background

The algorithm of Sylva and Crema (2004) starts with a positive weight vector $\boldsymbol{\lambda} > 0$ and solves:

$$
\begin{aligned}
&(P_{\boldsymbol{\lambda}})\\
&\text{Max } \sum_{j=1}^{p} \lambda_j z_j(\mathbf{x})\\
&\text{subject to}\\
&\mathbf{x} \in \mathbf{X}
\end{aligned}
$$

After finding a new nondominated point, they revise $(P_{\boldsymbol{\lambda}})$ by adding p binary variables and $p+1$ constraints. If there are n previously-generated nondominated points, they solve $(P_{\boldsymbol{\lambda}}^{n})$ in order to find the $(n+1)th$ nondominated point assuming that all z_{tj} are integer valued for all $t = 1, \ldots, n$ and $j = 1, \ldots, p$.

$$
\begin{aligned}
&\left(P_{\boldsymbol{\lambda}}^{n}\right)\\
&\text{Max } \sum_{j=1}^{p} \lambda_j z_j(\mathbf{x})\\
&\text{subject to}\\
&z_j(\mathbf{x}) \geq \left(z_{tj}+1\right) y_{tj} - M\left(1 - y_{tj}\right) \quad \forall j \;\; \forall t\\
&\sum_{j=1}^{p} y_{tj} \geq 1 \quad \forall t\\
&y_{tj} \in \{0, 1\} \quad j = 1, \ldots, p \quad t = 1, \ldots, n\\
&\mathbf{x} \in \mathbf{X}
\end{aligned}
$$

In problem $(P_{\boldsymbol{\lambda}}^{n})$, $\mathbf{z}^t = (z_{t1}, z_{t2}, \ldots, z_{tp})$ denotes the tth nondominated point, M is a sufficiently large positive constant used to create a lower bound for $z_j(\mathbf{x})$ and y_{tj} is a binary variable forcing $z_j(\mathbf{x}) \geq z_{tj} + 1$ when it takes a value of 1. The constraints "$\sum_{j=1}^{p} y_{tj} \geq 1$" force the optimal solution to have an objective value larger than that of the tth nondominated point in at least one of the objectives, guaranteeing a new nondominated point different from any of the existing ones. The algorithm keeps adding binary variables and constraints until no feasible solution can be found.

2.3 Development of Algorithm 1

Our first algorithm improves the algorithm of Sylva and Crema (2004) by reducing the number of binary variables and constraints. The algorithm first arbitrarily selects a criterion, m, to be maximized throughout the solution process. Consider problem (P_m^0), that yields the nondominated point having the largest mth criterion value.

$$
\begin{aligned}
&\left(P_m^0\right)\\
&\text{Max } z_m(\mathbf{x}) + \varepsilon \sum_{j \neq m} z_j(\mathbf{x})\\
&\text{subject to}\\
&\mathbf{x} \in X
\end{aligned}
$$

where ε is a sufficiently small positive constant that prevents obtaining weakly nondominated but dominated solutions (see Steuer 1986, p. 425).

Let the nondominated point corresponding to the optimal solution of (P_m^0) be $\mathbf{z}^1 = (z_{11}, z_{12}, \ldots, z_{1p})$. By construction, the mth component of this vector, z_{1m}, corresponds to the maximum value of the mth criterion among all feasible solutions. Similarly, let $\mathbf{z}^n = (z_{n1}, z_{n2}, \ldots, z_{np})$ be the nondominated point obtained in iteration n, given that $n-1$ nondominated points having mth criterion values greater than or equal to z_{nm} have already been identified. The vectors $\mathbf{z}^t \quad t = 1, 2, \ldots, n-1$ include all nondominated points having mth criterion values strictly greater than z_{nm} but they may include only a subset of nondominated points having mth criterion values equal to z_{nm}. All the remaining nondominated points having mth criterion value equal to z_{nm} will be identified in the succeeding iterations before identifying any nondominated point that has mth criterion value strictly smaller than z_{nm}. Eventually, all nondominated points $\mathbf{z}^n \quad n = 1, 2, \ldots, N$ will be identified in the non-increasing order of the mth criterion value. Let the set of the nondominated points found in the first n iterations be $S_n = \{\mathbf{z}^t : 1 \le t \le n\}$.

By solving (P_m^0), we obtain the nondominated point with the best value of the selected criterion. Proposition 1 claims that we find the nondominated point having the $(n+1)th$ best value of the selected criterion by using the nondominated points with best n values of the selected criterion. We assume that all z_{tj} are integer valued for all $t = 1, \ldots, n$ and $j = 1, \ldots, p$.

Proposition 1 *For a sufficiently small $\varepsilon > 0$ and a sufficiently large $M > 0$, if all the nondominated points in $S_n = \{\mathbf{z}^t : 1 \le t \le n\}$ are known, then the optimal solution to (P_m^n) gives the nondominated point, $\mathbf{z}^{n+1} = (z_{(n+1)1}, z_{(n+1)2}, \ldots, z_{(n+1)p})$, such that $z_{(n+1)m} \le z_{tm}$ for all $t = 1, 2, \ldots, n$. If (P_m^n) is infeasible, then S_n contains all nondominated points of the original problem (P).*

$$\left(P_m^n\right)$$

$$\text{Max } z_m(\mathbf{x}) + \varepsilon \sum_{j \ne m} z_j(\mathbf{x})$$

subject to

$$z_j(\mathbf{x}) \ge \left(z_{tj} + 1\right) y_{tj} - M\left(1 - y_{tj}\right) \quad \forall j \ne m, \quad \forall t \tag{1}$$

$$\sum_{j \ne m} y_{tj} = 1 \quad \forall t \tag{2}$$

$$y_{tj} \in \{0, 1\} \quad t = 1, \ldots, n \quad j = 1, \ldots, p \quad j \ne m$$

$$\mathbf{x} \in \mathbf{X}$$

Proof Let $n = 1$ where only the nondominated point, $\mathbf{z}^1 = (z_{11}, z_{12}, \ldots, z_{1p})$ is available. Since $\sum_{j \ne m} y_{1j} = 1$, exactly one of the $p-1$ constraints "$z_j(\mathbf{x}) \ge z_{1j} + 1$" will be active and the others will be redundant for sufficiently large M. Therefore, at least one criterion value of the new nondominated point will be strictly greater than the corresponding value in $\mathbf{z}^1$, guaranteeing a different nondominated point. Since our aim is to maximize the mth criterion and guarantee to obtain a different nondominated point, we will obtain the nondominated point, $\mathbf{z}^2$, having the maximum mth criterion value among all feasible solutions. Since the feasible space of (P_m^1) is a subset of that of (P_m^0), the optimal objective function value of (P_m^1) is less than or equal to that of (P_m^0). Since ε is sufficiently small, it follows that $z_{2m} \le z_{1m}$. In case of infeasibility, we conclude that there is only one nondominated point. Similarly, for $n > 1$, the type (1) and type (2) constraints guarantee that the new nondominated point will be different from all the nondominated points in set S_n. The model will find a nondominated point, $\mathbf{z}^{n+1}$, having maximum mth criterion value among all feasible solutions. Since the feasible space of (P_m^n) is a subset of that of (P_m^{n-1}), we have $z_{(n+1)m} \leq z_{nm}$, as was the case for $n = 1$. If the problem is infeasible, then $S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$ contains the entire set of nondominated points. □

The above results show that all nondominated points to problem (P) can be generated by solving P_m^n iteratively.

Algorithm 1

Step 0. Select a criterion, m, to be maximized throughout the algorithm. Initialize $n = 0$. If $\mathbf{X}$ is empty, then there are no nondominated points. Stop.

