A discussion of objective function representation methods in global optimization

Abstract

Non-convex optimization can be found in several smart manufacturing systems. This paper presents a short review on global optimization (GO) methods. We examine decomposition techniques and classify GO problems on the basis of objective function representation and decomposition techniques. We then explain Kolmogorov’s superposition and its application in GO. Finally, we conclude the paper by exploring the importance of objective function representation in integrated artificial intelligence, optimization, and decision support systems in smart manufacturing and Industry 4.0.

Keywords: global optimization, decomposition techniques, multi-objective, DC programming, Kolmogorov’s superposition, space-filling curve, smart manufacturing and Industry 4.0

1 Global optimization (GO) methods

Global non-convex programs can be solved using several approaches according to recent advances in GO literature (Pardalos and Rosen, 1986; Pardalos, 1991; Bomze et al., 1997; Pardalos and Wolkowicz, 1998; Horst et al., 2000; Nowak, 2005; Floudas and Pardalos, 2013; Horst and Pardalos, 2013; Floudas and Pardalos, 2014). These approaches can be divided into exact methods that can find and verify global solutions and heuristic methods, which only seek global solutions without checking optimality. Heuristics achieve a critical function in the optimization of large-scale non-convex problems and can be applied to provide upper bounds for global optimum, generate cuts and relaxations, and partition feasible sets.

Approximation algorithms are kinds of heuristics, wherein performance guarantee is considered estimated error (Fisher, 1980; Hochbaum et al., 1999; Ausiello et al., 2012; Vazirani, 2013). MIP approximation techniques work by approximating univariate functions to piecewise linear function with a performance guarantee for MINLP method. Goemans and Williamson (1995) solved a quadratic binary program using the MaxCut heuristic as first approximation algorithm.

In GO, an algorithm is called finite if it obtains and verifies a global solution in a finite number of steps. The exact methods are finite in finding and verifying solution. Moreover, simplex, active set, and enumeration methods are finite for solving LPs, convex QPs, and bounded integer or concave problems. However, interior point and solution methods for SQP as a nonlinear convex program are not finite.

All GO methods create a rough model of the program for finding global solutions. A GO method is called a sampling heuristic if the method uses a crude model based on a finite set of points. The considered regions of interest in sampling heuristic methods are bounded set. The distribution of points in this region is usually denser and should consider random behavior to obtain all possible solutions. In the continuous feasible region, the possible random sample is infinite, and a GO solution is not guaranteed. Moreover, the sample can prove that the method converges with probability that is arbitrarily close to 1. A GO method is called a relaxation-based method if the method uses relaxation as a crude model, such as a mathematical model, which is easier to solve than the original problem. The crude model influences the problem description. Modeling the problem in an aggregated form is efficient for sampling heuristics with few variables and a simple, feasible set in a disaggregated form for relaxation-based method with objective functions and constraints that can be relaxed.

Relaxation-based heuristics are classified into three relaxation-based methods classes, which include branch-and-bound methods. This method divides the GO problem...
into subproblems based on partitioning of the feasible set. Successive relaxation methods successively improve an initial relaxation without dividing it into subproblems. Heuristics retrieve potential solutions from a given relaxation without modifying the relaxation.

The MINLP solver technology should be further developed, and additional details on GO (Pardalos and Rosen, 1987; Pintér, 1996; Horst et al., 2000; Neumaier, 2004; Schichl, 2010; Horst and Pardalos, 2013; Horst and Tuy, 2013), MINLP methods (Floudas et al., 1989; Grossmann and Kravanja, 1997; Grossmann, 2002; Tawarmalani and Sahinidis, 2002; Floudas, 2013), and sampling heuristics (Torn and Zilinskas, 1989; Boender and Romeijn, 1995; Strongin and Sergeyev, 2000) should be identified. In summary, GO methods can be classified as follows:

- **Sampling heuristics**: 1) Ultistar (Strongin and Sergeyev, 2000), 2) Clustering method (Becker and Lago, 1970; Dixon and Szegö, 1974; Torn and Zilinskas, 1989), 3) Evolutionary algorithm (Forrest, 1993), 4) Simulated annealing (Metropolis et al., 1953; Kirkpatrick et al., 1983; Locatelli M, 2002), 5) Tabu search (Glover and Laguna, 1997; Mart et al., 2018), 6) Statistical GO (Mockus J, 2012), 7) Greedy randomized adaptive search procedure (Resende and Ribeiro, 2003; Hirsch et al., 2007)
- **Relaxation-based heuristics**: 1) Rounding heuristics (Mawengkang and Murtagh, 1986; Goemans and Williamson, 1995; Burkard et al., 1997; Zwick, 1999), 2) Lagrangian heuristics (Holmberg and Ling, 1997; Nowak and Römisch, 2000), 3) Deformation heuristics (Moré and Wu, 1997; Schelstraete et al., 1999; Alperin and Nowak, 2005), 4) MIP approximation (Neumaier, 2004), 5) Successive linear programming (Palacios-Gomez et al., 1982).

### 2 Decomposition theory

Large-scale problems can be solved by splitting them into subproblems, which are coupled by a master problem either in parallel or in sequence. The Dantzig–Wolfe decomposition employs separability to decompose a GO problem to subproblems; this method is one of the first decomposition approaches for linear programming that could be optimized in parallel (Dantzig and Wolfe, 1960). This method considers dual problem as a master problem, which coordinates the solutions and iterative modifications of the subproblems. The extension of Dantzig–Wolfe decomposition was applied to the nonlinear convex problem, and the Lagrangian dual is solved by using the cutting plane method. Details regarding decomposition methods in convex and non-convex GO problems are found in (Kelly et al., 1998; Bertsekas, 1999; Horst et al., 2000; Babayev and Bell, 2001; Svanberg, 2002; Palomar and Chiang, 2006; Zhang and Wang, 2006; Boyd et al., 2007; Chiang et al., 2007; Zhong et al., 2013; Rockafellar, 2016; Rahmaniani et al., 2017; Nowak et al., 2018). In general, decomposition techniques can be classified into dual and primal decomposition methods.

#### 2.1 Primal decomposition

The following program with objective function is considered:

\[
\begin{align*}
\max_{y, x_i} & \sum_i f_i(x_i); \\
\text{subject to } & x_i \in X_i
\end{align*}
\]

(1)

where \( \forall i A_i x_i \leq y \), and \( y \in Y \). Primal decomposition can be applied wherever a coupling variable is set to a fixed value. Thereafter, the GO problem is decoupled into several subproblems for each \( i \) as:

\[
\begin{align*}
\max_{x_i} & f_i(x_i); \\
\text{subject to } & x_i \in X_i, A_i x_i \leq y
\end{align*}
\]

(2)

The master problem updates the coupling variable by solving:

\[
\begin{align*}
\max_y & \sum_i f_i^*(y); \\
\text{subject to } & y \in Y
\end{align*}
\]

(3)

where \( \lambda f_i^*(y) \) is the optimal objective value in (2). Therefore, Problems (2) and (3) are convex optimization problems if Problem (1) is convex. The gradient method solves Problem (3). Therefore, the optimal Lagrange multiplier, \( \lambda_i^*(y) \) in (2), the subgradient for each \( f_i^*(y) \) obtained by \( s_i(y) = \lambda_i^*(y) \), and Problem (2) can be solved by \( y \), where \( s(y) = \sum_i s_i(y) = \sum_i \lambda_i^*(y) \) is the global subgradient.
2.2 Dual decomposition

Dual decomposition is suitable when a coupling constraint and its relaxation exist. The GO problem is divided into several subproblems.

\[
\max_x \sum_i f_i(x_i); \text{ subject to: } x_i \in X_i, \forall i \sum_i h_i(x_i) \leq c \] 

(4)

The following equation is obtained by applying Lagrangian relaxation to the coupling constraint in Problem (4):

\[
\left\{ \max_x \sum_i f_i(x_i) - \lambda^T (\sum_i h_i(x_i) - c) ; \text{ subject to: } x_i \in X_i, \forall i \right\}
\] 

