Pareto front generation with knee-point based pruning for mixed
discrete multi-objective optimization

Juseong Lee 1 · Sang-II Lee 2 · Jaemyung Ahn 1 · Han-Lim Choi 1

Received: 9 May 2017 / Revised: 30 December 2017 / Accepted: 24 January 2018 / Published online: 19 February 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract
This note proposes an algorithm to generate the Pareto front of a mixed discrete multi-objective optimization problem based on
the pruning of irrelevant subproblems. The knee point is introduced as a new reference point for pruning decision. The point can
overcome the drawback of the existing reference point – over-pruning, and be naturally defined and used in the context of multi-
objective optimization. The validity of the proposed procedure is demonstrated through case studies.

Keywords Pareto front · Mixed-discrete optimization · Multi-objective optimization · Pruning · Knee point

1 Introduction
This note concerns a mixed-discrete multi-objective optimization (MOO) problem defined as follows:

$$\begin{align*}
\min_{x} & \quad J(x) = \min_{y, z} \left[ J_1(y, z), \ldots, J_m(y, z) \right]^T \\
\text{subject to} & \quad g(y, z) \leq 0, \quad h(y, z) = 0 \\
& \quad l_i \leq y_i \leq u_i \quad (i = 1, \ldots, n_c) \\
& \quad z_j \in Z_j = \{ z_{j,1}, \ldots, z_{j,d_j} \} \quad (j = 1, \ldots, n_d)
\end{align*}$$

where $x = [y \ z]$ is the design vector, $y$ / $z$ are the continuous / discrete components of $x$ whose dimensions are respectively $n_c$ / $n_d$. $m$ is the number of objectives, $g$ / $h$ are the inequality / equality constraint vectors, and $Z_j$ is the set of values that $j$th discrete
design variable ($z_j$) can take ($|Z_j| = d_j$). The Pareto optimal solution of the problem ($\lambda^*$) is defined as follows:

$$\lambda^* \equiv \{ x^* \in \lambda \mid \exists x \in \lambda \setminus \{ x^* \} \text{ s.t. } J_i(x) \leq J_i(x^*) \quad \forall i, \quad J_j(x) < J_j(x^*) \quad \exists j \}$$

where $\lambda$ denotes the set of feasible design vectors. In addition, the Pareto front ($J^*$) is defined as the points in objective space corresponding to $\lambda^*$ as follows:

$$J^* = \{ J(x) \mid x \in \lambda^* \}$$

Hong et al. (2015) proposed a procedure to solve the bi-objective version of this problem ($m = 2$) based on two-phase pruning to eliminate irrelevant subproblems. A subproblem instantiated by specifying the discrete design variable is pruned out if its reference point (Phase A: utopia point, Phase B: center point) is dominated by the “master front,” which is defined in Section 2. Their method performed effectively for some mixed-discrete bi-objective optimization (BOO) problems. However, the center point used in their second phase may cause unsuccessful pruning decision depending on the shape of the resultant Pareto front. In addition, the application of the pruning-based Pareto front generation for general multi-objective optimization was not addressed in their study.
This paper proposes formulation/procedure that can handle three or more objective functions and utilizes a new reference point (the knee point) for Phase B pruning to overcome the aforementioned disadvantages of the previous study.

2 Review: Pareto-front generation for mixed discrete bi-objective optimization with center-point based pruning

The Pareto front generation procedure with center point based pruning by Hong et al. (2015) – referred to as C-Pruning in the rest of this note – is composed of two phases that adopt the utopia points and the center points as the references for pruning decision, respectively.