Step 1. Solve model (P_m^n). If (P_m^n) is infeasible, go to Step 2. Otherwise, denote the nondominated point corresponding to the optimal solution as $\mathbf{z}^{n+1}$. Set $n \leftarrow n + 1$ and repeat Step 1.

Step 2. Stop. $S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$ contains the entire set of n nondominated points for problem (P).

Algorithm 1 improves the algorithm of Sylva and Crema (2004) by decreasing the additional number of binary variables from np to $n(p-1)$ and additional constraints from $n(p+1)$ to np in iteration n introduced to the model for finding the new nondominated point.

2.4 Development of Algorithm 2

Although Algorithm 1 reduces the additional constraints and variables, the models still grow and cause computational difficulties when the number of nondominated points is large. Here, we develop a new algorithm to further improve Algorithm 1. We observe that (P_m^n) includes np additional constraints and $n(p-1)$ binary variables. For any feasible solution, at most one constraint is sufficient to characterize the region that is nondominated relative to the available points for each of the $p-1$ criteria. Based on this observation, Algorithm 2 identifies the constraints that are necessary for the optimal solution and solves a number of models with $p-1$ or fewer additional lower bound constraints to find the solution to (P_m^n). Without loss of generality, let us set $m = p$ in order to simplify the notation. We denote $\mathbf{z}^{(P^{\mathbf{b}})} = (z_1^{(P^{\mathbf{b}})}, z_2^{(P^{\mathbf{b}})}, \ldots, z_p^{(P^{\mathbf{b}})})$ as the optimal nondominated point corresponding to the following problem:

$$
\begin{aligned}
&(P^{\mathbf{b}}) \\
&\text{Max } z_p(\mathbf{x}) + \varepsilon \sum_{j=1}^{p-1} z_j(\mathbf{x}) \\
&\text{subject to} \\
&z_j(\mathbf{x}) \geq b_j \quad j = 1, 2, \ldots, p-1 \\
&\mathbf{x} \in \mathbf{X}
\end{aligned}
$$

where $\mathbf{b} = (b_1, b_2, \ldots, b_{p-1})$ defines a lower bound for each of the first $p-1$ criteria. If $(P^{\mathbf{b}})$ is infeasible, then we assign $\mathbf{z}^{(P^{\mathbf{b}})} = (-M, -M, \ldots, -M)$.

We first prove that model (P_p^n) can be decomposed into submodels for $p = 3$ in Proposition 2. We then generalize it for general p in Proposition 3.

We partition (P_3^n) into $n+1$ submodels, $(P^{\mathbf{b}^{k,n}})$ $k = 0, 1, 2, \ldots, n$, where we set different bounds for the first and second criteria by using available nondominated points in

$S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$. $\mathbf{b}^{k,n} = (b_1^{k,n}, b_2^{k,n})$ denotes the corresponding bound vector for each submodel, $(P^{\mathbf{b}^{k,n}})$ $k = 0, 1, 2, \ldots, n$.

$k > 0$ implies that we use the kth nondominated point, $\mathbf{z}^k$, to define a lower bound for the first criterion, that is $b_1^{k,n} = z_{k1} + 1$. Introducing this bound for the first criterion excludes the region dominated by the available nondominated points that have first criterion values less than $b_1^{k,n}$. Once $b_1^{k,n}$ is introduced, it is sufficient to set the bound for the second criterion considering the nondominated points that have first criterion values at least as big as $b_1^{k,n}$. That is, if we define $S_n^k = \{\mathbf{z}^t : z_{t1} \geq b_1^{k,n} \quad \mathbf{z}^t \in S_n\}$, then we set $b_2^{k,n} = \max_{\mathbf{z}^t \in S_n^k}\{z_{t2}\} + 1$. If $S_n^k = \emptyset$, then there is no need to define a bound for the second criterion.

If $k = 0$, we do not have a lower bound for the first criterion and S_n^0 includes all the available nondominated points, that is $S_n^0 = S_n$.

Proposition 2 *Let* $p = 3$, $\mathbf{b}^{k,n} = (b_1^{k,n}, b_2^{k,n})$ *and* $S_n^k = \{\mathbf{z}^t : z_{t1} \geq b_1^{k,n} \quad \mathbf{z}^t \in S_n\}$ $k = 0, 1, 2, \ldots, n$, *and* k^* *be such that* $z_3^{(P^{\mathbf{b}^{k^*,n}})} = \max_{k=0,1,\ldots,n} \{z_3^{(P^{\mathbf{b}^{k,n}})}\}$ *where*

$$b_1^{k,n} = \begin{Bmatrix} -M & \text{if} \quad k = 0 \\ z_{k1} + 1 & \text{if} \quad k > 0 \end{Bmatrix} \qquad b_2^{k,n} = \begin{Bmatrix} -M & \text{if} \quad S_n^k = \emptyset \\ \max\limits_{\mathbf{z}^t \in S_n^k} \{z_{t2}\} + 1 & \text{otherwise} \end{Bmatrix}$$

If $z_3^{(P^{\mathbf{b}^{k^*,n}})} = -M$, *then* $S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$ *contains all nondominated points of the original problem* (P). *Otherwise,* $\mathbf{z}^{n+1} = \mathbf{z}^{(P^{\mathbf{b}^{k^*,n}})}$.

Proof Recall from Proposition 1 that $\mathbf{z}^{n+1} = (z_{(n+1)1}, z_{(n+1)2}, \ldots, z_{(n+1)p})$ can be obtained by solving problem (P_m^n). Let us rewrite model (P_m^n) for the three criteria case and $m = 3$:

$$\begin{aligned}
&\left(P_3^n\right) \\
&\text{Max } z_3(\mathbf{x}) + \varepsilon z_1(\mathbf{x}) + \varepsilon z_2(\mathbf{x}) \\
&\text{subject to} \\
&z_1(\mathbf{x}) \geq (z_{t1} + 1)\, y_t - M(1 - y_t) \qquad \forall t \\
&z_2(\mathbf{x}) \geq (z_{t2} + 1)(1 - y_t) - M(y_t) \quad \forall t \\
&y_t \in \{0, 1\} \quad t = 1, \ldots, n \\
&\mathbf{x} \in \mathbf{X}
\end{aligned}$$

where $y_t = 1$ implies $z_1(\mathbf{x}) \geq z_{t1} + 1$ while $y_t = 0$ implies $z_2(\mathbf{x}) \geq z_{t2} + 1$.

Depending on the value of k, one of the following two cases holds for the optimal nondominated point of (P_3^n):

Case 1 ($k = 0$). In this case, $y_t = 0$ for all $t = 1, 2, \ldots, n$. Hence, we do not have any additional lower bound for the first criterion value and so we can set $b_1^{k,n} = -M$. It also implies that $z_2(\mathbf{x}) \geq z_{t2}+1$ for all $t = 1, 2, \ldots, n$, hence we can define $b_2^{k,n} = \max_{t=1,2,\ldots,n}\{z_{t2}\}+1$. This corresponds to the case $k = 0$ where $S_n^0 = S_n$ since $z_{t1} \geq b_1^{k,n}$ for all $t = 1, 2, \ldots, n$.

Case 2 ($0 < k \leq n$). In this case, $y_k = 1$ for some k, $1 \leq k \leq n$ and $y_t = 0$ for all t satisfying $z_{t1} \geq z_{k1} + 1$. This implies $z_1(\mathbf{x}) \geq z_{k1} + 1$ and we can set $b_1^k = z_{k1} + 1$. For all t satisfying $z_{t1} \leq z_{k1}$, we obtain $z_{t1} + 1 \leq z_{k1} + 1 \leq z_1(\mathbf{x})$ and we can also assign $y_t = 1$.

