European Journal of Operational Research 268 (2018) 281–299

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Decision Support

The search-and-remove algorithm for biobjective mixed-integer linear programming problems

Banu Soylu

Department of Industrial Engineering, Erciyes University, 38039 Kayseri, Turkey

ARTICLE INFO

Article history:
Received 13 March 2017
Accepted 11 January 2018
Available online 31 January 2018

Keywords:
Multiple objective programming
Biobjective mixed-integer linear programming
Integer programming
Bound sets

ABSTRACT

In this study, an exact algorithm, called the search-and-remove (SR) algorithm, is proposed to compute the Pareto frontier of biobjective mixed-integer linear programming problems. At each stage of the algorithm, efficient slices (all integer variables are fixed in a slice) are searched with the dichotomic search algorithm and found slices are recorded and excluded from the decision space with the help of Tabu constraints. The algorithm is also enhanced with lower and upper bounds, which are updated at each stage of the algorithm. The SR algorithm continues until it is proved that all efficient slices of the biobjective mixed-integer linear programming (BOMILP) problem are found. The algorithm finally returns a set of potentially efficient slices including all efficient slices of the problem. Then, an upper envelope finding algorithm merges the Pareto frontiers of these slices to the Pareto frontier of the original problem. A computational analysis is performed on several benchmark problems and the performance of the algorithm is compared with state of the art methods from the literature.

1. Introduction

Mixed-integer linear programming problems (MILP) comprise a wide class of discrete optimization problems. In a MILP, if there are two conflicting objective functions, the problem turns into the BOMILP problem.

When all integer variables of a BOMILP problem are fixed, a biobjective linear programming problem (BOLP) called the *slice problem* (Belotti, Soylu, & Wiecek, 2013) is obtained. It is well-known in the literature that the Pareto frontier of a BOLP can easily be found with the parametric simplex algorithm (Ehrgott, 2005) or the dichotomic search algorithm (Aneja & Nair, 1979; Cohon, 1978). If any Pareto point of a slice problem contributes to the Pareto frontier of the BOMILP problem, then it is called an *efficient slice*. Since the number of efficient slices can be of exponential in the size of the input (i.e., intractable), there is no chance for a polynomial time algorithm for BOMILP. A primitive algorithm would be to solve all slice problems of the BOMILP problem, obviously this requires the complete enumeration of the set of integer variables, and then finding the upper envelope (for maximization problems) of all resulting points. The aim of the SR algorithm is precisely to improve this primitive approach in such a way as to avoid this complete enumeration.

Multiobjective MILP (MOMILP) problems are common in real life. Some examples are multiobjective hub-location (Köksalan & Soylu, 2010), multiobjective hydro-thermal self-scheduling (Ahmadi, Aghaei, Shayanfar, & Rabiee, 2012), biobjective heat and power production planning (Rong, Figueira, & Lahdelma, 2015), multiobjective energy planning (Mavrotas, Diakoulaki, & Papayannakis, 1999) etc. Many MOMILP problems exist in the literature and classical algorithms such as the ε-constraint algorithm (Haimes, Lasdon, & Wismer, 1971) have been mostly used for finding efficient solutions. However, these algorithms are only able to provide a subset of efficient solutions of MOMILP problems.

The single objective MILP literature is vast in exact and in heuristic methods, however the MOMILP algorithms literature is relatively new. Decision space search algorithms such as branch-and-bound (BB) have been applied to MOMILP as well. One of the first attempts to develop a BB algorithm for MOMILP problems is by Mavrotas and Diakoulaki (1998, 2005). Vincent, Seipp, Ruzika, Przybylski, and Gandibleux (2013) improved the BB algorithm of Mavrotas and Diakoulaki in terms of generation of the Pareto frontier. They also presented better bounds and branching strategies. Stidsen, Andersen, and Dammann (2014) proposed a biobjective BB algorithm, which can handle a subclass of biobjective mixed-binary linear programming (BOMBLP) problems, where continuous variables are only to be part of one of two objective functions. Belotti et al. (2013) presented the first general purpose BB algorithm for BOMILP problems. They also proposed improved fathoming rules to eliminate more nodes for decreasing the computation

E-mail address: bsoylu@erciyes.edu.tr

https://doi.org/10.1016/j.ejor.2018.01.026

time (Belotti, Soylu, & Wiecek, 2016). Soylu and Yıldız (2015) have recently introduced a new tree based algorithm, called the local-branch-and bound algorithm, which uses the local branching concept of Fischetti and Lodi (2003).