(5)

The Lagrangian subproblem for each \( i \) decouples Problem (5)

\[
\{ \max_x f_i(x_i) - \lambda^T (h_i(x_i) - c) ; \text{ subject to: } x_i \in X_i \}. \] 

(6)

The dual variables are updated from the master dual problem as follows:

\[
\left\{ \min_{\lambda} = \sum_i g_i(\lambda) + \lambda^T c; \text{ subject to: } \lambda \geq 0 \right\}, \] 

(7)

where \( g_i(\lambda) \) is the dual function obtained as the maximum value of the Lagrangian solved in Problem (6) for a given \( \lambda \). Thus, a gradient method can solve Problem (7), and the subgradient for each \( g_i(\lambda) \) obtained by \( s_i(\lambda) = -h_i(x_i^*(\lambda)) \), where \( x_i^*(\lambda) \) is the optimal solution of Problem (6) for a given \( \lambda \). The global subgradient is \( s(\lambda) = \sum_i s_i(\lambda) + c = c - \sum_i h_i(x_i^*(\lambda)) \). Problem (6) can be independently and locally solved with knowledge of \( \lambda \).

3 Objective function representation based on decomposition methods

3.1 Separable optimization

The choice of decomposition (of objective function) influences the choice of the algorithm for solving the corresponding mathematical program.

**Definition 1:** Separable optimization Problem (Horst et al., 2000)

\[
\{ \min_{x \in \mathbb{R}} F_0(x) \text{ subject to: } F_i(x) \leq b_i, \ l_i \leq x_i \leq u_i, \ i = 1, \ldots, m \},
\] 

(8)

where \( F_i(x) = \sum_{j=1}^n F_{ij}(x_j), \ i = 0,1,\ldots,m \).

3.2 Factorable optimization

McCormick (1983, 1974, 1976) introduced factorable programming. A factorable program takes the following form

\[
\{ \min_{x \in \mathbb{R}^n} X^L(x) \text{ subject to: } l_i \leq X_i(x) \leq u_i, \ i = 1, \ldots, L-1 \},
\] 

(9)

where \( X^i : \mathbb{R}^n \to \mathbb{R} \)

\( X(x) = x_i \) for \( i = 1, \ldots, n \) and \( X^p(x), p = 1, \ldots, i-1 \), function \( X^i \) is \( X^i(x) = \sum_{\rho=1}^{i-1} T^{ho}_p(X^p(x)) + \sum_{\rho=1}^{i-1} \sum_{\phi=1}^p V^\phi_p (X^p(x)).U_p \) \( \cdot \) \( X^\phi(x) \), where \( T \) s, \( U \) s, and \( V \) s are the transformation functions of a single variable. The lower and upper bounds \( l_i \leq u_i \) are given constants. The function \( X^i(x), i = 1, \ldots, L \) can be written as factorable functions. McCormick (1974) developed a factorable programming language integrated with SUMT (Mylander et al., 1971) for NLPs. The functions \( X^i(x), i = 1, \ldots, L \) are called concomitant variable functions (cvfs). The cvfs includes separable and quadratic terms.

3.3 Almost block separable optimization

The following problem is considered:

\[
\min_{x \in \mathbb{R}} f(x) = f_1(u,v) + f_2(v,y),
\] 

(10)

where \( x = (u,v,y) \in \mathbb{R}^n \) and \( u \in \mathbb{R}^{n_1}, v \in \mathbb{R}^{n_2}, y \in \mathbb{R}^{n_3} \)

\( n_1 + n_2 + n_3 = n \), and \( y \) are called complicated variables

usually \( n_1, n_2 \gg n_3 \)

Let \( \varphi_1(y) = \min_{u \in \mathbb{R}^{n_1}} f_1(u,y), \varphi_2(y) = \min_{v \in \mathbb{R}^{n_2}} f_2(v,y) \). The problem is equivalent to:

\[
\min_{y \in \mathbb{R}^{n_2}} \varphi_1(y) + \varphi_2(y).
\] 

(11)

If \( f_1 \) and \( f_2 \) are convex, then \( \varphi_1(y) \) and \( \varphi_2(y) \) are convex.