[Phase A of C-Pruning]
(Step A-1) Computing anchor/utopia points of subproblems: The C-Pruning first generates \( Z \), the set of all feasible discrete variable combinations, and its index set \( K \). Then the subproblem \( P_k \) is instantiated by setting \( z = z_k \in Z \) in \( P \) as follows:

\[
\begin{align*}
\min_y J(y, z_k) &= [J_1(y, z_k), \ldots, J_m(y, z_k)]^T (P_k) \\
\text{subject to} & \\
g(y, z_k) &\leq 0, \ h(y, z_k) = 0 \\
l_i &\leq y_i \leq u_i (i = 1, \ldots, n_c)
\end{align*}
\]

The anchor points of \( P_k \) are defined as follows:

\[
J_k^U = J(Y_k^A, z_k) = [J_{k,1}^{A_1}, \ldots, J_{k,1}^{A_m}, \ldots, J_{k,m}^{A_1}, \ldots, J_{k,m}^{A_m}]^T \ (i = 1, \ldots, m)
\]

where \( Y_k^A = \arg\min_{y \in Y_k} J_i(y, z_k) \), or the design vector that minimizes the \( i^{th} \) objective, where \( Y_k \) denotes the set of feasible design vectors of \( P_k \). The utopia point of \( P_k \) is expressed as

\[
J_k^U = [J_{k,1}^{A_1}, J_{k,2}^{A_2}, \ldots, J_{k,m}^{A_m}]^T
\]

(Step A-2) Generating a master front: The index set of subproblems with efficient (non-dominated) utopia points \( (K_1^M) \) is identified using pairwise comparisons of utopia points as follows:

\[
K_1^M = \{ k \mid \exists k \in K \ \text{s.t.} \ J_k^U \leq J_k^U \}
\]

The solution of \( P_k (Y_k^* \) ) and associated sub Pareto front \( (J_k^*) \) are defined as follows:

\[
\begin{align*}
Y_k^* &= \{ y^* \in Y_k \mid \exists y \in Y_k \setminus \{ y^* \} \ s.t. \\
J_i(y, z_k) &\leq J_i(y^*, z_k) \ \forall i, \\
J_i(y, z_k) &< J_i(y^*, z_k) \ \exists j \}
\end{align*}
\]

\[
J_k^* = \{ J(y, z_k) \mid y \in Y_k^* \}
\]

In addition, we define a master front \( (J^M) \) as the set of non-dominated solutions out of the collection of sub Pareto fronts \( (J_k^*) \) for \( k \in K_1^M \):

It is assumed that the Pareto front of subproblem \( (P_k) \) can be found with NBI (Das and Dennis 1998; PK Shukla 2007) or AWS (Kim and de Weck 2006; Zhang and Gao 2006; Hwang and Masud 2012). This implies that a gradient-based method is used to solve the NLPs instantiated during NBI/AWS, which works only if the subproblem \( (P_k) \) is differentiable. Note that the overall Pareto is can be non-differentiable or even discontinuous though it is created by “patching” sub Pareto fronts \( (J_k^*) \).

(Step A-3) Utopia point based pruning: Subproblem \( P_k \) is pruned out if its utopia point is dominated by the master front. The index set of pruned subproblems during the utopia point based pruning \( (K_2^U) \) is defined as follows:

\[
k_2^U = \{ k \mid \exists J(x) \in J^M \ s.t. \ J(x) \leq J_k^U \}
\]

[Phase B of C-Pruning]
(Step B-1) Computing center points of subproblems: The center point of a subproblem \( (J_k^C = J(Y_k^C, z_k) \) ) is obtained by solving the following NLP.

\[
y_k^C = \arg\min_y J^1(y, z_k)
\]

subject to constraints for \( P_k \) and

\[
\frac{J_{1,k}^{A,1} - J_{k,2}^{A,1}}{J_{k,2}^{A,1} - J_{k,1}^{A,1}} \leq 0
\]

Figure1-(a) shows geometric interpretation of the center point. The constraint representing the feasible region (10) is expressed as gray area. The center point can be found by minimizing \( J_1 \) within the feasible region. This algorithm is
applicable to bi-objective case and the value of \( m \) introduced in \( P_k \) is selected as 2.

(Step B-2) Center point based pruning: A subproblem whose center point \( (J^C_k) \) is dominated by the master front \( (J^M) \) is pruned out. The index set of eliminated subproblems during the center point based pruning \( (K^C_∅) \) is defined as

\[
K^C_∅ = \{ k | \exists J(x) \in J^M \text{ s.t. } J(x) \leq J^C_k \}
\]

(Step B-3) Generating Pareto front: The (approximate) solution of the original problem is obtained by combining sub Pareto fronts for \( k \in K \cup K^C_∅ \). Note this solution is exact if \( K^C_∅ \) is identical to the set of all irrelevant subproblems \( (K_∅) \).

