A Biobjective Perspective for Mixed-Integer Programming

Jiao Liu, Yong Wang, *Senior Member, IEEE*, Bin Xin, *Member, IEEE*, and Ling Wang

***Abstract*—A mixed-integer programming (MIP) problem contains not only constraints but also integer restrictions. Integer restrictions divide the feasible region defined by constraints into multiple discontinuous feasible parts with different sizes. Several popular methods (e.g., rounding and truncation) have been proposed to deal with integer restrictions. Although it is easy for these methods to generate an integer, they tend to converge to an integer which is located in a feasible part with a big size. If the optimal solution is not in this feasible part, they are very likely to converge to a local optimal solution due to the loss of diversity of the population. To overcome this shortcoming, a biobjective optimization-based two-phase method is proposed in this paper. In the first phase, a measure function is designed to compute the degree that a solution violates integer restrictions. By employing this measure function as the second objective function and removing integer restrictions, a MIP problem is transformed into a constrained biobjective optimization problem (CBOP). It can be proven that the Pareto optimal solution of the transformed CBOP which satisfies integer restrictions is the optimal solution of the original MIP problem. To solve the transformed CBOP, a new comparison rule is designed. After the first phase, the population can approach the Pareto optimal solution which satisfies integer restrictions. Then, the second phase is implemented to enhance the convergence precision and obtain the optimal solution. In addition, we design 12 test problems to verify the effectiveness of the proposed method. The results demonstrate that the proposed method shows better performance against five state-of-the-art evolutionary algorithms for MIP.**

***Index Terms*—Mixed-integer programming, integer-restriction-handling technique, constraint-handling technique, biobjective optimization, evolutionary algorithms.**

I. Introduction

Mixed-integer programming (MIP) problems arise in various fields such as chemical processes [1], electronic circuits [2], and communications [3]. In general, the mathematical model of a MIP problem can be formulated as:

$$\begin{aligned} &minimize: \ f(\mathbf{x},\mathbf{y}) \\ &subject\ to: g_k(\mathbf{x},\mathbf{y}) \le 0,\ k=1,\ldots,l \\ &\qquad h_k(\mathbf{x},\mathbf{y}) = 0,\ k=l+1,\ldots,p \\ &\qquad x_i^L \le x_i \le x_i^U,\ i=1,\ldots,n_1 \\ &\qquad y_i^L \le y_i \le y_i^U,\ i=1,\ldots,n_2 \\ &\qquad [\mathbf{x},\mathbf{y}] \ \in \ S \\ &\qquad y_i\ is\ an\ integer \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_{n_1})$ is the continuous decision vector, $\mathbf{y} = (y_1, y_2, \ldots, y_{n_2})$ is the integer decision vector, x_i^L and x_i^U are the lower and upper bounds of x_i, respectively, y_i^L and y_i^U are the lower and upper bounds of y_i, respectively, S is the decision space, $f(\mathbf{x},\mathbf{y})$ is the objective function, $g_k(\mathbf{x},\mathbf{y})$ is the kth inequality constraint, and $h_k(\mathbf{x},\mathbf{y})$ is the $(k-l)$th equality constraint. In total, the MIP problem in (1) has n_1 continuous decision variables, n_2 integer decision variables, l inequality constraints, and $(p-l)$ equality constraints.

The degree of constraint violation of $[\mathbf{x},\mathbf{y}]$ is calculated as follows:

$$G_k(\mathbf{x},\mathbf{y}) = \begin{cases} max\{0, g_k(\mathbf{x},\mathbf{y})\}, & 1 \le k \le l \\ max\{0, |h_k(\mathbf{x},\mathbf{y})| - \delta\}, & l+1 \le k \le p \end{cases} \tag{2}$$

$$G(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{p} G_k(\mathbf{x},\mathbf{y}) \tag{3}$$

where δ is a positive tolerance value to relax equality constraints to a certain extent. The feasible region of a MIP problem is a set containing all feasible solutions:

$$\Omega = \{[\mathbf{x},\mathbf{y}] | G(\mathbf{x},\mathbf{y}) = 0, [\mathbf{x},\mathbf{y}] \in S\} \tag{4}$$

For a MIP problem, its feasible region Ω is defined by constraints, and then integer restrictions divide Ω into several discontinuous feasible parts. Next, the following MIP problem is employed as an example to illustrate this characteristic:

$$\begin{aligned} &minimize: 20x_1 + 10x_2 \\ &subject\ to: 5x_1 + 4x_2 \le 24 \\ &\qquad 2x_1 + 5x_2 \le 13 \\ &\qquad x_1, x_2 > 0 \\ &\qquad x_2\ is\ an\ integer \end{aligned} \tag{5}$$

As shown in Fig. 1, the blue area denotes the feasible region defined by constraints, and the red lines denote the discontinuous feasible parts defined by both constraints and

Manuscript received January 15, 2019; revised December 18, 2019, July 31, 2020, and September 19, 2020; accepted December 6, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61673397, Grant 61976225, and Grant 61673058, in part by the National Natural Science Fund for Distinguished Young Scholars of China under Grant 61525304, in part by the National Outstanding Youth Talents Support Program under Grant 61822304, and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06.
(Corresponding authors: Yong Wang; Bin Xin.)

J. Liu and Y. Wang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: liu_jiao@csu.edu.cn; ywang@csu.edu.cn).

B. Xin is with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, also with the Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China, and also with the Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China (e-mail: brucebin@bit.edu.cn).

L. Wang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: wangling@mail.tsinghua.edu.cn).

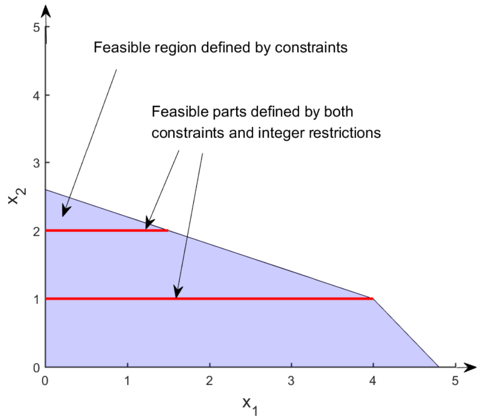


Fig. 1. The blue area denotes the feasible region defined by constraints, and the red lines denote the discontinuous feasible parts defined by both constraints and integer restrictions.

integer restrictions. Owing to the discontinuous feasible parts, a MIP problem is usually nonconvex and NP-hard [4].

To solve MIP problems, many classical methods have been proposed, such as branch and bound [4], cutting planes [5], and outer approximation [6], [7]. However, it is difficult for these methods to deal with nonconvex, nondifferentiable, or nonlinear MIP problems. Compared with classical methods, evolutionary algorithms (EAs) are insensitive to the landscapes of optimization problems such as nonconvexity and discontinuity, and easy to implement. During the past decades, there has been a growing interest in applying EAs to solve different kinds of optimization problems [8]-[14]. When solving MIP problems, EAs should be integrated with constraint-handling techniques and integer-restriction-handling (IRH) techniques due to the presence of both constraints and integer restrictions. To tackle constraints, a variety of constraint-handling techniques has been suggested. The current popular constraint-handling techniques can be mainly classified into three categories [15]: methods based on penalty functions [16], [17], methods based on the preference of feasible solutions over infeasible solutions [18]-[23], and methods based on multiobjective optimization [24]-[29]. For coping with integer restrictions, the most commonly used technique is transforming a continuous value into an integer by rounding or truncation [30]-[34]. Several other techniques have also been proposed to handle integer restrictions by directly generating an integer without any conversion [35]. However, these techniques tend to enter a feasible part with a big size. Note that if the optimal solution is located in a feasible part with a small size, these techniques may find a wrong feasible part and converge to a local optimal solution. Obviously, the performance of these techniques is significantly influenced by the sizes of the feasible parts.

Therefore, a question which arises naturally is whether we can transform a MIP problem into another problem without any integer restrictions, thus avoiding the influence of the feasible parts. In addition, in current IRH techniques, the solutions which do not satisfy integer restrictions are usually neglected. Note, however, that some of these solutions may be close to the optimal solution, which can contribute to the search of the optimal solution. Therefore, another interesting question is whether these solutions can be utilized.