3.4 DC optimization problems

3.4.1 Continuous DC programming

One of the special non-convex programs is DC program. DC function and dual DC programming are defined as follows:

**Definition 2:** DC function (Horst et al., 2000; Wu et al., 2018)

A real-valued function \( f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, +\infty\} \) subject to:

\[
\{ f(x) = g(x) - h(x), \forall x \in \mathbb{R}^n \},
\] 

(12)

where \( g, h : \mathbb{R}^n \to \mathbb{R} \cup +\infty \) is a convex function and is a
DC function for any \( h \) and \( g \).

**Definition 3**: DC program (Horst et al., 2000; Wu et al., 2018)

The following model is called a DC program

\[
\begin{align*}
\{ & \min f_0(x) \text{ subject to : } f_i(x) \leq 0, \forall i = 1,2,...,n, \} , \\
& \text{if } f_i \text{ are DC functions (} i = 0,1,2,...,n \text{) and it is the same as the following DC program, then,} \\
\inf_{x \in \mathbb{R}^n} f(x) &= g(x) - h(x). 
\end{align*}
\]

**Hartman Theorem 1**

The following DC programs are equal:

\[
\begin{align*}
\sup_{x \in C} f(x) & : x \in C, f : C \text{ : convex} \\
\inf_{x \in \mathbb{R}^n} g(x) - h(x) & : x \in \mathbb{R}^n, g, h : \text{convex} \\
\inf_{x \in \mathbb{R}^n} g(x) - h(x) & : x \in C, f_1(x) - f_2(x) \\
& \leq 0, g, h, f_1, f_2, C : \text{all convex}
\end{align*}
\]

**Hartman Theorem 2**

A function \( f \) is locally DC if an \( \epsilon \)-ball on which DC exists. Every function that is locally DC is considered a DC proposition. Let \( f_i \) be DC functions for \( i = 1,...,m \). Thus, \( \{ \sum \lambda_i f_i(x) \text{ for } \lambda_i \in \mathbb{R} \}; \{ \max f_i(x) \}; \{ \min f_i(x) \}; \{ \Pi f_i(x) \}; \) and \( \{ f \} \) are twice continuously differentiable DC. Moreover, \( (g \circ f) \) is DC if \( f \) is DC and \( g \) is convex, and every continuous function on \( C \) (convex set) is the limit of a sequence of uniformly converging DC functions.

**Definition 4**: Subgradient of convex function (Horst et al., 2000; Wu et al., 2018)

A vector \( x^* \) is a subgradient of a convex function \( h \) at a point \( x \) if \( h(z) \geq h(x) + (x^*-z)^T \langle x \rangle \), where \( (x^*,z-x) = n \sum_{i=1}^n x_i y_i \) is the inner product of two vectors with the same dimension. The subdifferential of \( h(x) \) is the set of all subgradients.

**Definition 5**: Conjugate functions (Horst et al., 2000; Wu et al., 2018)

A conjugate function \( h^* : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty \) of a convex function \( h : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty \) is:

\[
h^*(p) : = \sup_{y \in \mathbb{R}^n} \langle y, x \rangle - h(x).
\]

**Theorem 3**: The conjugate function \( h^*(y) \) of \( h(x) \) is convex. If \( h(x) \) is a closed proper convex function, then the bi-conjugate of \( h \) is itself, that is, \( h^{**} = h \).

**Theorem 4 (Toland–Singer duality)**: Given closed convex functions \( g, h : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty \), then:

\[
\inf_{x \in \mathbb{R}^n} \{ g(x) - h(x) \} = \inf_{p \in \mathbb{R}^n} \{ h^*(p) - g^*(p) \}.
\]

**Definition 6**: DC algorithm (Horst et al., 2000; Wu et al., 2018)

The following algorithm is used for obtaining a local optimal solution for the DC program.