While the C-Pruning effectively prunes out irrelevant subproblems for practical mixed-discrete BOOs, it has some disadvantages as well. First, their formulation was developed in the context of two objectives. Its extension to three or higher dimension, which is not very straightforward, was not provided in their work. Secondly, the center point can omit parts of the true Pareto front depending on the shape of the true Pareto front, which case is illustrated in Fig. 1. C-Pruning considers \( J^*_k \) presented in Fig. 1-(b) as irrelevant because \( J^C_k \) is dominated by \( J^M \). However the bulged part of \( J^*_k \) is actually Pareto optimal.

3 Knee point based pruning for Pareto front generation of a mixed-discrete multi-objective optimization

The knee point based pruning algorithm, which is referred to as \( K \)-Pruning in the rest of this note, modifies Phase B of the C-Pruning algorithm by changing its reference point for pruning decision (from center point to knee point). The knee point is conceptually the most bulged region, which has a couple of different definitions (Das 1999; Sudeng and Wattanapongsakorn 2015; Rachmawati and Srinivasan 2009).

[Phase B of K-Pruning]
(Step B-1) Computing Knee points of subproblems: The second phase of K-Pruning starts with solving the following optimization \( (\Phi^K_k) \) for \( k \in K \backslash (K^C_∅ \cup K^C_∅) \), which is the distance based method suggested by Das (1999).

\[
[y^K_k, β^K_k, t] = \arg \max_{y, β, t} t(\Phi^K_k)
\]

subject to constraints of \( \Phi^K_k \) and additional constraints:

\[
Φ_k β + n_k = J(y, z_k)\]
\[
\sum_{j=1}^m β_j = 1, β_j ≥ 0 \quad (j = 1, \cdots, m)
\]

where \( Φ_k = [J^{A_1}_k, \cdots, J^{A_m}_k] \) is the payoff matrix of \( P_k \), \( β = [β_1, \cdots, β_m] \), and vector \( n_k \) is normal to plane \( Φ_k β \). Das and Dennis (1998) mentioned that if \( n_k \) is not available, a
quasi-normal vector \( \mathbf{n} \) can be used for NBI method. For example, a vector normal to the plane \( \Phi_\beta \) can be suggested as \( \mathbf{n} \), where \( \Phi = [J^{A,1}, \ldots, J^{A,m}] \) is the payoff matrix of \( P \) \( (J^{A,i}: \text{anchor point of } P \text{ associated with } i^{th} \text{ objective}). \)

The knee point of \( P_k \) \( (J^K_k) \) is defined using \( y^K_k \) as

\[
J^K_k = J(y^K_k, z_k)
\]

Geometric interpretation of knee point is presented in Fig. 2.

(Step B-2) Knee-point based pruning: If \( J^K_k \) is dominated by \( J^M \), we can conclude that \( P_k \) is irrelevant.

\[
\mathcal{K}^K_{\emptyset} = \{ k | \exists J(x) \in J^M \text{ s.t. } J(x) \leq J^K_k \}
\]

(Step B-3) Generating Pareto front: \( \mathcal{J}^* \) can be generated by combining \( J^K_k \) for \( k \in \mathcal{K} \setminus (\mathcal{K}^M \cup \mathcal{K}^K_{\emptyset}) \) and extracting non-dominated solutions.

The K-Pruning has three meaningful advantages over the C-Pruning. First, K-Pruning is applicable to problems with more than two objectives since the knee point is properly defined in general \( m \)-dimensional objective space. Secondly, the knee point provides an intuitive and geometrically easy-to-understand reference point for pruning decision even for a complicated sub Pareto front. Figure 3 compares the reference points for pruning decision based on the center point and the knee point for three different sub Pareto front geometries – (a) convex and symmetric, (b) convex and skewed, and (c) nonconvex. Both methods provide proper references in Fig. 3-(a). However, in Fig. 3-(b) and 3-(c), the center point does not represent the sub Pareto front for pruning decision as the knee point, which can lead to over-pruning of relevant subproblems. This issue will be discussed in the first case study (TP1). Lastly, K-Pruning can find all knee points of the original problem \( P \) if each subproblem is convex. Note that the knee regions are where the maximum trade-off of objective functions takes place and thus can provide attractive design alternatives (Branke et al. 2004; Deb and Gupta 2010).