Criterion space search algorithms have also been receiving growing attention in the literature. The first general purpose criterion space search algorithm for BOMILP problems was introduced by Boland, Charkhgard, and Savelsbergh (2015). They developed a triangle splitting algorithm, which maintains a diverse set of Pareto points throughout the algorithm. Recently, Soylu and Yıldız (2016) presented the ε, *Tabu*-constraint algorithm, which sequentially finds all Pareto line segments and points of BOMILP problems starting with one end point of the Pareto frontier.

Particularly for finding the set of extreme Pareto points of MOMILP problems, Özpeynirci and Köksalan (2010) presented an algorithm based on dichotomic search and weight space decomposition. Przybylski, Gandibleux, and Ehrgott (2010) proposed an algorithm for finding the set of extreme Pareto points of multiobjective integer linear programming (MOILP) problems.

This paper proposes an exact algorithm for BOMILP problems. The motivation is that the Pareto frontier of BOMILP problems can be computed by considering Pareto frontiers of (BOLP) slice problems and eliminating dominated points. Since BOMILP problems are in general not convex, finding all efficient slices is not easy. However, removing searched slices from the decision space makes it easier to find others. For this purpose, at each iteration efficient slices of the problem are searched with the dichotomic search algorithm. Before starting the next iteration, slices found are eliminated from the search space with the help of Tabu constraints added to the problem. These search and remove iterations repeat until it is proved that all efficient slices of the problem are found. For this purpose, upper and lower bound sets are used. Obviously, tighter bound sets lead to earlier termination of the algorithm by reducing the number of performed iterations. As a performance improvement strategy, partitioning the objective space into several subregions and parallel processing of each subregion are suggested.

In Section 2, basic definitions and models are provided. In Section 3, fundamental mechanisms of the algorithm are discussed. In Section 4, the search-and-remove iterative framework is proposed for BOMILP problems. In Section 5, computational results and performance comparisons are presented. Finally, the conclusion and further research directions are given in Section 6.

2. Basic definitions and models

Readers may refer to Isermann (1974), Yu and Zeleny (1975), Naccache (1978), Steuer (1985) and Ehrgott (2005) for a detailed coverage of multiobjective optimization.

The BOMILP problem can be formulated as follows:

$$\begin{aligned} P:\ & Max\ z_1(\tilde{\boldsymbol{x}},\boldsymbol{x}) = \tilde{\boldsymbol{c}}_1^T\tilde{\boldsymbol{x}} + \boldsymbol{c}_1^T\boldsymbol{x} \\ & Max\ z_2(\tilde{\boldsymbol{x}},\boldsymbol{x}) = \tilde{\boldsymbol{c}}_2^T\tilde{\boldsymbol{x}} + \boldsymbol{c}_2^T\boldsymbol{x} \\ & Subject\ to\ (\tilde{\boldsymbol{x}},\boldsymbol{x}) \in X \end{aligned}$$

where the set $X := \{ (\tilde{\boldsymbol{x}},\boldsymbol{x}) \in \mathbb{R}_+^p \times \mathbb{Z}_+^{n-p} :\ \tilde{A}\tilde{\boldsymbol{x}} + A\boldsymbol{x} \leqq \boldsymbol{b} \}$ is the set of feasible solutions defined in the decision space. The vectors $\tilde{\boldsymbol{c}}_1,\ \tilde{\boldsymbol{c}}_2 \in \mathbb{R}^p$ and $\boldsymbol{c}_1,\ \boldsymbol{c}_2 \in \mathbb{R}^{n-p}$ are cost vectors, the matrices $\tilde{A} \in \mathbb{R}^{m\times p}, A \in \mathbb{R}^{m\times\ (n-p)}$ are coefficient matrices of m constraints, and $\boldsymbol{b} \in \mathbb{R}^m$ is the right-hand side vector. There are p continuous and $n-p$ integer variables. It is assumed that each integer variable x_i is bounded, i.e. $0 \le l_i \le x_i \le u_i < +\infty$ for $i = 1, 2, \ldots\ldots, n-p$ where l_i and u_i are lower and upper bound values, respectively. It is also assumed that each continuous variable $\tilde{x}_i$ is bounded, i.e. $0 \le \tilde{x}_i \le u_i < +\infty$ for $i = 1, 2, \ldots\ldots, p$. Here, the biobjective integer programming (BOIP) problem, where $p = 0$, and the biobjective linear programming (BOLP) problem, where $p = n$, are considered as special classes of the BOMILP problem. The image of the set X is $Y := z(X) := \{\boldsymbol{y} \in \mathbb{R}^2 :\ \boldsymbol{y} = z(\tilde{\boldsymbol{x}},\boldsymbol{x})$ for some $(\tilde{\boldsymbol{x}},\boldsymbol{x}) \in X \}$ called the set of attainable vectors defined in the objective/criterion space $\mathbb{R}^2$. Here $\boldsymbol{y} = z(\tilde{\boldsymbol{x}},\boldsymbol{x})$ where $z = (z_1, z_2)$.