Since $y_t = 0$ only for all t satisfying $z_{t1} \geq z_{k1} + 1$, $z_2(\mathbf{x}) \geq z_{t2} + 1$ for all t satisfying $z_{t1} \geq b_1^{k,n}$ (i.e., $\mathbf{z}^t \in S_n^k$). If $S_n^k \neq \emptyset$, the imposed lower bound for z_2 will be $b_2^{k,n} = \max_{\mathbf{z}^t \in S_n^k}\{z_{t2}\} + 1$. If $S_n^k = \emptyset$, then $b_2^{k,n} = -M$.

We find $\mathbf{z}^{(P^{\mathbf{b}^{k,n}})}$ for $k = 0, 1, \ldots, n$ by solving the corresponding models where all possible cases for the new solution are considered one by one. Since the aim is to maximize the third criterion, the submodel k^* that has the optimal solution with the largest third criterion value, $z_3^{(P^{\mathbf{b}^{k^*,n}})} = \max_{k=0,1,\ldots,n} \{z_3^{(P^{\mathbf{b}^{k,n}})}\}$, will give the next nondominated point. $z_3^{(P^{\mathbf{b}^{k^*,n}})} = -M$ corresponds to the case for which all submodels are infeasible and thus implies that (P_3^n) is infeasible and $S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$ contains the entire set of nondominated points. Otherwise, the corresponding submodel gives the next nondominated point, $\mathbf{z}^{n+1} = \mathbf{z}^{(P^{\mathbf{b}^{k^*,n}})}$. □

As in the case for $p = 3$, it is also possible to decompose (P_p^n) into submodels, $(P^{\mathbf{b}^{\mathbf{k},n}})$ $\mathbf{k} = (k_1, k_2, \ldots, k_{p-2})$, for any p. We use the available nondominated points to define the bounds for each criterion $j = 1, \ldots, p-1$. We denote the index of the nondominated point that we use to set a lower bound for the jth criterion as k_j, and $k_j = 0$ represents the special case of setting no lower bound for criterion j.

We first set a lower bound $b_1^{\mathbf{k},n} = z_{k_1 1} + 1$ to the first criterion when $0 < k_1 \leq n$ and $b_1^{\mathbf{k},n} = -M$ for $k_1 = 0$. Similar to the three-criteria case, we now consider only the nondominated points that have the first criterion values greater than or equal to $b_1^{\mathbf{k},n}$ to set the bound for the other criteria since the region dominated by the remaining solutions are already excluded with $b_1^{\mathbf{k},n}$. That is, the solution, $\mathbf{z}^{k_2}$, that will be used to set a lower bound for the second criterion will have the first criterion value $z_{k_2 1} \geq z_{k_1 1} + 1$ when $k_1, k_2 \neq 0$.

We continue defining bounds for each criterion $j = 1, 2, \ldots, p-2$ depending on the value of k_j by only considering the nondominated points, the dominated regions of which have not been excluded yet. That is, we have $z_{k_i i} + 1 \leq z_{k_j i}$ for all $k_i, k_j \neq 0 \quad i < j$.

In order to set a lower bound for criterion $p-1$, we define a subset $S_n^{\mathbf{k}} = \{\mathbf{z}^t : \mathbf{z}_{tj} \geq b_j^{\mathbf{k},n} \quad j = 1, \ldots, p-2, \quad \mathbf{z}^t \in S_n\}$ that contains the available nondominated points such that the regions that are dominated by them have not been excluded yet. That is, we should impose $z_{p-1}(\mathbf{x}) \geq z_{t(p-1)} + 1$ for each $\mathbf{z}^t \in S_n^{\mathbf{k}}$. Then, we have $b_{p-1}^{\mathbf{k},n} = \max_{\mathbf{z}^t \in S_n^{\mathbf{k}}} \{z_{t(p-1)}\} + 1$.

We next give the proof in Proposition 3.

Proposition 3 (Generalization of Proposition 2) *Let* $\mathbf{K}$ *denote the set of all possible combinations of* $\mathbf{k} = (k_1, k_2, \ldots, k_{p-2})$ *satisfying* $k_i = 0, 1, \ldots, n$ *and* $z_{k_i i} + 1 \leq z_{k_j i}$ *for all* $k_i, k_j \neq 0 \; i < j$. *Given* $\mathbf{b}^{\mathbf{k},n} = (b_1^{\mathbf{k},n}, b_2^{\mathbf{k},n}, \ldots, b_{p-1}^{\mathbf{k},n})$ *and* $S_n^{\mathbf{k}} = \{\mathbf{z}^t : \mathbf{z}_{tj} \geq b_j^{\mathbf{k},n} \quad j = 1, \ldots, p-2, \quad \mathbf{z}^t \in S_n\}$, *let* $\mathbf{k}^*$ *be such that* $z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = \max_{\mathbf{k} \in \mathbf{K}} \{z_p^{(P^{\mathbf{b}^{\mathbf{k},n}})}\}$ *where*

$$b_j^{\mathbf{k},n} = \begin{cases} -M & \text{if } \; k_j = 0 \\ z_{k_j j} + 1 & \text{otherwise} \end{cases} \qquad j = 1, \ldots, p-2$$

$$b_{p-1}^{\mathbf{k},n} = \begin{cases} -M & \text{if } \; S_n^{\mathbf{k}} = \emptyset \\ \max\limits_{\mathbf{z}^t \in S_n^{\mathbf{k}}} \left\{z_{t(p-1)}\right\} + 1 & \text{otherwise} \end{cases}$$

If $z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = -M$, *then* $S_n = \{\mathbf{z}^t : 1 \leq t \leq n\}$ *contains all nondominated points of the original problem* (P). *Otherwise,* $\mathbf{z}^{n+1} = \mathbf{z}^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})}$.

Proof Recall from Proposition 1 that $\mathbf{z}^{n+1} = (z_{(n+1)1}, z_{(n+1)2}, \ldots, z_{(n+1)p})$ can be obtained by solving problem (P_p^n).

$$\begin{aligned}
&\left(P_p^n\right)\\
&\text{Max } z_p(\mathbf{x}) + \varepsilon \sum_{j=1}^{p-1} z_j(\mathbf{x})\\
&\text{subject to}\\
&z_j(\mathbf{x}) \geq \left(z_{tj}+1\right) y_{tj} - M\left(1-y_{tj}\right) \quad \forall j \quad \forall t\\
&\sum_{j=1}^{p-1} y_{tj} = 1 \quad \forall t\\
&y_{tj} \in \{0,1\} \quad t=1,\ldots,n \quad j=1,\ldots,p-1\\
&\mathbf{x} \in \mathbf{X}
\end{aligned}$$

For the optimal nondominated point to the problem, we can write the following constraints:

If $y_{t1} = 0$ for all $t = 1, \ldots, n$, then there is no lower bound imposed for the first criterion, $b_1^{\mathbf{k},n} = -M$(corresponds to the case $k_1 = 0$). Otherwise, consider the case $y_{k_1 1} = 1$ for some k_1, $1 \leq k_1 \leq n$ and $y_{t1} = 0$ for all t satisfying $z_{t1} \geq z_{k_1 1} + 1$. Then, it implies $z_{k_1 1} + 1 \leq z_1(\mathbf{x})$, so we can define $b_1^{\mathbf{k},n} = z_{k_1 1} + 1$. For all t satisfying $z_{t1} \leq z_{k_1 1}$, we obtain $z_{t1} + 1 \leq z_{k_1 1} + 1 \leq z_1(\mathbf{x})$ and we can assign $y_{t1} = 1$.