The effectiveness of the proposed K-Pruning algorithm is demonstrated through case studies in the next section.

4 Case study

4.1 Test problem 1: Nonconvex sub Pareto fronts

Test problem 1 (TP1) is a bi-objective problem that can compare the performance of the two pruning methods. TP1 in-

### Table 1 Pruning characteristics of TP1

| Method     | \(|\mathcal{K}|\) | \(|\mathcal{K}^M\)\) | \(|\mathcal{K}^L\)\) | \(|\mathcal{K}^K_{\emptyset}\)\) | \(|\mathcal{K}_{\emptyset}\)\) | Over-pruning | Under-pruning |
|------------|------------------|---------------------|---------------------|--------------------------------|-----------------|--------------|--------------|
| C-Pruning  | 27               | 5                   | 0                   | 22                             | 18              | 6            | 2            |
| K-Pruning  | 16               |                      |                     |                                | 0               | 0            | 2            |

### Table 2 Pruning characteristics of TP2 (K-Pruning)

| \(|\mathcal{K}|\) | \(|\mathcal{K}^M\)\) | \(|\mathcal{K}^L\)\) | \(|\mathcal{K}^K_{\emptyset}\)\) | \(|\mathcal{K}_{\emptyset}\)\) | Over-pruning | Under-pruning |
|------------------|---------------------|---------------------|--------------------------------|-----------------|--------------|--------------|
| 34,992           | 2592                | 24,948              | 7206                           | 32,154          | 0            | 2753         |

© Springer
volves skewed/nonconvex sub Pareto fronts as presented in Fig. 3-(b) and (c). It is mathematically formulated as:

\[
\min_j \mathbf{J} = \min_j \left[ J_1(y, z), J_2(y, z) \right] \text{ (TP1)} \quad \forall 2 \leq y \leq 2, z_i \in \{1, 2, 3\}, z_2, z_3 \in \{-1, 0, 1\}
\]

where

\[
J_1(y, z) = C_{01} + z_2 - z_3 + A_{01} y^2 + B_{01} y^2 + (1 - 4A_{01}) y
\]

\[
J_2(y, z) = C_{02} + \frac{1}{2} z_2 - z_3 + A_{02} y^2 + B_{02} y^2 - 1 + 4A_{02} y
\]

\[
[A_{01}, B_{01}, C_{01}] = \left[ \frac{4}{45}, \frac{2}{45}, \frac{-6}{45} \right], \left[ \frac{0}{20}, \frac{-1}{5} \right], \left[ \frac{4}{45}, \frac{2}{45}, \frac{-6}{45} \right] (z_1 = 1, 2, 3)
\]

Note that \(z_1\) is a categorical variable that collectively changes the coefficients of the objective functions. Its total number of subproblems (|\(\mathcal{K}\)|) is 27. Table 1 indicates 6 out of 22 subproblems eliminated in Phase B of C-Pruning were actually relevant (over-pruning), and 2 irrelevant subproblems were not pruned (under-pruning). On the other hand, K-Pruning did not over-prune any relevant subproblems while 2 irrelevant subproblems were under-pruned. The obtained Pareto front is presented in Fig. 5-(a).