Motivated by the above considerations, a novel biobjective optimization-based two-phase method (called BOToP) is proposed. In the first phase, a measure function is designed to calculate the degree that a solution violates integer restrictions. Then, a MIP problem is transformed into a constrained biobjective optimization problem (CBOP) by employing this measure function as the second objective function and removing integer restrictions[1]. It can be proven that the Pareto optimal solution of this CBOP which satisfies integer restrictions is the optimal solution of the original MIP problem. Moreover, we design a comparison rule to guide the population to approach this Pareto optimal solution in the first phase. Then, the second phase, which adopts rounding to handle integer restrictions, is implemented to enhance the convergence precision and obtain the final solution.

The main contributions of this paper can be summarized as follows:

- The original MIP problem is transformed into a CBOP in the first phase by a measure function. Compared with directly solving the original MIP problem, solving the transformed CBOP has the following advantage: the integer restrictions have been removed from the transformed CBOP; therefore, the feasible parts defined by integer restrictions have also been eliminated and the disadvantage that the population converges to a wrong feasible part can be alleviated.
- A comparison rule is designed, which aims to approach the Pareto optimal solution satisfying integer restrictions. When comparing two feasible solutions, their nondominated rankings are calculated. If they have the same nondominated ranking, either the objective function or the measure function is employed to compare them. Based on our comparison rule, some feasible solutions with low violation degree of integer restrictions and/or good objective function values can be retained. Thus, the information of these feasible solutions can be utilized and the diversity of the population can be enhanced.
- Twelve MIP test problems are designed in this paper. Systematic experiments have been conducted to compare BOToP with five state-of-the-art EAs on these 12 test problems. The results suggest that BOToP performs better than the compared methods.

The rest of this paper is organized as follows. Section II introduces differential evolution (DE) and multiobjective optimization problems (MOPs). The related work and the disadvantages of the commonly used IRH techniques are discussed in Section III. The details of the proposed BOToP are presented in Section IV. The experimental studies are implemented in Section V. Finally, Section VI concludes this paper.

[1]To the best of our knowledge, BOToP is the first attempt to deal with a MIP problem by transforming it into a CBOP.

II. Preliminary Knowledge

A. DE

DE is a very popular EA paradigm proposed by Storn and Price [36]. It contains four processes: initialization, mutation, crossover, and selection.

Initialization: There are NP individuals generated randomly in initialization:

$$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D}),\ i = 1, 2, \ldots, NP \tag{6}$$

where $\mathbf{x}_i$ is the ith individual and D is the number of decision variables.

Mutation: For each individual $\mathbf{x}_i$, a mutant vector $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D})$ is created. The following two DE mutation operators are used in this paper:

- DE/current-to-rand/1:

$$\mathbf{v}_i = \mathbf{x}_i + rand \times (\mathbf{x}_{r_1} - \mathbf{x}_i) + F \times (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{7}$$

- DE/rand-to-best/1:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + rand \times (\mathbf{x}_{best} - \mathbf{x}_{r_1}) + F \times (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{8}$$

where r_1, r_2 and r_3 are three mutually different integers chosen randomly from [1, NP], $\mathbf{x}_{best}$ is the best individual in the population, $rand$ is a uniformly distributed random number from [0, 1], and F is the scaling factor.

Crossover: Through the binomial crossover, a trial vector $\mathbf{u}_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,D})$ is generated based on $\mathbf{x}_i$ and $\mathbf{v}_i$:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } rand_j < CR \text{ or } j = j_{rand} \\ x_{i,j}, & \text{otherwise} \end{cases} \tag{9}$$

where $i = 1, 2, \ldots, NP$, $j = 1, 2, \ldots, D$, $CR \in [0, 1]$ is the crossover control parameter, $rand_j$ is a uniformly distributed random number between 0 and 1, and j_{rand} is an integer randomly selected from [1, D].

Selection: Between $\mathbf{x}_i$ and $\mathbf{u}_i$, the better one is selected into the next generation:

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \tag{10}$$

B. MOPs

Since the proposed method is based on multiobjective optimization, we next introduce MOPs and the related concepts. A MOP can be expressed as:

$$minimize : \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_q(\mathbf{x})) \tag{11}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_D) \in S$ is the decision vector containing D decision variables, S is the decision space, and $\mathbf{f}(\mathbf{x})$ is the objective vector containing q objective functions.

For MOPs, the most important concept is called Pareto dominance. Assuming that there are two decision vectors **a** and **b**, if $\forall i \in \{1, 2, \ldots, q\}$, $f_i(\mathbf{a}) \leq f_i(\mathbf{b})$ and $\exists j \in \{1, 2, \ldots, q\}$, $f_j(\mathbf{a}) < f_j(\mathbf{b})$, **a** is said to Pareto dominate **b**. If no decision vector in S can Pareto dominate **a**, then **a** is called a Pareto optimal solution. The set of all the Pareto optimal solutions is the Pareto set (PS). The Pareto front (PF) is the image of the PS in the objective space.

III. Related Work and the Disadvantages of Rounding and Truncation

A. Related Work

When solving MIP problems by EAs, a key problem is how to deal with integer restrictions. Current IRH techniques can be briefly classified into two categories: 1) indirect IRH techniques and 2) direct IRH techniques.

1) Indirect IRH Techniques: This kind of techniques transforms a continuous value into an integer. The most easily implemented and widely used indirect IRH techniques are rounding and truncation. Many EAs have been introduced to solve MIP problems based on rounding or truncation. For example, Deep *et al.* [34] proposed an extended version of the real-coded genetic algorithm (GA) [37], in which a mixed truncation method is employed to handle integer restrictions. Lampinen *et al.* [38] designed an improved DE to solve MIP problems. In this algorithm, the truncation is only implemented when evaluating a solution. In addition, Li *et al.* [39] proposed a discrete hybrid DE, Mohamed [40] suggested an efficient modified DE, Liao *et al.* [41] developed two hybrid DE, Luo *et al.* [42] devised an improved particle swarm optimization (PSO), Mohan *et al.* [43] proposed a controlled random search technique, Lin *et al.* [44] presented a co-evolutionary hybrid DE based on [45], and Jun *et al.* [46] employed a novel population initialization technology and a dynamic nonlinear scaling factor to enhance the search ability of DE. There are also many other indirect IRH techniques. For example, Babu *et al.* [1] introduced an additional equality constraint to deal with binary variables. Moreover, many EAs have been designed to solve MIP problems in the real world. Ponsich and Coello Coello [47] studied the performance of DE on solving process engineering problems, which contain large-size instances, constraints, and integer variables. Sahoo *et al.* [48] proposed an efficient hybrid approach based on GA and PSO, which is applied to solve reliability MIP problems in series-parallel and bridge systems. Hinojosa *et al.* [49] discussed the application of a mixed-integer-binary small-population-based evolutionary PSO for optimal power flow. Ho-Huu *et al.* [30] modified the improved (μ+λ) constrained DE [50] to solve layout truss optimization problems. Balamurugan *et al.* [2] proposed a hybrid integer-coded DE dynamic programming scheme to solve the economic dispatch problem.

2) Direct IRH Techniques: This kind of techniques generates an integer without any transformation. For instance, Kennedy *et al.* [51] extended the continuous PSO to deal with combinatorial optimization problems and used a binary bit to represent each dimension of an particle's position vector. Datta *et al.* proposed a real-integer-discrete-coded DE [35] and a real-integer-discrete-coded PSO [52] to solve MIP problems. Li *et al.* [53] designed a mixed-integer evolution strategy based on maximum entropy principle. Wu *et al.* [54] proposed an enhanced integer-coded PSO for solving system feeder reconfiguration problems.