For the second criterion, we know $y_{t2} = 0$ for all t satisfying $z_{t1} < b_1^{\mathbf{k},n}$($z_{t1} \leq z_{k_1 1}$if $k_1 \neq 0$) since $y_{t1} = 1$. If $y_{t2} = 0$ for all $t = 1, \ldots, n$, then there is no lower bound imposed, $b_2^{\mathbf{k},n} = -M$(corresponds to the case $k_2 = 0$). If $y_{k_2 2} = 1$ for some k_2 satisfying $z_{k_2 1} \geq b_1^{\mathbf{k},n}$($z_{k_2 1} \geq z_{k_1 1} + 1$if $k_1, k_2 \neq 0$) and $y_{t2} = 0$ for all t satisfying $z_{t2} \geq z_{k_2 2} + 1$, then $b_2^{\mathbf{k},n} = z_{k_2 2} + 1$.

In the same way, for each $j = 1, 2, \ldots, p-2$, the previous lower bound constraints imply that $y_{tj} = 0$ for all t satisfying $z_{ti} < b_i^{\mathbf{k},n}$ ($z_{ti} \leq z_{k_i i}$ if $k_i \neq 0$) for at least one criterion $i < j$ since $y_{ti} = 1$. If $y_{tj} = 0$ for all $t = 1, \ldots, n$ ($k_j = 0$), we will have $b_j^{\mathbf{k},n} = -M$. If $y_{k_j j} = 1$ for some k_j satisfying $z_{k_j i} \geq b_i^{\mathbf{k},n}$ ($z_{k_j i} \geq z_{k_i i} + 1$ if $k_i, k_j \neq 0$) for all $i < j$ and $y_{tj} = 0$ for all t satisfying $z_{tj} \geq z_{k_j j} + 1$, then:

$$b_j^{\mathbf{k},n} = \begin{cases} -M & \text{if } \ k_j = 0 \\ z_{k_j j} + 1 & \text{otherwise} \end{cases} \quad j = 1, \ldots, p-2.$$

For all $t = 1, 2, \ldots, n$, if $z_{tj} < b_j^{\mathbf{k},n}$ for at least one of the criteria $j = 1, 2, \ldots, p-2$ (implying $S_n^{\mathbf{k}} = \emptyset$), then it implies $y_{tj} = 1$ for one such criteria $j = 1, 2, \ldots, p-2$, hence $y_{t(p-1)} = 0$. That is, there is no lower bound imposed for criterion $p-1$, $b_{p-1}^{\mathbf{k},n} = -M$. Otherwise, for each t satisfying $z_{tj} \geq b_j^{\mathbf{k},n}$ for all $j = 1, 2, \ldots, p-2$ (i.e., $\mathbf{z}^t \in S_n^{\mathbf{k}}$), we can write $y_{tj} = 0$ for all $j = 1, 2, \ldots, p-2$ and so $y_{t(p-1)} = 1$. Then, we can write $z_{p-1}(\mathbf{x}) \geq z_{t(p-1)} + 1$ for each $\mathbf{z}^t \in S_n^{\mathbf{k}}$ and define:

$$b_{p-1}^{\mathbf{k},n} = \begin{cases} -M & \text{if } S_n^{\mathbf{k}} = \emptyset \\ \max\limits_{\mathbf{z}^t \in S_n^{\mathbf{k}}} \left\{z_{t(p-1)}\right\} + 1 & \text{otherwise} \end{cases}$$

We find $\mathbf{z}^{(P^{\mathbf{b}^{\mathbf{k},n}})}$ for all $\mathbf{k} \in \mathbf{K}$ by solving the corresponding models where all possible cases for the new solution are considered one by one. Since the aim is to maximize pth criterion value, the submodel $\mathbf{k}^*$ that has the optimal solution with the largest pth criterion value,

$z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = \max_{\mathbf{k}\in\mathbf{K}} \{z_p^{(P^{\mathbf{b}^{\mathbf{k},n}})}\}$, will give the next nondominated point. $z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = -M$ represents the special case for which all submodels are infeasible implying (P_p^n) is infeasible and hence $S_n = \{\mathbf{z}^t : 1 \le t \le n\}$ contains the entire set of nondominated points. Otherwise, the corresponding submodel gives the next nondominated point, $\mathbf{z}^{n+1} = \mathbf{z}^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})}$. □

We next present a basic algorithm that implements the above findings to generate all nondominated points setting $m = p$. We will later introduce improvements over the basic algorithm.

Step 0. Initialize $n = 0$. If (P_p^0) is infeasible, then $\mathbf{X}$ is empty and hence there are no nondominated points. Stop.

Step 1. Find $\mathbf{z}^{(P^{\mathbf{b}^{\mathbf{k},n}})}$ for all $\mathbf{k} \in \mathbf{K}$. Determine $\mathbf{k}^*$ for which $z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = \max_{\mathbf{k}\in\mathbf{K}} \{z_p^{(P^{\mathbf{b}^{\mathbf{k},n}})}\}$ is satisfied. If $z_p^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})} = -M$ (all models are infeasible), go to Step 2.

Otherwise, the new nondominated point is $\mathbf{z}^{n+1} = \mathbf{z}^{(P^{\mathbf{b}^{\mathbf{k}^*,n}})}$. Set $n \leftarrow n+1$ and repeat Step 1.

Step 2. Stop. $S_n = \{\mathbf{z}^t : 1 \le t \le n\}$ contains the entire set of n nondominated points for problem (P).

We demonstrate the algorithm on the following three-objective knapsack problem with 10 items:

$$\begin{aligned}&\text{“Max”}\,\{\mathbf{Px}\}\\ &\text{subject to}\\ &\mathbf{Wx} \le \mathbf{q}\\ &\mathbf{x} \in \{0,1\}^{10}\end{aligned}$$

where

$$\mathbf{P} = \begin{bmatrix} 54 & 64 & 46 & 37 & 31 & 62 & 52 & 33 & 87 & 35\\ 52 & 65 & 58 & 63 & 46 & 66 & 72 & 95 & 42 & 29\\ 56 & 90 & 34 & 13 & 71 & 33 & 66 & 74 & 88 & 71 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 52 & 52 & 28 & 23 & 95 & 69 & 13 & 61 & 32 & 68\\ 88 & 98 & 49 & 28 & 43 & 98 & 53 & 52 & 84 & 66\\ 57 & 30 & 86 & 50 & 97 & 96 & 59 & 94 & 67 & 14 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} 246 \; 329 \; 325 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \; x_2 \; x_3 \; x_4 \; x_5 \; x_6 \; x_7 \; x_8 \; x_9 \; x_{10} \end{bmatrix}^{\mathrm{T}}$$

We first demonstrate how we decompose (P_3^n) into submodels. Let us consider the case $n = 4$. Table 1 shows that $S_4 = \{\mathbf{z}^1, \mathbf{z}^2, \mathbf{z}^3, \mathbf{z}^4\}$ where $\mathbf{z}^1 = (256, 294, 336)$, $\mathbf{z}^2 = (230, 319, 335)$, $\mathbf{z}^3 = (253, 296, 333)$, and $\mathbf{z}^4 = (273, 337, 331)$. Proposition 1 shows that (P_3^4) will give the next nondominated point, $\mathbf{z}^5$. Instead of solving (P_3^4), we partition it into submodels $(P^{\mathbf{b}^{k,4}})$ $k = 0, 1, \ldots, 4$ where:

Submodel $0(k = 0)$: We do not impose a lower bound for the first criterion, $b_1^{0,4} = -M$. Then, we should have $z_2(\mathbf{x}) \ge z_{t2} + 1$ for all $t = 1, 2, 3, 4$ that corresponds to the case $S_4^0 = S_4$. We set $b_2^{0,4} = \max_{t=1,2,3,4}\{z_{t2}\} + 1 = 338$. The submodel gives the solution $\mathbf{z}^{(P^{\mathbf{b}^{0,4}})} = (240, 347, 299)$.