---

Step 0 Find an initial solution \( x^0 \in dom(g) \). Set \( t = 0 \).

Step 1 Find \( \rho^* \in \partial h(x^t) \).

Step 2 Find \( \rho^{t+1} \in \partial g(\rho^t) \), where \( \rho^t \) is the “conjugate” of \( \rho \).

Step 3 If \( x^{t+1} = x^t \), stop. Otherwise, set \( t = t + 1 \), go to Step 1.

where \( x^{t+1} = \arg\min_{x \in \mathbb{R}^n} \{ g(y) - h(x) - \langle p, y - x \rangle \} \).

---

3.4.2 Continuous relaxations for discrete DC programming

The positive support of \( x \in \mathbb{Z}^n \) is presented as follows:

\[
\sup x = \{ i \in \{1,2,...,n\} : x_i > 0 \}.
\]

The indicator vector \( \chi_S \) is defined by:

\[
\chi_S(i) = \begin{cases} 
1 & i \in S \\
0 & i \notin S 
\end{cases}
\]

**M**-convex and \( L^2 \)-convex are two common discrete functions:

1. \( M \)-convex functions are defined as \( \forall x,y \in \mathbb{Z}^n \) and \( i \in \sup x = \{ i \in \{1,2,...,n\} : x_i > 0 \} \).

   The indicator vector \( \chi_S \) is defined by:

   \[
   \chi_S(i) = \begin{cases} 
1 & i \in S \\
0 & i \notin S 
\end{cases}
\]

2. \( L^2 \)-convex functions are defined as \( \forall x, y \in \mathbb{Z}^n \), \( h : \mathbb{Z}^n \rightarrow \mathbb{Z} \cup +\infty \) is \( L^2 \)-convex if it satisfies:

   \[
   h(x) + h(y) \geq \min \{ h(x-x_i) + h(x+x_i) \},
   \]

   \[
   \min_{j \in \sup x \cup \sup y} h(x-x_j + x_j) + h(y+y_j - x_j).
   \]

Consider the following discrete DC program:

\[
\{ \inf f(x) = g(x) - h(x) \} \text{ subject to : } x \in \mathbb{Z}^n.
\]

The four kinds of discrete DC programs include \( M^2 - L^1, M^2 - M^2, L^3 - L^1, \) and \( L^3 - M^2 \), wherein the first three are NP-hard, and the last one on \( \{0,1\}^n \) is in \( P \), can be defined on the basis of \( M \)’s and \( L^2 \)-convex function definitions (Kobayashi, 2014; Maehara et al., 2018). We assume functions \( g, h : \mathbb{Z}^n \rightarrow \mathbb{R} \cup +\infty \). The effective domain of \( g \) is \( \text{dom}_2 g := \{ x \in \mathbb{Z}^n : g(x) < +\infty \} \).

The convex closure \( \overline{g}(x) : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty \) of \( g \) is:

\[
\overline{g}(x) = \sup \{ s(x) : s \text{ is an affine function},
\]

\[
s(y) \leq g(y)(y \in \mathbb{Z}^n). \]

A convex extension \( \overline{g} : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty \) of \( g \) is a convex function with the same function value on \( x \in \mathbb{Z}^n \).

We assume

\[
\tilde{f}(x) := \overline{g}(x) - h(x). \text{ Then } \tilde{f}(x) := g(x) - h(x), \forall x \in \mathbb{Z}^n.
\]

Thus:

\[
\inf_{x \in \mathbb{Z}} \{ g(x) - h(x) \} = \inf_{x \in \mathbb{Z}} \tilde{f}(x) \geq \inf_{x \in \mathbb{R}} \tilde{f}(x).
\]
Theorem 5: For convex extensible functions \( g, h : \mathbb{Z}^n \times \mathbb{R} \cup + \infty \) with \( \text{dom} g \) bounded and \( \text{dom} g, h \in \text{dom} h \):
\[
\inf_{z \in \mathbb{Z}^n} \{ g(z) - h(z) \} = \inf_{x \in \mathbb{R}^n} \{ \bar{g}(x) - \bar{h}(x) \},
\]
where \( \bar{g}(x) \) is the linear closure of \( g(x) \), and \( \bar{h}(x) \) is any convex extension of \( h(x) \).

