European Journal of Operational Research 000 (2018) 1–17

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Finding a representative nondominated set for multi-objective mixed integer programs

Gökhan Ceyhan, Murat Köksalan, Banu Lokman*

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

ARTICLE INFO

Article history:
Received 4 October 2016
Accepted 6 June 2018
Available online xxx

Keywords:
Multiple objective programming
Mixed integer programming
Nondominated point
Representative subset

ABSTRACT

In this paper, we develop algorithms to find small representative sets of nondominated points that are well spread over the nondominated frontiers for multi-objective mixed integer programs. We evaluate the quality of representations of the sets by a Tchebycheff distance-based coverage gap measure. The first algorithm aims to substantially improve the computational efficiency of an existing algorithm that is designed to continue generating new points until the decision maker (DM) finds the generated set satisfactory. The algorithm improves the coverage gap value in each iteration by including the worst represented point into the set. The second algorithm, on the other hand, guarantees to achieve a desired coverage gap value imposed by the DM at the outset. In generating a new point, the algorithm constructs territories around the previously generated points that are inadmissible for the new point based on the desired coverage gap value. The third algorithm brings a holistic approach considering the solution space and the number of representative points that will be generated together. The algorithm first approximates the nondominated set by a hypersurface and uses it to plan the locations of the representative points. We conduct computational experiments on randomly generated instances of multi-objective knapsack, assignment, and mixed integer knapsack problems and show that the algorithms work well.

1. Introduction

Mixed integer programs (MIPs) are encountered in many different fields. Practical MIPs need to be evaluated by multiple objectives in general. Multi-objective mixed integer programs (MOMIPs) have important practical value. Solving MOMIPs bring computational challenges as finding each nondominated solution is hard and the nondominated frontier is at least partially continuous.

MOMIPs have been addressed by a number of researchers. Supported nondominated points can be obtained by optimizing a weighted sum of objectives with suitably chosen weights and they constitute a subset of all nondominated points. Özpeynirci and Köksalan (2010) and Przybylski, Gandibleux, and Ehrgott (2010) developed methods to generate all extreme supported nondominated points for MOMIPs. Since the nondominated set is not finite, generating all nondominated solutions is not possible and characterizing the nondominated set is not straightforward.

Multi-objective integer programs (MOIPs) constitute a special case of MOMIPs where all variables are integers. MOIPs are encountered in many decision problems and have been widely studied. The cardinality of the nondominated set is finite in MOIPs. Although it is possible to generate all nondominated points in MOIPs, there are some difficulties. Not only is generating each nondominated point hard, but also the number of nondominated points could become very large with increased problem size (see Ehrgott & Gandibleux, 2000).

Many existing approaches for MOIPs attempt to generate all nondominated points. Epsilon constraint, two-phase, and branch-and-bound methods are commonly utilized for this purpose (for example, see Delort & Spanjaard, 2010; Dhaenens, Lemesre, & Talbi, 2010; Lemesre, Dhaenens, & Talbi, 2007; Przybylski, Gandibleux, & Ehrgott, 2008). It is not very practical to generate all nondominated points for more than two objectives for even moderate-sized MOIPs.

A number of approaches attempt to generate all nondominated points by partitioning the solution space. Sylva and Crema (2004) partition the solution space by adding binary variables and constraints to exclude the already-generated points from the feasible set. Naturally, the problem becomes computationally more challenging with every new nondominated point. Özlen and Azizoğlu (2009) and Laumanns, Thiele, and Zitzler (2006) developed methods to generate all nondominated points utilizing the epsilon-constraint method. Recently, more efficient algorithms

* Corresponding author.
E-mail address: lbanu@metu.edu.tr (B. Lokman).

https://doi.org/10.1016/j.ejor.2018.06.012
0377-2217/

were developed by Lokman and Köksalan (2013), Kirlik and Sayın (2014), Ozlen, Burton, and MacRae (2014), Dächert and Klamroth (2015), Boland, Charkhgard, and Savelsbergh (2016), and Boland, Charkhgard, and Savelsbergh (2017) to generate all nondominated points by partitioning the solution space effectively. Of these, Dächert and Klamroth (2015), Boland et al. (2016), and Boland et al. (2017) are restricted to three-objective problems while the others can be used for more objectives as well.