Submodel $1(k = 1)$: We use $\mathbf{z}^1$ to set a lower bound for $z_1(\mathbf{x})$, $b_1^{1,4} = 257$. Then, we consider only set $S_4^1 = \{\mathbf{z}^4\}$ since the region that is dominated by $\mathbf{z}^4$ is not excluded by

Table 1 Demonstration of the Algorithm on a 10-item, 3-objective knapsack problem

n	k	$\mathbf{b}^{k,n} = (b_1^{k,n}, b_2^{k,n})$	$\mathbf{z}(P^{\mathbf{b}^{k,n}})$	Need to solve $(P^{\mathbf{b}^{n,k}})$**?**	Number of model solved	k^*	$\mathbf{z}^{n+1}$
0	0	$(-M, -M)$	(256, 294, 336)	√	1	0	(256, 294, 336)
1	0	$(-M, 295)$	(230, 319, 335)	√	2	0	(230, 319, 335)
	1	$(257, -M)$	(273, 337, 331)	√			
2	0	$(-M, 320)$	(273, 337, 331)	√	2	2	(253, 296, 333)
	2	(231, 295)	(253, 296, 333)	√			
	1	$(257, -M)$	(273, 337, 331)	Identical to $(P^{\mathbf{b}^{1,1}})$			
3	0	$(-M, 320)$	(273, 337, 331)	Identical to $(P^{\mathbf{b}^{0,2}})$	2	1	(273, 337, 331)
	2	(231, 297)	(273, 337, 331)	√			
	3	(254, 295)	(273, 337, 331)	√			
	1	$(257, -M)$	(273, 337, 331)	Identical to $(P^{\mathbf{b}^{1,1}})$			
4	0	$(-M, 338)$	(240, 347, 299)	√	3	4	(275, 271, 328)
	2	(231, 338)	(240, 347, 299)	Identical to $(P^{\mathbf{b}^{0,4}})$			
	3	(254, 338)	Infeasible	√			
	1	(257, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	4	$(274, -M)$	(275, 271, 328)	√			
5	0	$(-M, 338)$	(240, 347, 299)	Identical to $(P^{\mathbf{b}^{0,4}})$	2	2	(240, 347, 299)
	2	(231, 338)	(240, 347, 299)	Identical to $(P^{\mathbf{b}^{0,4}})$			
	3	(254, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	1	(257, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	4	(274, 272)	(286, 300, 291)	√			
	5	$(276,-M)$	(286, 300, 291)	√			
6	0	$(-M,348)$	(232, 353, 277)	√	2	5	(286, 300, 291)
	2	(231, 348)	(232, 353, 277)	Identical to $(P^{\mathbf{b}^{0,6}})$			
	6	(241, 338)	Infeasible	√			
	3	(254, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	1	(257, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	4	(274, 272)	(286, 300, 291)	Identical to $(P^{\mathbf{b}^{4,5}})$			
	5	$(276, -M)$	(286, 300, 291)	Identical to $(P^{\mathbf{b}^{5,5}})$			
7	0	$(-M, 348)$	(232, 353, 277)	Identical to $(P^{\mathbf{b}^{0,6}})$	2	2	(232, 353, 277)
	2	(231, 348)	(232, 353, 277)	Identical to $(P^{\mathbf{b}^{0,6}})$			
	6	(241, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{6,6}})$			
	3	(254, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	1	(257, 338)	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	4	(274, 301)	Infeasible	√			
	5	(276, 301)	Infeasible	Identical to $(P^{\mathbf{b}^{4,7}})$			
	7	$(287, -M)$	Infeasible	√			

Table 1 continued

n	k	$\mathbf{b}^{k,n} = (b_1^{k,n}, b_2^{k,n})$	$\mathbf{z}^{(P^{\mathbf{b}^{k,n}})}$	Need to solve $(P^{\mathbf{b}^{n,k}})$?	Number of model solved	$k*$	$\mathbf{z}^{n+1}$
8	0	$(-M, 354)$	Infeasible	$\surd$	2		Stop. All nondominated points are found
	2	$(231, 354)$	Infeasible	Identical to $(P^{\mathbf{b}^{0,8}})$			
	8	$(233, 348)$	Infeasible	$\surd$			
	6	$(241, 338)$	Infeasible	Identical to $(P^{\mathbf{b}^{6,6}})$			
	3	$(254, 338)$	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	1	$(257, 338)$	Infeasible	Identical to $(P^{\mathbf{b}^{3,4}})$			
	4	$(274, 301)$	Infeasible	Identical to $(P^{\mathbf{b}^{4,7}})$			
	5	$(276,301)$	Infeasible	Identical to $(P^{\mathbf{b}^{4,7}})$			
	7	$(287, -M)$	Infeasible	Identical to $(P^{\mathbf{b}^{7,7}})$			

introducing $b_1^{1,4}$. We set $b_2^{1,4} = 338$. The submodel is infeasible and thus we assign $\mathbf{z}^{(P^{\mathbf{b}^{1,4}})} = (-M, -M, -M)$.

Submodel $2(k = 2)$: By using $\mathbf{z}^2$, we define $b_1^{2,4} = 231$. By considering only the solutions that have first criterion values greater than or equal to $b_1^{2,4}$, that is $S_4^2 = \{\mathbf{z}^1, \mathbf{z}^3, \mathbf{z}^4\}$, we set $b_2^{2,4} = \max_{\mathbf{z}^t \in S_4^2}\{z_{t2}\} + 1 = 338$. The optimal solution of the submodel is $\mathbf{z}^{(P^{\mathbf{b}^{2,4}})} = (240, 347, 299)$.

Submodel $3(k = 3)$: We set $b_1^{3,4} = 254$ by using $\mathbf{z}^3$ and $b_2^{3,4} = \max_{\mathbf{z}^t \in S_4^3}\{z_{t2}\} + 1 = 338$ by using set $S_4^3 = \{\mathbf{z}^1, \mathbf{z}^4\}$. We assign $\mathbf{z}^{(P^{\mathbf{b}^{3,4}})} = (-M, -M, -M)$ since the submodel turns out to be infeasible.

Submodel $4(k = 4)$: We set $b_1^{4,4} = 274$ using $\mathbf{z}^4$. Since we have $z_{t1} < b_1^{4,4}$ for all $t = 1, 2, 3, 4$, (i.e., $S_4^4 = \emptyset$), we do not need to define a lower bound for the second criterion. We set $b_2^{4,4} = -M$. The optimal solution turns out to be $\mathbf{z}^{(P^{\mathbf{b}^{4,4}})} = (275, 271, 328)$.

Since we are searching for the nondominated point having the largest third criterion value and $z_3^{(P^{\mathbf{b}^{4,4}})} = \max_{k=0,1,\ldots,n} \{z_3^{(P^{\mathbf{b}^{k,4}})}\} = \max\{299, -M, 299, -M, 328\} = 328$, we set $k^* = 4$. Hence, $\mathbf{z}^5 = \mathbf{z}^{(P^{\mathbf{b}^{4,4}})} = (275, 271, 328)$ as shown in Table 1.