B. Disadvantages of Rounding and Truncation

Rounding and truncation are widely accepted to handle integer restrictions. In this subsection, we are interested in

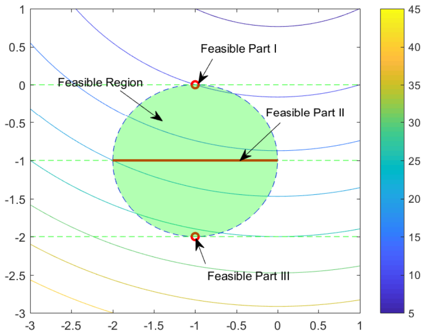


Fig. 2. Contours of the objective function and the feasible region of $P1$. The green area is the feasible region defined by constraints, the red line and red points are the feasible parts defined by both constraints and integer restrictions, and the optimal solution is located in feasible part I.

disclosing their disadvantages. To this end, an artificial test function named $P1$ is designed as an example:

$$\begin{aligned} P1: \ & minimize: \ (x_1-1)^2+(x_2-3)^2 \\ & subject\ to: (x_1+1)^2+(x_2+1)^2 \leq 1 \\ & \qquad\qquad x_1 \in [-3,1] \\ & \qquad\qquad x_2 \in \{-3,-2,-1,0,1\} \end{aligned} \tag{12}$$

The optimal solution of $P1$ is $(-1,0)$ and the optimal objective function value is 13. The contours of the objective function and the feasible region of $P1$ are shown in Fig. 2. The green area is the feasible region defined by constraints, and the red line and red points are the feasible parts defined by both constraints and integer restrictions. Obviously, $P1$ contains three discontinuous feasible parts, which are named as feasible part I, feasible part II and feasible part III, respectively. Note that the optimal solution is located in feasible part I.

When solving a MIP problem, our purpose is to find the optimal solution which satisfies both constraints and integer restrictions. To achieve this purpose, two tasks should be accomplished: 1) entering the feasible parts and 2) obtaining the optimal solution in the end. However, handling integer restrictions by rounding or truncation may have a negative effect on the accomplishment of these two tasks. Taking $P1$ as an example, the negative effects of rounding or truncation are explained as follows:

1) Influence on Entering the Feasible Parts: When solving $P1$, the population tends to enter feasible part II due to the fact that the size of feasible part II is significantly greater than that of feasible part I and feasible part III. As a result, feasible part I, which contains the optimal solution, is often neglected.

2) Influence on Finding the Optimal Solution: Once the whole population has entered a wrong feasible part, the individuals could not jump out due to the loss of diversity. As shown in Fig. 2, if all the individuals have entered feasible part II, the values of x_2 of these individuals are the same with each other, i.e., -1. Then, no matter what operator is adopted to generate offspring, the value of x_2 of this offspring is -1. Thus, the population stagnates in feasible part II.

To confirm our analysis, a set of experiments was implemented. Since rounding and truncation have the similar disadvantages, only the performance of rounding was tested in the experimental study. Two DE-based algorithms, named MIPDE-FR and MIPDE-ϵ, were designed. In MIPDE-FR and MIPDE-ϵ, Deb's feasibility rule [18] and ϵ constrained method [20] were employed to handle constraints, respectively. The parameters of MIPDE-FR and MIPDE-ϵ were the same as in Section V-B. The setting of ϵ in MIPDE-ϵ was the same with [20]. Fig. 3 and Fig. 4 provide a typical run derived from MIPDE-FR and MIPDE-ϵ on solving $P1$, respectively. We can observe that:

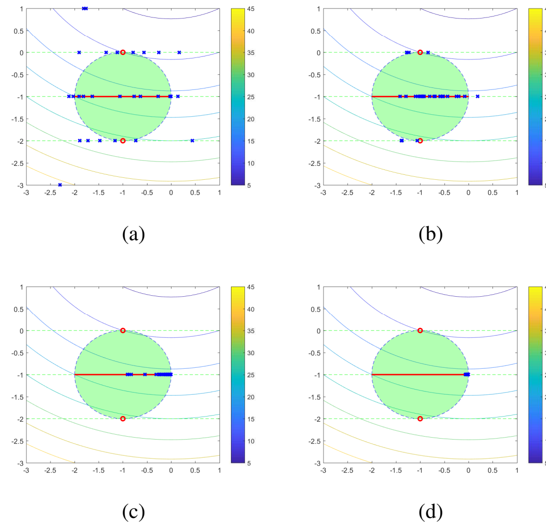


Fig. 3. Evolution of MIPDE-FR over a typical run on $P1$. (a) The 10th generation. (b) The 50th generation. (c) The 200th generation. (d) The 500th generation.

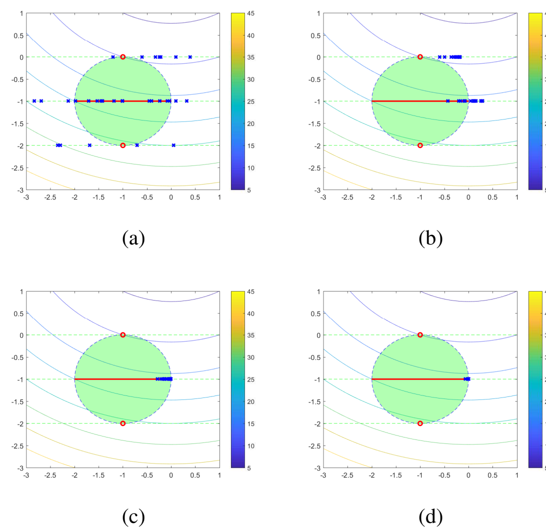


Fig. 4. Evolution of MIPDE-ϵ over a typical run on $P1$. (a) The 10th generation. (b) The 50th generation. (c) The 200th generation. (d) The 500th generation.

- From Fig. 3(a) and Fig. 3(b), MIPDE-FR can approach the feasible parts in the early stage. However, the individuals are hard to enter feasible part I because it is too small and easily ignored. As shown in Fig. 3(c), after

Algorithm 1 *BOToP*

1: $t = 0$; // t denotes the generation number
2: $FEs = 0$; // FEs denotes the number of function evaluations
3: $\mathbf{X}_0 \leftarrow$ *Initialization*;
4: $(\mathbf{X}_t, t, FEs) \leftarrow$ *The_first_phase*$(\mathbf{X}_0, t, FEs)$;
5: *FinalSolution* $\leftarrow$ *The_second_phase*$(\mathbf{X}_t, t, FEs)$;

200 generations, all the individuals enter feasible part II. Under this condition, the population cannot jump out because of the poor diversity. It can be seen from Fig. 3(d) that the population converges to a local optimal solution in the end.

- MIPDE-ϵ mainly searches around feasible part I and feasible part II in the early stage, as shown in Fig. 4(a) and Fig. 4(b). However, similar to MIPDE-FR, MIPDE-ϵ also misses feasible part I in the middle stage, as shown in Fig. 4(c). At last, premature convergence occurs, as shown in Fig. 4(d).

In summary, when rounding or truncation is used to deal with integer restrictions, the population is very likely to enter a feasible part with a big size. If the optimal solution is not in this feasible part, the population may be trapped into a local optimal solution due to the loss of diversity. Therefore, the performance of rounding or truncation is significantly influenced by the sizes of the feasible parts.

To overcome this disadvantage, one of the possible ways is to transform a MIP problem into another problem without any feasible parts, thus alleviating the influence of multiple feasible parts. In addition, when applying rounding or truncation to handle integer restrictions, the solutions which do not satisfy integer restrictions will be eliminated. Note that some of them may be very close to the optimal solution, which are beneficial to find the optimal solution. However, they are neglected unreasonably. Whether they can be utilized to enhance the search performance is also an interesting question. Based on the above considerations, we propose a biobjective optimization-based two-phase method for solving MIP problems, called BOToP.

IV. Proposed Approach

A. BOToP

The framework of BOToP is given in **Algorithm 1**. At first, an initial population is randomly produced in the decision space: $\mathbf{X}_0 = \{\mathbf{x}_{0,1}, \mathbf{x}_{0,2}, \ldots, \mathbf{x}_{0,NP}\}$, where $\mathbf{x}_{0,j}$ $(j \in \{1, 2, \ldots, NP\})$ is an individual containing both continuous decision variables and integer decision variables. During the evolution, BOToP consists of two main phases. The aim of the first phase is to make the population approach the optimal solution of a MIP problem. This task is accomplished by the following two steps: 1) transforming a MIP problem into a CBOP with no integer restrictions, and 2) solving the transformed CBOP by combining DE with a new comparison rule. After the first phase, the second phase is implemented to enhance the convergence precision and obtain the final solution. Similar to most existing methods, the second phase handles integer restrictions by rounding. Note that since the population has approached the optimal solution of a MIP problem after the first phase, the probability that the population stalls in a wrong feasible part greatly reduces, even though rounding is utilized in the second phase. Next, we introduce these two phases in detail.