### 4.2 Test problem 2: DEB3DK with discrete variables

A tri-objective optimization problem referred to as DEB3DK (Branke et al. 2004) is modified and used as the second test problem (TP2). This problem was originally developed as a test case for studies to find knee regions (Rachmawati and Srinivasan 2009; Bechikh et al. 2011). Note that the original formulation of DEB3DK involved only continuous variables, and some of its parameters are switched to discrete design variables (\(z\)) in this test problem. The problem is mathematically formulated as follows:

\[
\min_j \mathbf{J} = \min_j \left[ J_1(y, z), J_2(y, z), J_3(y, z) \right] \text{ (TP2)}
\]

subject to

\[
J_1(y, z) = g(y, z) r(y, z) \sin \left( \frac{y_1 \pi}{2} \right) \sin \left( \frac{y_2 \pi}{2} \right) + z_5 z_6
\]

\[
J_2(y, z) = g(y, z) r(y, z) \cos \left( \frac{y_1 \pi}{2} \right) + z_5 z_7
\]

\[
g(y, z) = 1 + (9 \sum_{i=2}^{n} y_i) \left/ (n-1) \right.
\]

\[
r_1(y, z) = 5 + z_4 (y_1 - z_1)^2 + \frac{2 \cos(2 z_0 y_1)}{z_0}
\]

\[
r_2(y, z) = 5 + (20 - z_4) (y_2 - z_2)^2 + \frac{2 \cos(2 z_0 y_2)}{z_0}
\]

\[
r(y, z) = z_3 r_1 + (1 - z_3) r_2
\]

\[
z_i \in \mathcal{Z}_i \quad \forall i \in \{0, \ldots, 7\}
\]

\[
\mathcal{Z}_0 = \{1, 2\}, \mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3 = \{0.2, 0.5, 0.8\},
\]

\[
\mathcal{Z}_4 = \{5, 10, 15\}, \mathcal{Z}_5, \mathcal{Z}_6, \mathcal{Z}_7 = \{0, 2, 4, 6, 8, 10\}
\]

Pruning characteristics of TP2 is summarized in Table 2. TP2 has total 34,992 \((2 \times 3^4 \times 6^3)\) subproblems and K-Pruning concluded that only 2838 are relevant, which accounts for only 8% of the total number. Figure 5-(b) shows the Pareto front generated – 85 subproblems are relevant, and K-Pruning found all the relevant subproblems.

### 4.3 Test problem 3: Tri-objective nine bar truss problem

The third test problem (TP3) is created by modifying a “classic” structural optimization problem called Nine Bar Truss.
Problem (e.g. Mela et al. 2007; Hong et al. 2015). The structural configuration of the nine bar truss is presented in Fig. 4. Cross sectional areas of trusses $1 \sim 3$ ($x_1 \sim x_3$), and discrete choices of cross sectional area / material type combination for trusses $4 \sim 9$ ($x_4 \sim x_9$) are defined as decision variables of the problem. Note that variables $x_1$, $x_2$, and $x_3$ are continuous and the other variables are discrete ($y = [x_1, x_2, x_3]$, $z = [x_4, x_5, x_6, x_7, x_8, x_9]$). The problem is formulated using parameters summarized in Tables 3 and 4 as follows:

$$
\begin{align*}
\text{min } J &= \min_x \begin{bmatrix} J_1(x) \\ J_2(x) \\ J_3(x) \end{bmatrix} \\
&= \begin{bmatrix}
\frac{L}{9} \left( \sum_{i=1}^{9} A(x_i) E(x_i) \right) \\
\frac{2}{3} \left( \sum_{i=1}^{3} A(x_i) E(x_i) \right) + \frac{1}{x_2 E_0} + \frac{1}{x_3 E_0} \\
\frac{L}{9} \left( \sum_{i=4}^{9} A(x_i) E(x_i) \right)
\end{bmatrix} 
\end{align*}
$$

subject to

$$
\begin{align*}
\frac{2}{3} \leq x_1 \leq 10, & \quad \frac{1}{3} \leq x_2 \leq 10, & \quad \frac{1}{3} \leq x_3 \leq 10 \\
x_i \in \{1, 2, 3, 4, 5\} & \quad (i = 4, \cdots, 9)
\end{align*}
$$