Due to the difficulty of finding all nondominated points for realistic-sized MOIPs, many heuristics and metaheuristics have been developed. Ehrgott and Gandibleux (2004, 2008) review approximate and hybrid metaheuristics for multi-objective combinatorial optimization (MOCO) problems (MOIPs having special structures). Although these heuristic methods are computationally efficient, the generated points are not necessarily nondominated and they do not provide any performance guarantees on the quality level of the approximation set.

In this paper, our goal is to find a reasonable number of well-distributed nondominated points to represent the nondominated set for MOMIPs. Sayın (2000) proposed three measures: coverage, uniformity and cardinality, to evaluate the quality of a subset of nondominated points. There have been a number of studies for finding a representative set to approximate the nondominated frontier. Sayın (2003) and Karasakal and Köksalan (2009) generated discrete representations of the continuous nondominated frontiers for multi-objective linear programs (MOLPs). Sylva and Crema (2007) and Masin and Bukchin (2008) developed algorithms that are similar to each other and generate similar representative subsets of nondominated points for MOMIPs. Other algorithms that have been designed to generate all nondominated points for any number of objectives for MOIPs are not concerned with the quality of representing the nondominated set during the generation process. Therefore, the set of nondominated points generated until a certain iteration of the algorithm is unlikely to produce a good representation except by chance. It is not clear whether they can be modified to consider the quality of representation in the generation of every new nondominated point.

Generating a representative subset of nondominated points has two main benefits. Firstly, it is a computationally-efficient alternative to characterizing the nondominated set, which is impractical for MOMIPs. Additionally, a well-dispersed sample provides information about the layout of the nondominated set. An approximate knowledge of the distribution of solutions would provide the DM valuable information during the decision making process that might follow.

In this paper, we develop exact methods that find nondominated representative points either guaranteeing the quality of representation or searching for the best representation given the number of representative points. We develop three methods that serve different purposes to generate high quality representative nondominated sets for MOMIPs. The paper is organized as follows: in Section 2, we provide the necessary background. We develop our methods and provide computational results in Section 3 and conclude in Section 4.

2. Background

In this section, we provide the necessary background, define the performance measure we use, and develop the requisite theory.

2.1. Definitions

Consider the following problem:

(MOMIP):

$$\begin{aligned}
\text{“Max”} \quad & \mathbf{z(x)} = \{z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_m(\mathbf{x})\} \\
\text{s.t.} \quad & \mathbf{x} \in X \\
& x_v \in \mathbb{Z}, \forall v \in V \\
& x_u \in \mathbb{R}, \forall u \in U
\end{aligned}$$

where $z_i(\mathbf{x})$ is a continuous function of $\mathbf{x}$ denoting the ith objective function. We denote the feasible decision space as X and the image of X in the objective space as Z. $X = \{\mathbf{x} \in P : x_v \in \mathbb{Z} \quad \forall v \in V, \quad x_u \in \mathbb{R} \quad \forall u \in U\}$, where $P \subseteq \mathbb{R}^n$ is compact, V is the index set of integer decision variables, U is the index set of real-valued decision variables and $n = |V| + |U|$. The quotation marks are used since there does not exist a unique solution that is maximal in all objective functions. Without loss of generality, we assume we have m maximization-type objectives.

Definition 1. $\mathbf{x}^k \in X$ is an efficient solution if $\nexists$ $\mathbf{x}^j \in X$ such that $z_i(\mathbf{x}^j) \geq z_i(\mathbf{x}^k)$ $\forall i$ and $z_i(\mathbf{x}^j) > z_i(\mathbf{x}^k)$ for at least one i. If $\mathbf{x}^k$ is efficient, then $\mathbf{z}(\mathbf{x}^k)$ is said to be nondominated. On the other hand, if there exists such an $\mathbf{x}^j$, then $\mathbf{x}^k$ is said to be inefficient and $\mathbf{z}(\mathbf{x}^k)$ is said to be dominated.