The number of models to be solved to find the $(n+1)th$ solution by using the existing n solutions is $\sum_{\mathbf{k}\in\mathbf{K}} 1$ in the worst case. If N is the total number of nondominated points of (P), the number of models to be solved to find all N of them will be $\sum_{n=0}^{N} \sum_{\mathbf{k}\in\mathbf{K}} 1$ in the worst case, which is $O(N^{p-1})$. For three criteria case, if there are n previously generated nondominated points, we may need to solve $\sum_{k=0}^{n} 1 = n + 1$ models, in the worst case, in order to find the next nondominated point, $\mathbf{z}^{n+1}$. Therefore, we may need to solve a total of $\sum_{n=0}^{N} (n + 1) = \frac{(N+1)(N+2)}{2}$ models to find all N points in the worst case, which is $O(N^2)$. The number of models to be solved can be improved by keeping some information in the memory. Many of the models yield the same solution since we have only N nondominated points whereas we solve much more than N models each of which gives a nondominated point. We can detect if the optimal solution will be identical to any of the previous solutions by keeping the lower bound vector, $\mathbf{b}$, and the corresponding solution $\mathbf{z}^{(P^{\mathbf{b}})}$ as shown by Proposition 4.

Proposition 4 *Given* $\mathbf{b}_1 = (b_1^1, \ldots, b_{p-1}^1)$ *and* $\mathbf{b}_2 = (b_1^2, \ldots, b_{p-1}^2)$*, if* $b_j^1 \leq b_j^2 \leq z_j^{(P^{\mathbf{b}_1})}$ *for all* $j = 1, 2, \ldots, p-1$*, then* $\mathbf{z}^{(P^{\mathbf{b}_2})} = \mathbf{z}^{(P^{\mathbf{b}_1})}$.

Proof Since $b_j^2 \leq z_j^{(P^{\mathbf{b}_1})}$ for all $j = 1, 2, \ldots, p-1$, then the nondominated point $\mathbf{z}^{(P^{\mathbf{b}_1})}$ is also feasible for the problem $(P^{\mathbf{b}_2})$. Assume that problem $(P^{\mathbf{b}_2})$ has an optimal solution $\mathbf{z}^{(P^{\mathbf{b}_2})} \neq \mathbf{z}^{(P^{\mathbf{b}_1})}$. Since $\mathbf{z}^{(P^{\mathbf{b}_1})}$ is not an optimal solution to problem $(P^{\mathbf{b}_2})$ and both problems try to maximize pth criterion value, then $z_p^{(P^{\mathbf{b}_1})} < z_p^{(P^{\mathbf{b}_2})}$. Furthermore, we can write $b_j^2 \leq z_j^{(P^{\mathbf{b}_2})}$ for all $j = 1, 2, \ldots, p-1$, in order to provide the feasibility. Since we also know $b_j^1 \leq b_j^2$ for all $j = 1, 2, \ldots, p-1$, we obtain $b_j^1 \leq z_j^{(P^{\mathbf{b}_2})}$ for all $j = 1, 2, \ldots, p-1$ which implies $\mathbf{z}^{(P^{\mathbf{b}_2})}$ is also a feasible solution for problem $(P^{\mathbf{b}_1})$. However, $z_p^{(P^{\mathbf{b}_1})} < z_p^{(P^{\mathbf{b}_2})}$ implies that $\mathbf{z}^{(P^{\mathbf{b}_2})}$ has a better objective function value. Then, $\mathbf{z}^{(P^{\mathbf{b}_1})}$ cannot be an optimal solution to problem $(P^{\mathbf{b}_1})$,which is a contradiction. Therefore, $\mathbf{z}^{(P^{\mathbf{b}_2})} = \mathbf{z}^{(P^{\mathbf{b}_1})}$. □

In a similar manner, we can store the lower bounds that create infeasibility and utilize this information. Corollary 1 formalizes detecting infeasibility in future iterations without solving a model.

Corollary 1 *Given* $\mathbf{b}_1 = (b_1^1, \ldots, b_{p-1}^1)$ *and* $\mathbf{b}_2 = (b_1^2, \ldots, b_{p-1}^2)$*, if* $(P^{\mathbf{b}_1})$ *is infeasible and* $b_j^1 \leq b_j^2$ *for all* $j = 1, 2, \ldots, p-1$*, then* $(P^{\mathbf{b}_2})$*is also infeasible.*

Proof Assume that problem $(P^{\mathbf{b}_2})$ is feasible with an optimal solution, $\mathbf{z}^{(P^{\mathbf{b}_2})}$. Then, $\mathbf{z}^{(P^{\mathbf{b}_2})}$will also be feasible to problem $(P^{\mathbf{b}_1})$ since $b_j^1 \leq b_j^2$ for all $j = 1, 2, \ldots, p-1$ which contradicts that problem $(P^{\mathbf{b}_1})$ is infeasible. □

Using all these results, the algorithm is modified to store and utilize the lower bounds and corresponding candidate nondominated points. We store the lower bounds even if the problem is infeasible in order to detect infeasibility in future iterations. By using the information kept in the archives, we first check whether the solution of $(P^{\mathbf{b}})$ is identical to any previous solution and then solve if necessary. We also modify the order of solving submodels based on the bounds derived from the existing nondominated points. Rather than discussing these in detail, we illustrate them on the example problem in Table 1.

Although we need to consider $n+1$ models at each iteration, we can avoid solving many of them. By solving submodels in the nondecreasing order of $b_1^{k,n}$ and keeping some information in the memory, we end up solving only 2 models in most iterations as seen in Table 1. While the number of models to be solved could be as high as $\frac{(N+1)(N+2)}{2} = \frac{(8+1)(8+2)}{2} = 45$, in the worst case, we solve only 18 $(= 2(N+1))$ due to the improvements we made. The number of models solved in all three-objective experiments we report in the next section also shows a similar result of solving approximately two models for finding each nondominated point.

3 Computational experiments

We tested the performance of the algorithms on MOKP, MOST and MOSP problems. We conducted experiments on the same randomly generated problems used by

Table 2 Comparison of Algorithms on MOKP for $p = 3$

Number of items	Problem	Number of nondominated points (N)	Solution time (CPU time in seconds)		
			Sylva and Crema	Algorithm 1	Algorithm 2
20	1	35	14.24	6.75	3.62
	2	43	38.47	15.18	5.41
	3	61	102.40	39.29	8.80
	4	67	121.82	31.34	8.31
	5	77	259.51	48.76	10.59
25	1	57	118.13	40.73	9.65
	2	76	314.61	54.80	10.70
	3	103	818.26	53.35	15.08
	4	108	2,043.33	192.63	25.15
	5	132	5,291.38	193.52	20.67
	6	157	5,285.43	276.52	33.27
	7	163	5,253.49	245.59	25.56
	8	168	12,406.04	551.48	38.16
	9	182	14,740.24	407.65	30.23
	10	470	Could not be solved in 15 h	1,619.32	44.75

Köksalan and Lokman (2009). We formulate the minimum spanning tree problem as a multi-commodity flow problem [see Lokman (2007) for details] in order to convert it into a mathematical program to be solved by our algorithm. However, we should note that any other integer program formulation can be used for this algorithm.

We code all algorithms on Microsoft Visual Studio 2010 and use the callable library of CPLEX 12.3 on an Intel (R) Core (TM) i5-2410M CPU @ 2.30 GHz computer with 4.00 GB RAM and Microsoft Windows 7 Professional.

We compare our two exact algorithms with the algorithm developed by Sylva and Crema (2004) on MOKP. As the number of nondominated points increases, the complexity of the algorithm proposed by Sylva and Crema increases considerably as seen in Table 2. Therefore, we only solved small-sized knapsack problems with three objectives ($p = 3$).