B. The First Phase

1) Transformation: A MIP problem is transformed into the following CBOP:

$$\begin{aligned} minimize:\ & (f(\mathbf{x},\mathbf{y}),\ m(\mathbf{y})) \\ subject\ to:\ & g_k(\mathbf{x},\mathbf{y}) \le 0,\ k = 1, \ldots, l \\ & h_k(\mathbf{x},\mathbf{y}) = 0,\ k = l+1, \ldots, p \\ & x_i^L \le x_i \le x_i^U,\ i = 1, \ldots, n_1 \\ & y_i^L \le y_i \le y_i^U,\ i = 1, \ldots, n_2 \\ & [\mathbf{x},\mathbf{y}] \in S \end{aligned} \tag{13}$$

The main difference between the transformed CBOP and the original MIP problem is that integer restrictions are removed from the transformed CBOP and y_i is now treated as a real number. Meanwhile, a measure function $m(\mathbf{y})$, which can measure the degree of $\mathbf{y}$ violating integer restrictions, is considered as the second objective function. To measure the degree of $\mathbf{y}$ violating integer restrictions, the most direct method is to measure the distance between $\mathbf{y}$ and the integer solution nearest to $\mathbf{y}$ (denoted as $\mathbf{y}_{int}$). This distance can be represented as $||\mathbf{y} - \mathbf{y}_{int}||$, where $\mathbf{y}_{int}$ can be obtained by rounding $\mathbf{y}$ and $||\cdot||$ represents a kind of norm. In this paper, L_∞ norm[2] (denoted as $||\cdot||_\infty$) is employed as it is one of the simplest ways to calculate the distance between $\mathbf{y}$ and $\mathbf{y}_{int}$; thus, $m(\mathbf{y})$ is expressed as:

$$\begin{aligned} m(\mathbf{y}) &= ||\mathbf{y} - \mathbf{y}_{int}||_\infty \\ &= max(|y_1 - round(y_1)|, \ldots, |y_{n_2} - round(y_{n_2})|) \end{aligned} \tag{14}$$

where $round(\cdot)$ is the rounding operation. It is clear that if $m(\mathbf{y}) = 0$, then $\mathbf{y}$ satisfies integer restrictions. The relationship between the transformed CBOP and the original MIP problem can be described by the following two theorems.

Theorem 1: *The optimal solution of the original MIP problem is a Pareto optimal solution of the transformed CBOP.*

Proof: Suppose that $[\mathbf{x}^*, \mathbf{y}^*]$ is the optimal solution of the original MIP problem, and it does not belong to the PS of the transformed CBOP. According to the definition in Section II-B, there must exist a solution (denoted as $[\mathbf{x}', \mathbf{y}']$) which Pareto dominates $[\mathbf{x}^*, \mathbf{y}^*]$. Due to the fact that $[\mathbf{x}^*, \mathbf{y}^*]$ is the optimal solution of the original MIP problem, it must satisfy integer restrictions, i.e., $m(\mathbf{y}^*) = 0$. According to (14), it is obvious that the value of $m(\mathbf{y})$ is not negative, so $m(\mathbf{y}') \ge 0$. According to the definition of Pareto dominance, it can be concluded that $m(\mathbf{y}') = m(\mathbf{y}^*) = 0$ and $f(\mathbf{x}', \mathbf{y}') < f(\mathbf{x}^*, \mathbf{y}^*)$. However, this is a contradiction with the previous hypothesis: $[\mathbf{x}^*, \mathbf{y}^*]$ is the optimal solution of the original MIP. Therefore,

[2]Note that, based on different norms, $m(\mathbf{y})$ can be written as different forms. We investigated the influence of different forms in Section S-III of the supplementary file, and found that the performance of BOToP is insensitive to them.

Algorithm 2 *The_first_ phase*

Input: $\mathbf{X}_0$, $t = 0$, and $FEs = 0$
Output: $\mathbf{X}_t$, t, and FEs

Evaluate the f value, m value, and G value of each individual in $\mathbf{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,NP}\}$;
$FEs \leftarrow FEs + NP$;
$\mathbf{X}_{t+1} = \emptyset$;
while $FEs < MaxFEs/2$ **do**
  Generate a trial vector $\mathbf{u}_{t,j}$ for each individual $\mathbf{x}_{t,j}$ $(j \in \{1, 2, \dots, NP\})$ through DE's mutation and crossover operators, and obtain the offspring set $\mathbf{U}_t = \{\mathbf{u}_{t,1}, \mathbf{u}_{t,2}, \dots, \mathbf{u}_{t,NP}\}$;
  Evaluate the f value, m value, and G value of each individual in $\mathbf{U}_t$;
  $FEs \leftarrow FEs + NP$;
  Implement the nondominated sorting on all the feasible solutions in $\mathbf{X}_t$ and $\mathbf{U}_t$, and obtain their rankings;
  for $j = 1, 2, \dots, NP$ **do**
    Select the better one between $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ according to the comparison rule introduced in Section IV-B, and store it into $\mathbf{X}_{t+1}$;
  end for
  $t \leftarrow t + 1$;
end while

Algorithm 3 *The_second_phase*

Input: $\mathbf{X}_t$, t, and FEs
Output: The best individual in $\mathbf{X}_t$

Implement rounding on the integer decision variables of each individual $\mathbf{x}_{t,j}$ $(j \in \{1, 2, \dots, NP\})$ in $\mathbf{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,NP}\}$;
Evaluate the f value and G value of each individual in $\mathbf{X}_t$;
$FEs \leftarrow FEs + NP$;
$\mathbf{X}_{t+1} = \emptyset$;
while $(FEs \geq MaxFEs/2) \& (FEs < MaxFEs)$ **do**
  for each individual $\mathbf{x}_{t,j}$ in $\mathbf{X}_t$ **do**
    Generate a trial vector $\mathbf{u}_{t,j}$ through DE's mutation and crossover operators;
    Implement rounding on the integer decision variables of $\mathbf{u}_{t,j}$;
    Evaluate the f value and G value of $\mathbf{u}_{t,j}$;
    $FEs \leftarrow FEs + 1$;
    Select the better one between $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ according to Deb's feasibility rule and store it into $\mathbf{X}_{t+1}$;
    if $FEs \geq MaxFEs$ **then**
      break
    end if
  end for
  $t \leftarrow t + 1$;
end while

$[\mathbf{x}^*, \mathbf{y}^*]$ must be a Pareto optimal solution of the transformed CBOP.

Theorem 2: *If the original MIP problem has only one optimal solution, this solution is the only one Pareto optimal solution of the transformed CBOP which satisfies integer restrictions.*

Proof: According to Theorem 1, $[\mathbf{x}^*, \mathbf{y}^*]$, the optimal solution of the original MIP problem, is a Pareto optimal solution of the transformed CBOP. Suppose that there exists another Pareto optimal solution $[\mathbf{x}'', \mathbf{y}'']$, which also satisfies integer restrictions. It is clear that $m(\mathbf{y}^*) = m(\mathbf{y}'') = 0$. Due to the fact that $[\mathbf{x}^*, \mathbf{y}^*]$ and $[\mathbf{x}'', \mathbf{y}'']$ are nondominated with each other, it can be obtained that $f(\mathbf{x}^*, \mathbf{y}^*) = f(\mathbf{x}'', \mathbf{y}'')$. Since the original MIP problem has only one optimal solution, so $[\mathbf{x}^*, \mathbf{y}^*]$ and $[\mathbf{x}'', \mathbf{y}'']$ are the same solution. Therefore, we can conclude that $[\mathbf{x}^*, \mathbf{y}^*]$ is the only one Pareto optimal solution which satisfies integer restrictions.

According to these two theorems, the optimal solution of a MIP problem can be obtained by finding the Pareto optimal solution of the transformed CBOP which satisfies integer restrictions. The technical advantage of solving the transformed CBOP over directly solving the original MIP problem is the following: the feasible parts defined by integer restrictions have been removed, thus reducing the possibility that the population converges to a false feasible part.