Definition 2. If $\exists$ $\mathbf{x}^j \in X$ such that $z_i(\mathbf{x}^j) > z_i(\mathbf{x}^k)$ $\forall i$, then $\mathbf{x}^k$ is said to be *strictly inefficient* and $\mathbf{z}(\mathbf{x}^k)$ is *strictly dominated*. If there exists no such $\mathbf{x}^j$, then $\mathbf{x}^k$ is *weakly efficient* and $\mathbf{z}(\mathbf{x}^k)$ is *weakly nondominated*.

Note that weakly efficient (nondominated) set contains all efficient (nondominated) solutions and a set of special inefficient (dominated) solutions. It is a common misconception to think that this set consists of only the special inefficient (dominated) solutions, ignoring the fact that all efficient (nondominated) solutions are also in this set.

Definition 3. Given $\bar{Z}, \tilde{Z} \subseteq Z$ and $\varepsilon \in \mathbb{R}$, $\bar{Z}$ is said to *ε–dominate* $\tilde{Z}$ if there exists $\bar{\mathbf{z}} \in \bar{Z}$ for each $\tilde{\mathbf{z}} \in \tilde{Z}$ such that $\tilde{z}_i \leq \bar{z}_i + \varepsilon$ $\forall i$, and for at least one $\tilde{\mathbf{z}} \in \tilde{Z}$ there exists $\bar{\mathbf{z}} \in \bar{Z}$ such that $\tilde{z}_i < \bar{z}_i + \varepsilon$ for at least one i.

Let $Z_{ND} \subseteq Z$ be the nondominated set.

Definition 4. Let $\mathbf{z}^{IP} = (z_1^{IP}, z_2^{IP}, \ldots, z_m^{IP})$ be the ideal point where $z_i^{IP} = \max_{\mathbf{z} \in Z} z_i$.

Definition 5. Let $\mathbf{z}^{NP} = (z_1^{NP}, z_2^{NP}, \ldots, z_m^{NP})$ be the nadir point where $z_i^{NP} = \min_{\mathbf{z} \in Z_{ND}} z_i$.

2.2. Measure of representativeness

We use the measure, additive epsilon-indicator, developed by Zitzler, Thiele, Laumanns, Fonseca, and Grunert Da Fonseca (2003) and also used by Masin and Bukchin (2008) to assess the representativeness of the generated subsets for minimization problems. We adapt this measure for maximization-type objectives and refer to it as coverage gap in the rest of this paper.

Definition 6. Let $R \subseteq Z$ be a representative subset and $\mathbf{y} = (y_1, y_2, \ldots, y_m) \in R$ denote a representative point. We denote $\mathbf{r(z)} = (r_1(\mathbf{z}), r_2(\mathbf{z}), \ldots, r_m(\mathbf{z})) \in R$ as the representative point of $\mathbf{z} \in Z$ where:

$$\max_{1 \leq i \leq m} (z_i - r_i(\mathbf{z})) = \min_{\mathbf{y} \in R} \left\{ \max_{1 \leq i \leq m} (z_i - y_i) \right\}.$$

Let $\alpha_R(\mathbf{z}) = \min_{\mathbf{y} \in R} \{\max_{1 \leq i \leq m} (z_i - y_i)\}$ measure how well set R covers point $\mathbf{z}$. Then, the coverage gap of R, α_R, is determined by the worst represented point $\mathbf{z}^*$ (the point that has the maximum $\alpha_R(\mathbf{z})$ value). That is, $\mathbf{z}^* = \text{argmax}_{\mathbf{z} \in Z}\, \alpha_R(\mathbf{z})$ and $\alpha_R = \alpha_R(\mathbf{z}^*)$.

To clarify, consider the following example.