The three algorithms in Table 2 are all exact algorithms generating all nondominated points. Therefore, we employ corresponding solution times as the performance measure. Algorithm 1 outperforms the algorithm developed by Sylva and Crema as seen in the computational times. This is expected since we decrease the number of binary variables and constraints. The computational times depend on N because we keep adding new binary variables and constraints until all nondominated points are obtained and this increases the computational complexity at each iteration. Table 2 also indicates that there is a significant increase in the difference in the computational times even when N increases slightly.

The additional constraints and variables cause computational difficulty in Algorithm 1 for larger problems. Algorithm 2 requires a sorting and searching mechanism and performs much better than Algorithm 1. We solve more models with Algorithm 2 but each model has the same number of constraints and variables regardless of the solutions on hand. Furthermore, we do not require any additional binary variables in Algorithm 2. While the computational times of Algorithm 1 and Algorithm 2 for the knapsack problem with 20 items are not much

Table 3 Comparison of Algorithms 1 and 2 on MOKP for $p = 3$

Number of items	Problem	Number of nondominated points (N)	Solution time (CPU time in seconds)	
			Algorithm 1	Algorithm 2
50	1	280	4,823.95	121.82
	2	356	5,173.84	139.63
	3	519	12,082.91	186.40
	4	784	33,699.41	360.54
	5	912	35,557.58	383.64

different in Table 2, we observe that the performance of Algorithm 2 gets much better as the problem size increases as seen in Table 3.

We also compared the performance of Algorithm 2 with that of the algorithm of Özlen and Azizoğlu (2009) on the three-criteria knapsack problems we present in Table 4. Özlen and Azizoğlu (2009) do not report any computational results. Therefore, we implemented their algorithm ourselves. Their average solution time for the 25-item knapsack problem turned out to be 90.16 s as opposed to 29.88 s of Algorithm 2. On the knapsack problem with 50 items, the algorithm of Özlen and Azizoğlu (2009) had an average solution time of 1,128.70 s while Algorithm 2 required 238.41 s on average. Özlen and Azizoğlu (2009)'s algorithm could not solve four of the five instances of the 100-item knapsack problem within a 15-h time limit that we set. In the remaining problem, their algorithm required 15,440.22 s, as opposed to the 3,061.91 s of our algorithm to generate all 2,790 nondominated points.

We conducted further experiments with Algorithm 2 on randomly generated instances of MOKP, MOST and MOSP problems with three and four objectives. The summary of the results are presented in Table 4.

If we consider the problems where $p = 3$, we solve $N + 1$ increasing-sized models where we insert three new constraints and two binary variables at each step of Algorithm 1. On the other hand, we may need to solve $(N + 1)(N + 2)/2$ problems in the worst case of Algorithm 2 if we cannot predict the optimal solution of any problem without solving the model by using the information kept in our archive. On the other hand, we may end up solving $N + 1$ models in the best case where we always have the opportunity to determine the next nondominated point by using the solutions kept in the archive after we solve $N + 1$ models. Then, the number of models solved, MS, to find all N nondominated points will be in the interval $N + 1 \leq MS \leq (N + 1)(N + 2)/2$. Since all these MS problems are equal-sized in terms of the variables and constraints, we use the average number of models solved per nondominated point, MS/N, as a performance measure. Based on the data in Table 5, we observe that MS/N is in the interval [1.80, 2.36] with an average of 2.13 when $p = 3$. That is, we roughly solve 2 models for each nondominated point on average. This indicates the importance of the information obtained from the archives of Algorithm 2 since $MS \ll (N + 1)(N + 2)/2$ especially for large N values. The value of MS decreases up to 0.01 % of $(N + 1)(N + 2)/2$ as demonstrated in Table 5. Furthermore, the ratio, MS/N, is not very sensitive to the value of Nwhich implies that we solve approximately the same number of models for each nondominated point. We should also note that all models include only two additional constraints and no additional variables regardless of the value of N.

Table 4 Performance of Algorithm-2 on random problems

Problem	Number of nondominated points (N)		Number of models solved (MS)		Sol. time (CPU time in seconds) (ST)		Avg. sol. time (ST/N)		MS/N	
	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD	Avg.	SD
MOKP 25 items $p = 3$	211.8	150.2	450.2	273.4	29.9	13.0	0.16	0.05	2.21	0.16
MOKP 50 items $p = 3$	570.2	271.7	1,218.0	534.8	238.4	124.6	0.41	0.04	2.17	0.09
MOKP 100 items $p = 3$	6786.2	2,954.6	12,495.2	5,197.0	23,204.8	19,549.9	2.91	1.48	1.86	0.06
MOKP 25 items $p = 4$	425.2	152.4	3,709.6	1,723.2	357.1	195.7	0.80	0.16	8.46	1.09
MOST problem 10 nodes $p = 3$	625.4	104.8	1,311.0	192.6	241.9	50.7	0.39	0.03	2.11	0.11
MOSP problem 25 nodes $p = 3$	86.4	55.9	195.4	129.7	6.0	4.8	0.07	0.01	2.24	0.09
MOSP problem 50 nodes $p = 3$	266.2	38.9	602.0	90.5	49.5	12.1	0.19	0.03	2.26	0.02
MOSP problem 100 nodes $p = 3$	469.8	98.5	1,010.6	203.5	212.7	34.9	0.46	0.03	2.15	0.04
MOSP problem 150 nodes $p = 3$	731.6	187.4	1,525.6	373.8	545.1	135.9	0.75	0.08	2.09	0.03
MOSP problem 200 nodes $p = 3$	778.2	180.9	1612.2	350.8	1066.3	234.2	1.38	0.12	2.08	0.06
MOSP problem 25 nodes $p = 4$	211.2	83.9	1,668.6	920.8	54.2	29.2	0.25	0.04	7.65	1.40
MOSP problem 50 nodes $p = 4$	1349.0	451.3	13,058.2	5125.6	1,498.2	691.8	1.07	0.20	9.49	1.03

Averages and SD of 5 problems per cell
The input and output files of all instances are available at: http://www.ie.metu.edu.tr/~koksalan/DataFiles.htm

If we consider the instances with $p = 4$, then MS/N again does not seem to be sensitive to the value of N where the ratio is within the interval [5.67,10.14] with an average value of 8.53. The value of MS/N is larger compared to the case of $p = 3$ for all instances indicating that it increases with the number of objectives. We should also note that the number of models to be solved in the worst case is $\sum_{n=0}^{N} (n+1)(n+2)/2$ (i.e. $O(N^3)$) for $p = 4$, which is much larger than the corresponding worst case of $p = 3$, $(N+1)(N+2)/2$. We can write $N + 1 \leq MS \leq \sum_{n=0}^{N} (n+1)(n+2)/2$ since the number of models to be solved in the best case is equal to $N + 1$. As we discuss for $p = 3$, if we consider the random instances demonstrated in Table 6, we observe $MS \ll \sum_{n=0}^{N} (n+1)(n+2)/2$ especially for large N values. The value of MS decreases up to 0.001 % of $\sum_{n=0}^{N} (n+1)(n+2)/2$.