Definition 1. A feasible solution $(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*) \in X$ is said to be *efficient* for problem P if and only if there is no $(\tilde{\boldsymbol{x}},\boldsymbol{x}) \in X$ such that $z(\tilde{\boldsymbol{x}},\boldsymbol{x}) \ge z(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*)$. If $(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*) \in X$ is efficient, then $z(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*)$ is called a *Pareto point*. If $(\tilde{\boldsymbol{x}}^1,\boldsymbol{x}^1)$, $(\tilde{\boldsymbol{x}}^2,\boldsymbol{x}^2) \in X$ and $z(\tilde{\boldsymbol{x}}^1,\boldsymbol{x}^1) \ge z(\tilde{\boldsymbol{x}}^2,\boldsymbol{x}^2)$, then it is said that $(\tilde{\boldsymbol{x}}^1,\boldsymbol{x}^1)$ dominates $(\tilde{\boldsymbol{x}}^2,\boldsymbol{x}^2)$, and $z(\tilde{\boldsymbol{x}}^1,\boldsymbol{x}^1)$ dominates $z(\tilde{\boldsymbol{x}}^2,\boldsymbol{x}^2)$. The set of all efficient solutions $(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*) \in X$ is denoted by X_E. The set of all Pareto points $z(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*) \in Y$ for some $(\tilde{\boldsymbol{x}}^*,\boldsymbol{x}^*) \in X_E$ is denoted by Y_N, also referred to as the *Pareto frontier*.

Given two vectors $\boldsymbol{y}^1,\ \boldsymbol{y}^2 \in \mathbb{R}^2$, the following notation is used:

$\boldsymbol{y}^1 \geqq \boldsymbol{y}^2$ if $y_k^1 \geqq y_k^2$ for all $k = 1, 2$
$\boldsymbol{y}^1 \ge \boldsymbol{y}^2$ if $\boldsymbol{y}^1 \geqq \boldsymbol{y}^2$ and $\boldsymbol{y}^1 \ne \boldsymbol{y}^2$
$\boldsymbol{y}^1 > \boldsymbol{y}^2$ if $y_k^1 > y_k^2$ for all $k = 1, 2$.

The set $\mathbb{R}^2_{\geqq}$ is defined as $\mathbb{R}^2_{\geqq} := \{\boldsymbol{y} \in \mathbb{R}^2 : \boldsymbol{y} \geqq 0\}$, and sets $\mathbb{R}^2_>$, $\mathbb{R}^2_\ge$, $\mathbb{R}^2_<$, $\mathbb{R}^2_\le$, $\mathbb{R}^2_\leqq$ are defined similarly. Let S be an arbitrary set. Then, the set $S^{\geqq}$ is defined as $S^{\geqq} = S + \mathbb{R}^2_{\geqq} := \{s + \boldsymbol{y} : s \in S, \boldsymbol{y} \in \mathbb{R}^2_{\geqq}\}$, and sets $S^>$, $S^\ge$, $S^<$, $S^\le$, $S^\leqq$ are defined similarly.

Definition 2. A set $S \subset \mathbb{R}^2$ is called *connected* if there are no open sets O_1, O_2 such that $S \subset O_1 \cup O_2$, $S \cap O_1 \ne \emptyset$, $S \cap O_2 \ne \emptyset$, $S \cap O_1 \cap O_2 = \emptyset$.

Definition 3. A feasible solution $(\tilde{\boldsymbol{x}}',\boldsymbol{x}') \in X$ is said to be *weakly efficient* for problem P if and only if there is no $(\tilde{\boldsymbol{x}},\boldsymbol{x}) \in X$ such that $z(\tilde{\boldsymbol{x}},\boldsymbol{x}) > z(\tilde{\boldsymbol{x}}',\boldsymbol{x}')$. Then the point $\boldsymbol{y}' = z(\tilde{\boldsymbol{x}}',\boldsymbol{x}')$ is called a *weak Pareto point*.