2) Solving the Transformed CBOP: For each individual $\mathbf{x}_{t,j}$ $(j \in \{1, 2, \dots, NP\})$, offspring $\mathbf{u}_{t,j}$ is generated by DE. In BOToP, DE/current-to-rand/1 and DE/rand-to-best/1 in (7) and (8) are employed as the mutation operators [23], each of which is applied to $\mathbf{x}_{t,j}$ with the same probability, i.e., 0.5. Subsequently, the binomial crossover in (9) is only executed after DE/rand-to-best/1. In order to approach the Pareto optimal solution of the transformed CBOP which satisfies integer restrictions, we propose a new comparison rule to select the better one between $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$.

On the basis of the f values and m values, the nondominated sorting [55] is implemented on all the feasible solutions in the current population $\mathbf{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \dots, \mathbf{x}_{t,NP}\}$ and the offspring population $\mathbf{U}_t = \{\mathbf{u}_{t,1}, \mathbf{u}_{t,2}, \dots, \mathbf{u}_{t,NP}\}$.

Afterward, each feasible individual in $\mathbf{X}_t$ and $\mathbf{U}_t$ can obtain a ranking. We divide these feasible individuals into two categories: the feasible individuals with the best ranking (i.e., 1) belong to category I, and the other feasible individuals belong to category II. Then, $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ are compared as follows:

- If both $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ are infeasible solutions, the one with smaller degree of constraint violation is preferred.
- If only one of $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ is an infeasible solution, the feasible one is preferred.
- If both $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ are feasible solutions:
  Condition 1: If they have different nondominated rankings, select the one with better nondominated ranking.
  Condition 2: If they have the same nondominated ranking and both of them belong to category I, select the one with better m value.
  Condition 3: If they have the same nondominated ranking and both of them belong to category II, select the one with better f value.

It can be observed that if both $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ are infeasible solutions or if only one of $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ is an infeasible solution, our comparison rule is the same with Deb's feasibility rule [18]. However, if both $\mathbf{x}_{t,j}$ and $\mathbf{u}_{t,j}$ are feasible solutions, our comparison rule is different from Deb's feasibility rule. Under this condition, three conditions are considered. The reasons are explained in the following.

- *Condition 1*: According to Theorem 1, to find the optimal solution of the original MIP problem, one of the important tasks is to approach the PF of the transformed CBOP. This task can be accomplished through selecting the feasible solution with better nondominated ranking.
- *Condition 2*: The feasible solutions in category I are the feasible solutions in $\mathbf{X}_t$ and $\mathbf{U}_t$ closest to the PF of the transformed CBOP. According to Theorem 2, to find the Pareto optimal solution which satisfies integer restrictions, the feasible solution with smaller m value should be selected.
- *Condition 3*: When comparing two feasible solutions in category II, we put more emphases on the f value. It is because the comparison between two feasible solutions is based on the m value in condition 2 and we would like to make a balance between the two objective functions (i.e., f and m) of the transformed CBOP.

In rounding and truncation, the information provided by the solutions, which do not satisfy integer restrictions, will be ignored. In contrast, the solutions with good f values and/or good m values, which satisfy constraints yet do not satisfy integer restrictions, could survive into the next generation in our comparison rule, thus maintaining the diversity of the population. Moreover, these solutions can promote the exploration of the whole feasible region defined by constraints. Thus, the search ability of the population can be enhanced.

Algorithm 2 introduces the implementation of the first phase.

C. The Second Phase

Even though the population can approach the optimal solution of the original MIP problem in the first phase, it is hard to converge to the optimal solution with a high precision. The reason is that the total computational resources are mainly used to find the PS of the transformed CBOP, rather than just the optimal solution of the original MIP problem. To enhance the convergence precision, the second phase is designed, as shown in **Algorithm 3**. The second phase solves the original MIP problem directly. It utilizes the same mutation and crossover operators of DE as in the first phase to generate offspring. Subsequently, rounding is implemented to make this offspring satisfy integer restrictions. Then, the better one between a parent and its offspring is selected to the next generation according to Deb's feasibility rule [18]. By doing this, the population will finally converge to the optimal solution.

V. Experimental Study

A. Proof-of-Principle Results

$P1$ introduced in Section III-B was used to illustrate the working principle of BOToP. In BOToP, $P1$ is transformed into the following CBOP:

$$P1_{CBOP}: \begin{aligned} minimize: &\ f(x_1,x_2) = (x_1-1)^2 + (x_2-3)^2 \\ minimize: &\ m(x_1,x_2) = |x_2 - round(x_2)| \\ subject\ to: &\ (x_1+1)^2 + (x_2+1)^2 \le 1 \\ &\ x_1, x_2 \in [-3,1] \end{aligned} \quad (15)$$

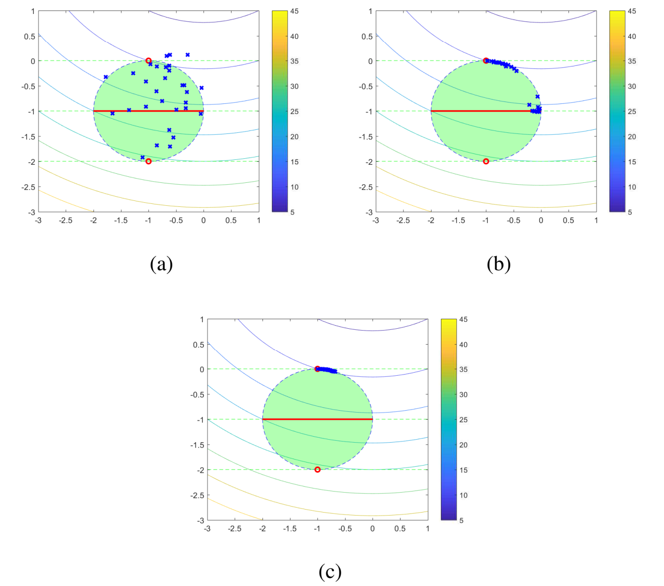


Fig. 5. A typical run derived from the first phase of BOToP on solving $P1_{CBOP}$. (a) The 10th generation. (b) The 50th generation. (c) The 200th generation.

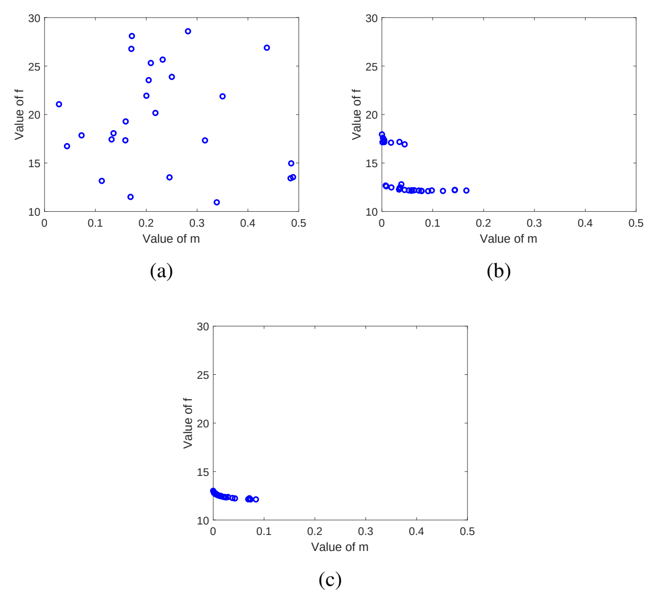
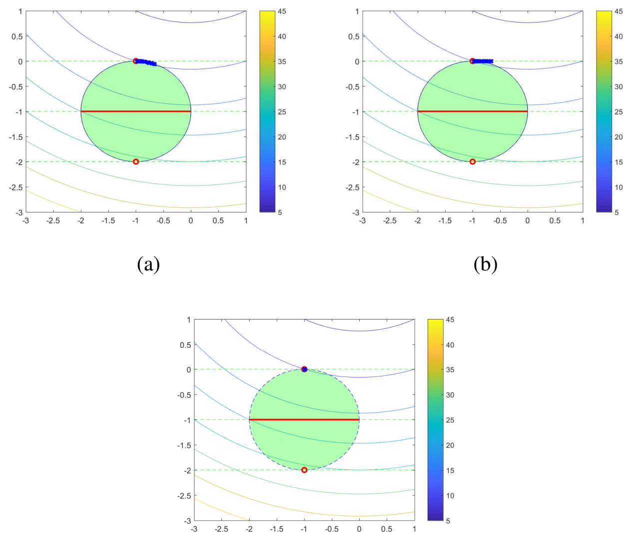


Fig. 6. A typical run derived from the first phase of BOToP in the objective space on solving $P1_{CBOP}$. (a) The 10th generation. (b) The 50th generation. (c) The 200th generation.