Three objectives to be minimized are the following: material cost ($J_1$), determined by length ($l_i$) and price per length ($C(x_i)$), vertical displacement of node $N$ ($J_2$), and the horizontal displacement of node $M$ ($J_3$). The cross-section area, Young’s modulus ($E(x_i)$), and unit cost ($C(x_i)$) of truss $i$ are given as follows:

$$
\begin{align*}
A(x_i) &= \begin{cases} x_i, & 1 \leq i \leq 3 \\ A_v, & 4 \leq i \leq 9 \end{cases} \\
E(x_i) &= \begin{cases} E_0, & 1 \leq i \leq 3 \\ E_v, & 4 \leq i \leq 9 \end{cases} \\
C(x_i) &= \begin{cases} x_i C_0, & 1 \leq i \leq 3 \\ C_v, & 4 \leq i \leq 9 \end{cases}
\end{align*}
$$

The number of unique discrete variable combinations ($|K|$) is 15,625 ($6^5$). K-Pruning removed 15,146 subproblems. The pruning characteristics of K-Pruning is presented in Table 3. The obtained Pareto front and over-pruned solution points are presented in Fig. 5-(c). The pruning characteristics of K-pruning for TP3 is presented in Table 5.

Note that $J_2$ and $J_3$ of TP3 are not in significant conflict. This is reflected in the projection of the Pareto front in $J_2$-$J_3$ plane, which does not present clear trade-off relationship. In practice, it is desirable that the trade-off between objectives of MOO are clearly explainable to obtain meaningful study results.

\subsection*{4.4 Comparison of K-pruning and multi-objective genetic algorithm (MOGA)}

The performance of the proposed pruning-based algorithm was compared with that of a multi-objective genetic algorithm

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$|K|$ & $|K|^\text{R}$ & $|K|^\text{U}$ & $|K|$ & $|K|$ & Over-pruning & Under-Pruning \\
\hline
15,625 & 25 & 0 & 15,146 & 15,146 & 37 & 401 \\
\hline
\end{tabular}
\end{table}
MOGA), one of the most popular heuristic approaches to obtain the Pareto front of a multi-objective optimization problem. The Non Sorting Genetic Algorithm – II (NSGA-II) developed by Deb (2001) included in the Global Optimization Toolbox of MATLAB was used for MOGA (MathWorks 2007). For all three test problems (TP1, TP2, and TP3), the two algorithms were compared based on the two criteria. (1) The ratio of solutions in Pareto front obtained by one algorithm that are dominated by the solutions obtained by the other algorithm; when this ratio is high, it can be concluded that a large portion of Pareto optimal solutions obtained by the algorithm is actually inefficient. (2) The number of function calls during the generation of the solution, which represents the computing cost of each algorithm. For fair comparisons, the number of populations for the MOGA were adjusted such that the computation times for the two methodologies are similar. The crossover fraction and the Pareto fraction used in the NSGA-II algorithm were set as 0.8 and 0.35, respectively.

The comparison results are summarized in Table 6. In TP1, none of the solutions obtained by K-Pruning was dominated by the solutions from MOGA, but 8.5% of the solutions obtained by MOGA were dominated by those from K-Pruning. This result indicate that the K-Pruning outperforms MOGA slightly from the perspective of the Pareto optimality of the obtained solution in this problem. The gap is much larger for TP2, where 15.6% of the solutions generated by MOGA were dominated by those from the K-Pruning, but there was no K-Pruning solution dominated by MOGA solution. It was observed that many of MOGA solutions for TP2 finally converged to irrelevant sub Pareto fronts, which yields a large fraction of inefficient solutions (see Fig. 6). The comparison result for TP3 was similar to that of TP1: the fractions were 1.8% (K-Pruning solutions dominated by MOGA) versus 7.1% (MOGA solutions dominated by K-Pruning).

Based on these results, it can be concluded that K-Pruning makes better performance for the proposed test problems while the magnitude of performance gaps vary depending on the characteristics of the problem. The pruning based algorithm proposed in this paper consistently creates high-quality (close to true optimum) Pareto front with relatively large computational cost, where the MOGA based approach has the trade-off between the accuracy of the obtained solutions and computational cost.

### 5 Discussion: Computational cost and limitation

The primary target of the proposed approach is the problems including “categorical” discrete variables, which cannot be relaxed as real numbers (e.g. material types: steel/aluminum/tungsten, moving mechanism: wheel/legs). While there are efficient algorithms based on relaxation of discrete variables, these algorithms do not work on problems involving the categorical variables because they cannot be relaxed.