Definition 4. Let $(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a) \in X_E$. If there is some $\lambda \in (0,1)$ such that $(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a) \in X_E$ is an optimal solution of $max_{(\tilde{\boldsymbol{x}},\boldsymbol{x})\in X}\ \{\lambda z_1(\tilde{\boldsymbol{x}},\boldsymbol{x}) + (1-\lambda) z_2(\tilde{\boldsymbol{x}},\boldsymbol{x})\}$, then $(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a) \in X_E$ is called a *supported efficient solution* and $\boldsymbol{y}^a = z(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a)$ is called a *supported Pareto point*. Otherwise, $(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a) \in X_E$ is called a *nonsupported efficient solution* and $\boldsymbol{y}^a = z(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a)$ is called a *nonsupported Pareto point*.

Definition 5. A Pareto point $\boldsymbol{y}^a = z(\tilde{\boldsymbol{x}}^a,\boldsymbol{x}^a) \in Y_N$ is called an *extreme supported Pareto (ExSP) point* if $\boldsymbol{y}^a$ is an extreme point of $Conv(Y)$. The set Y_{XN} denotes the set of *ExSP* points.

Definition 6. Let $Y_{XN} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots,\ \boldsymbol{y}^h\}$ such that $y_1^1 < y_1^2 < \ldots < y_1^h$. Points $\boldsymbol{y}^j$ and $\boldsymbol{y}^{j+1}$, $\forall j = 1, 2, \ldots, h-1$ are called *adjacent*.

Definition 7. Let $\boldsymbol{y}^a$, $\boldsymbol{y}^b \in Y$ be two attainable points such that $y_1^a < y_1^b$ and $y_2^a > y_2^b$. The point $\check{\boldsymbol{y}} = (y_1^a, y_2^b)$ is called the *local nadir point* with respect to $\boldsymbol{y}^a$ and $\boldsymbol{y}^b$. The point $\check{\check{\boldsymbol{y}}} = (\check{\check{y}}_1, \check{\check{y}}_2)$ given by $\check{\check{y}}_k = \min_{\boldsymbol{y}\in Y_N} y_k$ for $k = 1, 2$ is called the *nadir point*. The point $\boldsymbol{y}^I = (y_1^I, y_2^I)$ given by $y_k^I = max\{z_k(\tilde{\boldsymbol{x}},\boldsymbol{x}) : (\tilde{\boldsymbol{x}},\boldsymbol{x}) \in X\}$ for $k = 1, 2$ is called the *ideal point*.

Given a weight $\lambda \in (0,1)$ the single objective weighted-sum problem P_λ is defined as $max_{(\tilde{\boldsymbol{x}},\boldsymbol{x})\in X}\ \{\lambda z_1(\tilde{\boldsymbol{x}},\boldsymbol{x}) + (1-\lambda) z_2(\tilde{\boldsymbol{x}},\boldsymbol{x})\}$.

A *slice problem* of P is a BOLP obtained by fixing the integer variables of P and defined as follows (Belotti et al., 2013):

$$\begin{aligned} P\left(\bar{\boldsymbol{x}}^j\right):\ & Max\ z_1(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{c}}_1^T\tilde{\boldsymbol{x}} + \boldsymbol{c}_1^T\bar{\boldsymbol{x}}^j \\ & Max\ z_2(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{c}}_2^T\tilde{\boldsymbol{x}} + \boldsymbol{c}_2^T\bar{\boldsymbol{x}}^j \\ & Subject\ to\ \tilde{\boldsymbol{x}} \in X\left(\bar{\boldsymbol{x}}^j\right) \end{aligned}$$

where $X(\bar{\boldsymbol{x}}^j) := \{ \tilde{\boldsymbol{x}} \in \mathbb{R}_+^p : \tilde{A}\tilde{\boldsymbol{x}} \leqq \boldsymbol{b} - A\bar{\boldsymbol{x}}^j$ for a given $\bar{\boldsymbol{x}}^j \in \mathbb{Z}_+^{n-p}\}$ defines a slice of the set X. Here the $\bar{\boldsymbol{x}}^j$ vector refers to integer values defined priorly. The set $Y(\bar{\boldsymbol{x}}^j) := z(X(\bar{\boldsymbol{x}}^j)) :=$