Fig. 5 provides a typical run derived from the first phase of BOToP on solving $P1_{CBOP}$. The parameter settings of BOToP were the same as in Section V-B. As shown in Fig. 5, the



(a) (b) (c)

Fig. 7. Implement the second phase at 201th generation. (a) The 200th generation. (b) The 201th generation. (c) The 300th generation.

solutions which do not satisfy integer restrictions can also be involved in the evolution when solving $P1_{CBOP}$. It is because $P1_{CBOP}$ removes the integer restrictions. From Fig. 5(a) and Fig. 5(b), it can be seen that the population can approach the area around feasible part I and the area on the right-hand side of feasible part II in the early stage. The reasons are twofold: 1) the solutions in these two areas are feasible solutions, and 2) the solutions in these two areas have good m values and f values. According to our comparison rule, the individuals in these two areas are very likely to be preserved to the next generation. Since the solutions around feasible part I have better f values than those around feasible part II, the population gradually concentrates on feasible part I during the evolution. This phenomenon can be observed from Fig. 5(c). As shown in Fig. 5(c), the population is close to the optimal solution.

Next, we exhibit a typical run of the first phase of BOToP in the objective space. As shown in Fig. 6(a), Fig. 6(b), and Fig. 6(c), the population is able to gradually approach the PF of $P1_{CBOP}$. Our comparison rule has the capability to guide the population toward the Pareto optimal solution which satisfies integer restrictions (i.e., the Pareto optimal solution with $m(x_1, x_2) = 0$). According to **Theorem 2**, it is the optimal solution of the original MIP.

Finally, we show the behaviour of BOToP in the second phase. To clearly observe this behaviour, we implemented the second phase at the 201th generation. From the 201th generation, rounding was implemented to make all the individuals satisfy integer restrictions. As shown in Fig. 7(b), the population only contains the solutions which satisfy integer restrictions. At last, the population can converge to the optimal solution, as shown in Fig. 7(c).

TABLE I

CHARACTERISTICS OF THE TWELVE TEST PROBLEMS, WHERE D IS THE NUMBER OF DECISION VARIABLES, *IC* IS THE NUMBER OF INEQUALITY CONSTRAINTS, *EC* IS THE NUMBER OF EQUALITY CONSTRAINTS, n_1 IS THE NUMBER OF CONTINUOUS DECISION VARIABLES, AND n_2 IS THE NUMBER OF INTEGER DECISION VARIABLES.

Problem	D	*IC*	*EC*	n_1	n_2
F1	2	1	0	1	1
F2	3	1	0	1	2
F3	2	3	0	1	1
F4	2	2	0	1	1
F5	2	0	1	1	1
F6	2	2	0	1	1
F7	5	0	3	3	2
F8	8	6	0	5	3
F9	8	6	0	5	3
F10	8	6	0	5	3
F11	15	5	0	12	3
F12	15	5	0	10	5

B. Test Problems and Parameters Settings

Twelve test problems (F1-F12) were constructed in this paper to investigate the effectiveness of BOToP. Specifically, F1-F4 have many feasible parts with different sizes and the optimal solution is located in a feasible part with a small size. F5-F7 are designed based on the test functions collected in IEEE CEC2006 [56]. Note that both F5 and F7 have equality constraints. F8-F12 are five MIP problems with at least eight decision variables. These five test problems are also designed based on [56]. The characteristics of F1-F12 are listed in Table I, where D is the number of decision variables, *IC* is the number of inequality constraints, *EC* is the number of equality constraints, n_1 is the number of continuous decision variables, and n_2 is the number of integer decision variables. The details of F1-F12 are presented in the supplementary file.

In the experimental study, the maximum number of function evaluations (FEs), denoted as $MaxFEs$, and the population size NP of BOToP were set to 1.8E+05 and 30, respectively. The first half of FEs was used in the first phase, and the rest of FEs was used in the second phase. For each test problem, 25 independent runs were performed. For equality constraints, the tolerance value δ was set to 0.0001. Inspired by [23], the scaling factor F and the crossover rate CR of DE were randomly chosen from two parameter pools: $F_{pool} = \{0.6, 0.8, 1.0\}$ and $CR_{pool} = \{0.1, 0.2, 1.0\}$. Subsequently, the mutation and crossover operators of DE were implemented according to the chosen F and CR values.

C. Is the First Phase Effective?

The unique characteristic of BOToP is its first phase. To verify the effectiveness of the first phase, BOToP was compared with another algorithm called MIPDE-FR, which has been introduced in Section III-B. To ensure a fair comparison, MIPDE-FR had the same parameter settings, mutation operator, and crossover operator with BOToP. The main differences between MIPDE-FR and BOToP are twofold: 1) in MIPDE-FR, only rounding was used to handle integer restrictions, which means that MIPDE-FR does not have the first phase; and 2) MIPDE-FR employed Deb's feasibility rule to compare individuals. Both MIPDE-FR and BOToP were applied to solve the 12 test problems constructed in this paper.

TABLE II
RESULTS OF MIPDE-FR AND BOTOP OVER 25 INDEPENDENT RUNS. *Ave* AND *Std Dev* INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE BEST FEASIBLE OBJECTIVE FUNCTION VALUES, RESPECTIVELY. *FR* AND *SR* INDICATE THE FEASIBLE RATE AND THE SUCCESSFUL RATE, RESPECTIVELY.

Problem	Status	MIPDE-FR	BOToP
F1	*FR*	**100%**	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	17.0000 ± 0.0000	**13.0000 ± 0.0000**
F2	*FR*	**100%**	**100%**
	SR	56%	**100%**
	Ave ± *Std Dev*	1.4400 ± 0.5066	**1.0000 ± 0.0000**
F3	*FR*	**100%**	**100%**
	SR	12%	**100%**
	Ave ± *Std Dev*	-3.5600 ± 0.1658	**-4.0000 ± 0.0000**
F4	*FR*	**100%**	**100%**
	SR	**100%**	**100%**
	Ave ± *Std Dev*	**-6.0000 ± 0.0000**	**-6.0000 ± 0.0000**
F5	*FR*	**100%**	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	1.2500 ± 0.0000	**0.2500 ± 0.0000**
F6	*FR*	**100%**	**100%**
	SR	92%	**100%**
	Ave ± *Std Dev*	-6699.7579 ± 290.1222	**-6783.5818 ± 0.0000**
F7	*FR*	40%	**84%**
	SR	4%	**20%**
	Ave ± *Std Dev*	NA	NA
F8	*FR*	**100%**	**100%**
	SR	8%	**80%**
	Ave ± *Std Dev*	7115.6596 ± 55.9088	**7070.3408 ± 26.8423**
F9	*FR*	**100%**	**100%**
	SR	0%	**40%**
	Ave ± *Std Dev*	7495.1141 ± 191.2120	**7209.5545 ± 179.1296**
F10	*FR*	**100%**	**100%**
	SR	16%	**64%**
	Ave ± *Std Dev*	7706.6959 ± 440.8140	**7373.3323 ± 369.6853**
F11	*FR*	**100%**	**100%**
	SR	16%	**44%**
	Ave ± *Std Dev*	36.9294 ± 5.4288	**33.7144 ± 0.3114**
F12	*FR*	**100%**	**100%**
	SR	12%	**60%**
	Ave ± *Std Dev*	88.0704 ± 38.4602	**42.1461 ± 0.6974**

TABLE III
RESULTS OF SOTOP AND BOTOP OVER 25 INDEPENDENT RUNS. *Ave* AND *Std Dev* INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE BEST FEASIBLE OBJECTIVE FUNCTION VALUES, RESPECTIVELY. *FR* AND *SR* INDICATE THE FEASIBLE RATE AND THE SUCCESSFUL RATE, RESPECTIVELY.