We found that the discrete DC programming (20) is equivalent to the corresponding continuous relaxation DC programming based on Theorem 5.

3.5 DI optimization problems

Total and partial monotonicity are related to monotoncity for all and some variables with many GO applications. The d.i. monotonic optimization with increasing functions in \( \mathbb{R}_+^n \) can be generally described as follows:
\[
\{ \min f(x) - g(x) \text{ subject to } f_i(x) - g_i(x) \leq 0, \ 1 \leq i \leq m \}. \tag{24}
\]
Let \( g(x) = 0 \), and,
\[
\forall i, f_i(x) - g_i(x) \leq 0 \iff \max_{1 \leq i \leq m} \{ f_i(x) - g_i(x) \} \leq 0 \iff F(x) - G(x) \leq 0,
\]
with increasing \( F \), and \( G \)  \( F(x) = \max_{1 \leq i \leq m} \{ f_i(x) + \sum_{i \neq j} g_i(x) \}, G(x) = \sum_{i \leq m} g_i(x) \).

Then, the problem is reduced to:
\[
\{ \min f(x) \text{ subject to } F(x) + t \leq F(b), \ G(x) + t \leq g(b), \ 0 \leq t \leq F(b) - F(0), \ x \in \mathbb{R}_+^n \}. \tag{25}
\]
For any \( x, x' \) where \( x' \leq x \), if \( x \in G \), then \( x' \subseteq G \), a set \( G \subseteq \mathbb{R}_+^n \) is normal.

Many GO problems, including polynomial, multiplicative, Lipschitz optimization problems, and non-convex quadratic programming, can be considered monotonic optimization problems.

3.6 Decomposition and multi-objective optimization

We consider the following problems:
\[
P1 : \min_{x \in D} F(x) = f_1(x) + \ldots + f_k(x), \tag{27}
\]
\[
P2 : \min_{x \in D} f(x) = (f_1(x), \ldots, f_k(x)). \tag{28}
\]
Objective function \( F(x) \) in many GO problems can be represented by the summation of \( k \) relatively simple functions as \( F(x) = f_1(x) + f_2(x) + \ldots + f_k(x) \). P2 is a multi-objective optimization problem. Let \( E(f, D) \subseteq D \) be the set of all Pareto optimal solutions in \( D \). We obtain the following theorems for optimal solutions of P1 and the optimal Pareto frontier of P2.

Theorem 6: If \( \bar{x} \) is an optimal solution of P1, then \( x \in E(f, D) \) of P2.

Theorem 7: Let \( h_i(x) \) be a monotonic increasing function for \( i = 1, \ldots, k \). We consider the multi-objective optimization problem \( \min_{x \in D \subseteq \mathbb{R}^n} h(x) = (h_1(f_1(x)), \ldots, h_k(f_k(x))) \). Then, \( E(f, D) = E(h, D) \) (Miettinen, 1999; Chinchnuuan and Pardalos, 2007; Pardalos et al., 2008; Du and Pardalos, 2013; Migdalas et al., 2013; Pardalos et al., 2017)

Theorems 1 and 2 show that the extended Pareto optimal frontier set \( E(h,D) \) can be obtained by solving P2 and searching for the optimal \( \bar{x} \) of P1 from \( E(h,D) \).