In order to have an idea about the performance of our algorithm on integer programs, we also solved the three-objective knapsack problems replacing the binary restrictions on the variables with integer restrictions. We observed that both the averages and the standard deviations of the number of nondominated points increased in this case. For the 25-item knapsack problem, the average number of nondominated points increased from 211.80 to 560.80 while the standard deviation increased from 150.25 to 486.05. Similarly, the average number of nondominated points and standard deviation turned out to be 2,107.20 and 2,092.54, respectively, for the 50-item multi-objective integer knapsack problem while these values were 570.20 and 271.69, respectively, for the corresponding problems with 0–1 decision

Table 5 Percentage of models solved when $p = 3$ for all problem types

Number of nondominated points (N)	$\frac{MS}{(N+1)(N+2)/2} \times 100$ (%)	Number of nondominated points (N)	$\frac{MS}{(N+1)(N+2)/2} \times 100$ (%)
32	2.702	549	0.171
56	1.653	554	0.172
76	1.130	594	0.155
81	1.087	599	0.161
84	0.995	617	0.152
163	0.543	655	0.142
168	0.509	665	0.143
179	0.485	693	0.136
182	0.481	704	0.148
206	0.427	721	0.132
249	0.357	733	0.128
280	0.317	784	0.121
283	0.312	799	0.123
295	0.300	843	0.117
298	0.291	912	0.106
356	0.246	1,022	0.094
375	0.243	1,056	0.092
392	0.231	2,790	0.037
434	0.216	5,652	0.019
470	0.219	6,500	0.016
486	0.186	8,288	0.013
519	0.177	10,701	0.010
534	0.173		

variables. In our experiments on 100-item multi-objective integer knapsack problem, the average number of nondominated points increased to 9,361.00 from 6,786.20 and the standard deviation increased to 14,756.72 from 2,954.61.

Our experiments on the three-objective integer knapsack problem also showed that we solve approximately the same number of models for each nondominated point on average. We observe that MS/N is in the interval [1.80, 2.32] with an average of 2.08 for the three-objective 0–1 knapsack problem while this ratio is in the interval [1.78, 2.43] with an average value of 2.18 for the three-objective integer knapsack problem. We also observed that the average solution time per nondominated point improved from 1.16 to 0.24 s, and the standard deviation of the solution time per nondominated point decreased to 0.22 from 1.51, for the multi-objective integer knapsack problem, considering all three objective knapsack problems.

Although we develop two exact algorithms, Algorithm 1 and Algorithm 2, to generate all nondominated points and Algorithm 2 provides substantial decrease in the computational times, determining all nondominated points may still not be very practical especially for large-sized MOCO problems. The total number of nondominated points could be prohibitively large. The solution times may be substantially reduced if problem specific algorithms that exploit the structures of the problems can be used.

Table 6 Percentage of models solved when $p = 4$ for all problem types

Number of nondominated points (N)	$\frac{MS}{\sum_{n=0}^{N}(n+1)(n+2)/2} \times 100$ (%)
110	0.345
169	0.147
211	0.095
212	0.074
228	0.097
337	0.048
396	0.034
401	0.028
489	0.021
629	0.015
794	0.007
1,228	0.004
1,297	0.004
1,378	0.003
2,048	0.001

4 Conclusions

We developed two exact algorithms to generate all nondominated points for MOCO problems. We compared the performance of our algorithm with the algorithms proposed by Sylva and Crema (2004) and Özlen and Azizoğlu (2009). Although we showed that our algorithm works much better on selected test problems including MOKP, MOST and MOSP problems, computational times increase considerably as the problem size and the number of conflicting objectives increase. This is natural since the number of nondominated points increases substantially with the problem size.

As a future work, it may be useful to identify and focus on preferred regions incorporating decision maker's preferences. Our exact algorithms may be modified to accomplish this task by concentrating at the identified regions of the nondominated frontier. Currently, we are working on such variations of our algorithm.

We may also modify the algorithms by using some smart start techniques for solving the integer programs since we solve a number of closely related models at each iteration. For example, the solutions of previous iterations can be introduced as the starting solution of the current iteration. Furthermore, it may be useful to employ problem-specific information to customize the algorithms to certain MOCO problems. This and other potential improvements await future research.

References

Corley, H.W.: Efficient spanning trees. J. Optim. Theory Appl. **45**, 481–485 (1985)
Ehrgott, M., Gandibleux, X.: An annotated bibliography of multiobjective combinatorial optimization. OR Spektrum **22**, 425–460 (2000)
Ehrgott, M., Gandibleux, X.: Approximative solution methods for multiobjective combinatorial optimization. Sociedad de Estadística E Investigación Operativa **12**(1), 1–89 (2004)

Gomes da Silva, C., Clímaco, J., Figueira, J.R.: Core problems in bi-criteria 0–1 knapsack problems. Comput. Oper. Res. **35**(7), 2292–2306 (2008)
Guerriero, F., Musmanno, R.: Label correcting methods to solve multicriteria shortest path problems. J. Optim. Theory Appl. **111**(3), 589–613 (2001)
Köksalan, M.M.: A heuristic approach to bicriteria scheduling. Nav. Res. Logist. **46**, 777–789 (1999)
Köksalan, M., Lokman, B.: Approximating the nondominated frontiers of multi-objective combinatorial optimization problems. Nav. Res. Logist. **56**, 191–198 (2009)
Laumanns, M., Thiele, L., Zitzler, E.: An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. Eur. J. Oper. Res. **169**, 932–942 (2006)
Lokman, B.: Approaches for Multi-Objective Combinatorial Optimization Problems. Masters Thesis, Industrial Engineering Department, Middle East Technical University, Ankara (2007)
Martins, E.Q.V.: On a multicriteria shortest path problem. Eur. J. Oper. Res. **16**, 236–245 (1984)
Mavrotas, G., Figueira, J.R., Antoniadis, A.: Using the idea of expanded core for the exact solution of bi-Objective multi-dimensional knapsack problems. J. Glob. Optim. **49**(4), 589–606 (2011)
Mavrotas, G., Diakoulaki, D.: Multi-criteria branch and bound: A vector maximization algorithm for mixed 0-1 multiple objective linear programming. Appl. Math. Comput. **171**, 53–71 (2005)
Özlen, M., Azizoğlu, M.: Multi-objective integer programming: a general approach for generating all nondominated solutions. Eur. J. Oper. Res. **199**, 25–35 (2009)
Phelps, S., Köksalan, M.: An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. Manag. Sci. **49**, 1726–1738 (2003)
Prim, R.C.: Shortest connection networks and some generations. Bell Syst. Tech. J. **36**, 1389–1401 (1957)
Przybylski, A., Gandibleux, X., Ehrgott, M.: A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. Discret. Optim. **7**(3), 149–165 (2010)
Ramos, R.M., Alonso, S., Sicilia, J., Gonzáles, C.: The problem of the optimal biobjective spanning tree. Eur. J. Oper. Res. **111**, 617–628 (1998)
Steiner, S., Radzik, T.: Computing all efficient solutions of the biobjective minimum spanning tree problem. Comput. Oper. Res. **35**(1), 198–211 (2008)
Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Application. Wiley, New York (1986)
Sylva, J., Crema, A.: A method for finding the set of nondominated vectors for multiple objective integer linear programs. Eur. J. Oper. Res. **158**, 46–55 (2004)
Sylva, J., Crema, A.: A method for finding well-dispersed subsets of nondominated vectors for multiple objective mixed integer linear programs. Eur. J. Oper. Res. **180**, 1011–1027 (2007)
Tung, C.T., Chew, K.L.: A multicriteria pareto-optimal path algorithm. Eur. J. Oper. Res. **62**, 203–209 (1992)
Ulungu, E.L., Teghem, J., Fortemps, P.H., Tuyttens, D.: MOSA method: a tool for solving multiobjective combinatorial optimization problems. J. Multi-Criteria Decis. Anal. **8**, 221–236 (1999)
Visée, M., Teghem, J., Pirlot, M., Ulungu, E.L.: Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. J. Glob. Optim. **12**, 139–155 (1998)