Problem	Status	SOToP	BOToP
F1	*FR*	**100%**	**100%**
	SR	**100%**	**100%**
	Ave ± *Std Dev*	**13.0000 ± 0.0000**	**13.0000 ± 0.0000**
F2	*FR*	0%	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	NA	**1.0000 ± 0.0000**
F3	*FR*	**100%**	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	-3.500 ± 0.0000	**-4.0000 ± 0.0000**
F4	*FR*	0%	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	NA	**-6.0000 ± 0.0000**
F5	*FR*	**100%**	**100%**
	SR	**100%**	**100%**
	Ave ± *Std Dev*	**0.2500 ± 0.0000**	**0.2500 ± 0.0000**
F6	*FR*	0%	**100%**
	SR	0%	**100%**
	Ave ± *Std Dev*	NA	**-6783.5818 ± 0.0000**
F7	*FR*	48%	**84%**
	SR	8%	**20%**
	Ave ± *Std Dev*	NA	NA
F8	*FR*	0%	**100%**
	SR	0%	**80%**
	Ave ± *Std Dev*	NA	**7070.3408 ± 26.8423**
F9	*FR*	**100%**	**100%**
	SR	0%	**40%**
	Ave ± *Std Dev*	7183.6435 ± 0.1426	7209.5545 ± 179.1296
F10	*FR*	**100%**	**100%**
	SR	0%	**64%**
	Ave ± *Std Dev*	7233.6717 ± 0.1187	7373.3323 ± 369.6853
F11	*FR*	**100%**	**100%**
	SR	0%	**44%**
	Ave ± *Std Dev*	36.5705 ± 0.1601	**33.7144 ± 0.3114**
F12	*FR*	**100%**	**100%**
	SR	0%	**60%**
	Ave ± *Std Dev*	85.6429 ± 0.9900	**42.1461 ± 0.6974**

The average and standard deviation (denoted as *Ave* and *Std Dev*) of the best feasible objective function values resulting from MIPDE-FR and BOToP over 25 independent runs are summarized in Table II. Table II also records the feasible rate (denoted as *FR*) and the successful rate (denoted as *SR*) of MIPDE-FR and BOToP. For F1-F11, a run is successful if the following condition is satisfied: $|f(\mathbf{x}_{best}) - f(\mathbf{x}^*)| \leq 0.0001$, where $\mathbf{x}^*$ is the best known solution and $\mathbf{x}_{best}$ is the best feasible solution provided by an algorithm. Due to the fact that the optimal solution of F12 is hard to find, the successful condition of F12 is revised to: $f(\mathbf{x}_{best}) \leq 41.9000$. "$NA$" represents that an algorithm cannot achieve 100% *FR* over 25 independent runs.

From Table II, it is clear that the performance of BOToP is significantly better than that of MIPDE-FR. To be specific, for ten test problems (i.e., F1-F3, F5, F6, and F8-F12), although both MIPDE and BOToP provide 100% *FR*, BOToP achieves higher *SR* than MIPDE-FR. Moreover, for five out of these ten test problems (i.e., F1-F3, F5, and F6), BOToP successfully solves them over 25 runs. However, MIPDE-FR cannot provide any successful run on F1, F5, and F9. For F7, both MIPDE-FR and BOToP fail to consistently produce feasible solutions. However, BOToP provides higher *FR* and *SR*. With respect to F4, MIPDE-FR and BOToP show similar performance. The poor performance of MIPDE-FR is largely because the optimal solutions of most of the 12 test problems are located in the feasible parts with small sizes. Thus, it is easy for MIPDE-FR to converge to a false optimal solution due to rounding. The superiority of BOToP can be attributed to the two main components of the first phase: the transformation from a MIP problem to a CBOP and the comparison rule. The former aims at alleviating the misleading of multiple feasible parts with different sizes, and the latter has the capability to exploit the information of some solutions violating integer restrictions.

In summary, the above results verify the advantage of the first phase of BOToP over rounding when solving MIP problems.

D. Is the Biobjective Optimization-based Transformation Effective?

As pointed out previously, the biobjective optimization-based transformation in the first phase of BOToP is to remove the influence of integer restrictions. Another alternative way is to directly solve the original MIP problem without considering any integer restrictions in the first phase. Under this condition, the original MIP problem is equivalent to a traditional single-objective constrained optimization problem because integer restrictions have been ignored. Based on the above idea, we designed a compared method, called SOToP. To verify the effectiveness of the biobjective optimization-based transformation, the second phase of BOToP and SOToP were the same. Table III summarizes the results derived from BOToP and SOToP in terms of *Ave*, *Std Dev*, *SR*, and *FR*. Again, "NA" denotes that an algorithm cannot achieve 100% *FR* over 25 independent runs.

As shown in Table III, BOToP outperforms SOToP on all the 12 test problems except F1 and F5. SOToP fails to find any feasible solution on F2, F4, F6, and F8, and cannot provide any successful run on F2-F4, F6, and F8-F12 over 25 independent runs. The poor performance of SOToP is not difficult to understand since we cannot theoretically analyze the relationship between the original MIP problem and the transformed single-objective constrained optimization problem. Therefore, the transformation of SOToP cannot provide promising solutions for the second phase. This conclusion has also been mentioned in [57]. Different from SOToP, the transformation of BOToP can guarantee that the Pareto optimal solution which satisfies integer restrictions is the optimal solution of the original MIP problem, thus providing a reasonable guideline toward the optimal solution.

E. Is the Comparison Rule Effective?

As introduced in Section IV-B, in our comparison rule, two feasible individuals are compared according to three different conditions. To identify the benefit of these three conditions, three variants of BOToP were devised: BOToP-C1, BOToP-C1-C3 and BOToP-C1-C2. In BOToP-C1, only condition 1 was employed. In BOToP-C1-C3 and BOToP-C1-C2, condition 2 and condition 3 were eliminated, respectively. *Ave*, *Std Dev*, *FR*, and *SR* derived from BOToP, BOToP-C1, BOToP-C1-C2, and BOToP-C1-C3 are presented in Table S-I of the supplementary file. When an algorithm cannot achieve 100% *FR* over 25 independent runs, its *Ave* and *Std Dev* are denoted as "NA".

As shown in Table S-I, the three BOToP variants provide similar results with BOToP on F2, F4, F6, and F7. However, when solving the other test problems, BOToP is more competitive except that BOToP shows similar performance with BOToP-C1-C2 and BOToP-C1 on F1 and F12, respectively. To test the statistical significance, the Wilcoxon's rank-sum test at a 0.05 significance level was implemented between BOToP and each of its three variants. In Table S-I, "$-$", "$+$", and "$\approx$" denote that BOToP performs better than, worse than, and similar to its variant, respectively. From Table S-I, it can be seen that BOToP surpasses BOToP-C1, BOToP-C1-C3, and BOToP-C1-C2 on seven, seven, and four test problems, respectively. However, the three variants cannot beat BOToP on any test problem. The above phenomenon can be explained as follows. Condition 2 and condition 3 aim to approach the optimal solution and strike a balance between the two objective functions of the transformed CBOP, respectively. Therefore, without condition 2 or condition 3, the performance of BOToP degrades significantly.

From the above discussion, we can conclude that all of the three conditions are necessary.

F. Comparison with Five State-of-the-Art EAs for Solving MIP Problems

In this subsection, we compared BOToP with five other state-of-the-art EAs: MDE [1], MDE-LS [41], MDE-IHS [41], EMDE [40], and DE_{MV} [58]. MDE is a modified version of the traditional DE to solve chemical process synthesis and design problems. MDE-LS is a hybrid algorithm which incorporates a local search operator to enhance the exploitation ability. MDE-IHS is a hybrid algorithm which adds a second metaheuristic called harmony search. In EMDE, a novel triangular mutation operator is designed to enhance the search ability. DE_{MV} is a recently proposed hybrid DE algorithm, which introduces set-based DE operators to deal with integer and categorical variables. Same with BOToP, NP and $MaxFEs$ of the five compared algorithms were set to 30 and 1.8E+05, respectively. For fairness, in our experiments, the DE operators used in MDE, MDE-LS, MDE-IHS, and DE_{MV} were the same with BOToP. Since one contribution of EMDE is the triangular mutation operator, EMDE was implemented without any modification.