It should be noted that the increase in the numbers of discrete variables/options that each of them can take can result in the explosion in prohibitively large number of subproblems that we have to solve. For example, if there are 10 discrete variables each of which can take 10 values, we have to find sub Pareto fronts associated with $10^{10}$ subproblems. Such a case can be addressed using multi-objective optimization approaches.
heuristics (e.g., multi-objective genetic algorithm); its performance would be highly dependent on the nature of the discrete variables as well.

Application of the proposed methodology to more mixed-discrete multi-objective optimization problems with various characteristics (e.g., non-differentiable/discontinuous Pareto front generation problems (Zhang and Gao 2006)) for its further validation can be a potential subject for future studies.

6 Conclusion

A procedure to generate the Pareto front of mixed-discrete multi-objective optimization problems using the knee-point based pruning (K-Pruning) is proposed. The procedure is developed by extension of existing method to solve a mixed-discrete bi-objective optimization using the center-point based pruning (C-Pruning). The new reference point introduced in this study can ensure that the region where high degree of trade-off between different objectives is included in the solution, which results in the reduction of the over-pruning – the disadvantage of the center point as the pruning reference. The knee point is naturally defined and utilized for multi-objective optimization problems. Case study with three test problems demonstrated the effectiveness of the proposed procedure for solving the mixed-discrete MOO, compared to the C-Pruning and MOGA.

References

Bechikh S, Said LB, Ghédira K (2011) Searching for knee regions of the Pareto front using mobile reference points. Soft Comput 15(9):1807–1823. https://doi.org/10.1007/s00500-011-0694-3
Branke J, Deb K, Dierolf H, Osswald M (2004) Finding knees in multi-objective optimization. In: parallel problem solving from nature (PPSN) 2004 - Lecture notes in Computer Science Vol 3242, pp 722–731
Das I (1999) On characterizing the ‘knee’ of the Pareto curve based on normal-boundary intersection. Struct Optim 18(2):107–115
Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multi-criteria optimization problems. Soc Ind Appl Math J Optim 8(3):631–657
Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester
Deb K, Gupta S (2010) Towards a link between knee solutions and preferred solution methodologies. In: Panigrahi B, Das S, Suganthan P, Dash S (eds) Swarm, evolutionary, and memetic computing (SEMCCO) 2010 – Lecture notes in computer science Vol 6466, pp 182–189
Hong S, Ahn J, Choi HL (2015) Pruning-based Pareto front generation for mixed-discrete bi-objective optimization. Struct Multidiscip Optim 51(1):193–198
Hwang C, Masud A (2012) Multiple objective decision making - methods and applications: a state-of-the-art survey. Springer, Berlin
Kim IY, de Weck O (2006) Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. Struct Multidiscip Optim 31(2):105–116
MathWorks (2007) Global optimization toolbox: gamultiobj. Retrieved December 27, 2017 (https://kr.mathworks.com/help/gads/gamultiobj.html)
Mela K, Koski J, Silvennoinen R (2007) Algorithm for generating the Pareto optimal set of multiobjective nonlinear mixed-integer optimization problems. In: 48th AIAA/ASME/ASC/ASE/ASC structures, structural dynamics, and materials conference. Structures, structural dynamics, and materials and co-located conferences. AIAA, Honolulu, Hawaii
Rachmawati L, Srinivasan D (2009) Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front. IEEE Trans Evol Comput 13(4):810–824
Shukla PK (2007) On the normal boundary intersection method for generation of efficient front. In: Shi Y, van Albada GD, Dongarra J, Sloot PMA (eds) Computational science – ICCS 2007. ICCS 2007. Lecture notes in computer science, vol 4487. Springer, Berlin
Sudeng S, Wattanapongsakorn N (2015) Finding knee solutions in multi-objective optimization using extended angle dominance approach. In: Information science and applications. Lecture notes in Electrical Engineering Vol 339, pp 673–679
Zhang WH, Gao T (2006) A min-max method with adaptive weightings for uniformly spaced Pareto optimum points. Comput Struct 84(28):1760–1769