P2 can be a multi-objective optimization problem (MaOP). The algorithms for solving MaOPs can be classified as: 1) Algorithm adaptation methods, which modify/extend the classical EMO algorithms for solving MaOPs, including preference-based MOEA (PICEA; PBEA), Pareto-based MOEA (NSGA-II; SPEA2), indicator-based MOEA (HyPE; SMSEMOA), decomposition-based MOEA (MOEAD; M2M); and 2) Problem transformation methods, which transform the MaOP into a problem with few objectives, including objective selection (\( \sigma \)-MOSS; \( k \)-MOSS; L-PCA) and objective extraction (Gu, 2016). Refer to Gu (2016) and Mane and Rao (2017), for a review of solution algorithms and real-world applications of MaOPs, such as flight control system, engineering design, data mining, nurse scheduling, car controller optimization, and water supply portfolio planning.

MOEA/D is a mostly used method for solving P2. Its goals can be categorized as: 1) convergence to detect solutions close to the Pareto frontier; 2) diversity to determine well-distributed solutions; and 3) coverage to cover the entire Pareto frontier. Several MOEAs for these goals are found in literature, which can be broadly categorized under three categories, namely, 1) domination-, 2) indicator-, and 3) decomposition-based frameworks (Ehrgott and Gandibleux, 2000; Trivedi et al., 2017).

In MOEA/D literature, three decomposition methods, including the weighted sum (WS), the weighted Tchebycheff (TCH), and penalty based boundary intersection (PBI) approaches.

The \( i \)th subproblem of the WS approach is given as:
\[
\min g^{ws}(x|z_i) = \sum_{j=1}^m \lambda_i^j f_j(x). \tag{29}
\]
This method is efficient for solving convex Pareto solutions with min objective function.

The \( i \)th subproblem of the TCH approach is defined as follows:
\[
\min g^{ws}(x|z^*_i) = \max_{1 \leq j \leq m} \{ \lambda_i^j f_j(x) - z_j^* \}, \tag{30}
\]
where \( z^* = (z_1^*, \ldots, z_m^*)^T \) is the ideal reference point with \( z_j^* < \min \{ f_j(x) | x \in \Omega \} \) for \( j = 1, 2, \ldots, m \).

The \( i \)th subproblem of the PBI approach is defined as follows:

\[
\begin{align*}
\min g^{pbi}(x|\lambda_i)z^* &= d_1 + \theta d_2, \\
\text{where } d_1 &= \left\| (F(x) - z^*)^T \lambda_i \right\| / \| \lambda_i \| \\
\text{and } d_2 &= \left\| F(x) - \left( z^* - d_1 \frac{\lambda_i}{\| \lambda_i \|} \right) \right\|.
\end{align*}
\]

5 Conclusions

This paper reviewed different GO and decomposition methods on the basis of objective function representation. Many GO methods are derived from the branch and bound method, which are inefficient for finding a remarkable solution. This paper provides opportunity for additional research on decomposition techniques based on objective function representation, multi-objective optimization, and Kolmogorov’s superposition. The development of other parallel decomposition-based GO methods based on the objective function representation for MINLP, such as Decogo solver (Nowak et al., 2018), can be a challenging area in MINLP solver development. Kolmogorov theorem in GO will be discussed in future studies.

Industry 4.0 is known as the future of smart manufacturing and industrial revolution. Making decentralized decision is critical in Industry 4.0 (Marques et al., 2017). Horizontal and vertical integrations are two principal characteristics in Industry 4.0. Decentralized decision support systems are needed depending on the different types of decisions, including operational, tactical, real-time, and strategic. Many optimization problems are integrated with artificial intelligence in Industry 4.0, in which decision makers (DMs) should make a decentralized decision. This paper will help DMs in Industry 4.0 represent their objective function based on different GO techniques, such as Kolmogorov’s superposition and DC programming, which can be solved separately. Finally, Khakifirooz, Pardalos, et al. (2018) and Khakifirooz, Chien, et al. (2018) reported that applications of non-convex optimization in decision support system development for smart manufacturing and Industry 4.0 can be a challenging direction for future research.

Acknowledgements Professor Pardalos’ research is partially supported by the Paul and Heidi Brown Preeminent Professorship at ISE, University of Florida. Dr. Mahdi Fathi would like to thank Prof. Murray Brown and Mrs. Helen Brown for their encouragement and support during this research.

References