Ave, *Std Dev*, *FR*, and *SR* provided by BOToP and the five compared algorithms are summarized in Table S-II of the supplementary file. "NA" denotes that an algorithm cannot achieve 100% *FR* over 25 runs. From Table S-II, BOToP is better than MDE, MDE-LS, and MDE-IHS on all test problems except F2 and F4. For F2 and F4, BOToP, MDE, MDE-LS, and MDE-IHS show similar performance. BOToP provides similar results with EMDE on F4 and F6, and better results than EMDE on the remaining test problems. In addition, BOToP shows better performance than DE_{MV} on all test problems expect F6. To detect the statistical difference, the Wilcoxon's rank-sum test at a 0.05 significance level was implemented between BOToP and each of MDE, MDE-LS, MDE-IHS, EMDE, and DE_{MV} in Table S-II, where "$-$", "$+$", and "$\approx$" denote that BOToP performs better than, worse than, and similar to the competitor, respectively. It can be seen that BOToP has an edge over MDE, MDE-LS, MDE-IHS, EMDE, and DE_{MV} on eight, eight, eight, nine, and nine test problems, respectively. However, MDE, MDE-LS, MDE-IHS, EMDE, and DE_{MV} cannot surpass BOToP on any test problem. The above comparison demonstrates that, overall, BOToP is better than the five competitors on solving the 12 test problems.

VI. Conclusion

In MIP, integer restrictions divide the feasible region defined by constraints into several discontinuous feasible parts. When employing rounding or truncation to handle integer restrictions in MIP, the population runs the risk of converging to a local optimal solution due to the fact that the evolution is remarkably influenced by the feasible parts with different sizes. To overcome this disadvantage, a biobjective optimization-based two-phase method (called BOToP) was proposed. In the first phase, a MIP problem was transformed into a CBOP with no integer restrictions. We had proven that the Pareto optimal solution of the transformed CBOP which satisfies integer restrictions is the optimal solution of the original MIP, thus the optimal solution can be approached by solving the transformed CBOP. DE was combined with a new comparison rule to accomplish this task. Afterward, the second phase, which employed rounding to handle integer restrictions, was implemented to enhance the convergence precision and obtain the final solution. From the comparative studies on the 12 test problems designed in this paper, the effectiveness of some components in BOToP (i.e., the first phase, the biobjective optimization-based transformation, and the comparison rule) was verified. Moreover, the results also showed that the performance of BOToP is better than that of five other state-of-the-art EAs.

However, BOToP also has some limitations. Firstly, it is worth noting that BOToP requires a specification of the maximum number of FEs beforehand to the execution. Secondly, in BOToP, the solutions which do not satisfy integer restrictions are also evaluated by the objective function and constraints. However, in some applications, only the solution satisfying integer restrictions can be evaluated by the objective function and constraints. The following is an example:

$$minimize : z = \begin{cases} cos(6.8\pi x_1/2), & \text{if } x_2 = 0 \\ -cos(7\pi x_1/2), & \text{if } x_2 = 1 \end{cases} \tag{16}$$

where $x_1 \in [0, 1]$. In this example, the objective function value of a solution can be computed only when $x_2 = 0$ or $x_2 = 1$. In the future, we will try to solve this kind of MIP problems.

References

[1] R. Angira and B. Babu, "Optimization of process synthesis and design problems: A modified differential evolution approach," *Chemical Engineering Science*, vol. 61, no. 14, pp. 4707–4721, 2006.

[2] R. Balamurugan and S. Subramanian, "Hybrid integer coded differential evolution dynamic programming approach for economic load dispatch with multiple fuel options," *Energy Conversion and Management*, vol. 49, no. 4, pp. 608–614, 2008.

[3] H. Wu, C. Nie, F. C. Kuo, H. Leung, and C. J. Colbourn, "A discrete particle swarm optimization for covering array generation," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 575–591, 2015.

[4] M. W. Cooper, "A survey of methods for pure nonlinear integer programming," *Management Science*, vol. 27, no. 3, pp. 353–361, 1981.

[5] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, "Cutting planes in integer and mixed integer programming," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 397–446, 1999.

[6] C. A. Floudas, "Nonlinear and mixed-integer optimization: Fundamentals and applications," *Oxford University Press*, 1995.

[7] I. E. Grossmann, "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and Engineering*, vol. 3, no. 3, pp. 227–252, 2002.

[8] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y. Ong, "A new decomposition-based NSGA-II for many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1191–1210, July 2018.

[9] Y. R. Naidu and A. K. Ojha, "Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 821–832, June 2018.

[10] D. Li, C. Zhang, G. Tian, X. Shao, and Z. Li, "Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 119–129, Jan 2018.

[11] A. Song, W. Chen, T. Gu, H. Yuan, S. Kwong, and J. Zhang, "Distributed virtual network embedding system with historical archives and set-based particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, DOI: 10.1109/TSMC.2018.2881018.

[12] Z. Liao, W. Gong, X. Yan, L. Wang, and C. Hu, "Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1590–1601, 2020.

[13] W. Gong, Y. Wang, Z. Cai, and L. Wang, "Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution," *IEEE Transactions on Systems Man and Cybernetics: Systems*, vol. 50, no. 4, pp. 1642–1656, 2020.

[14] Y. Jia, W. Chen, H. Yuan, T. Gu, H. Zhang, Y. Gao, and J. Zhang, "An intelligent cloud workflow scheduling system with time estimation and adaptive ant colony optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, DOI: 10.1109/TSMC.2018.2881018.

[15] B.-C. Wang, H.-X. Li, J. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1482–1495, 2019.

[16] D. W. Coit, A. E. Smith, and D. M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *Informs Journal on Computing*, vol. 8, no. 2, pp. 173–182, 1996.

[17] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 445–455, 2003.

[18] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[19] T. Takahama and S. Sakai, "Constrained optimization by applying the α-constrained method to the nonlinear simplex method with mutations," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 437–451, 2005.

[20] ——, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2006.

[21] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[22] E. Mezura-Montes and C. A. Coello Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, 2005.

[23] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2015.

[24] C. A. Coello Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civil Engineering and Environmental Systems*, vol. 17, no. 4, pp. 319–346, 2000.

[25] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 658–675, 2006.

[26] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 560–575, 2007.

[27] B.-C. Wang, H.-X. Li, Q. Zhang, and Y. Wang, "Decomposition-based multiobjective optimization for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, DOI: 10.1109/TSMC.2018.2876335.

[28] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[29] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "A dynamic hybrid framework for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 203–217, 2012.

[30] V. Ho-Huu, T. Nguyen-Thoi, M. Nguyen-Thoi, and L. Le-Anh, "An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7057–7069, 2015.

[31] S. Jing and J. Yang, "A novel set-based discrete differential evolution algorithm for mining process model from event Log," in *International Conference on Computational Intelligence and Security (CIS)*, 2019, pp. 161–165.

[32] S. Y. Zheng, B. J. Xiang, X. Y. Zhang, and J. Zhang, "Differential evolution optimization algorithm for antenna designs with mixed discrete-continuous variables," in *International Conference on Microwave and Millimeter Wave Technology (ICMMT)*, 2019, pp. 1–3.

[33] F. Zhao, L. Zhao, L. Wang, and H. Song, "An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion," *Expert Systems with Applications*, vol. 160, pp. 113–678, 2020.

[34] K. Deep, K. P. Singh, M. Kansal, and C. Mohan, "A real coded genetic algorithm for solving integer and mixed integer optimization problems," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 505–518, 2009.

[35] D. Datta and J. R. Figueira, "A real-integer-discrete-coded differential evolution," *Applied Soft Computing*, vol. 13, no. 9, pp. 3884–3893, 2013.

[36] R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[37] K. Deep, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211–230, 2007.
[38] J. Lampinen and I. Zelinka, "Mixed variable non-linear optimization by differential evolution," *Proceedings of Nostradamus*, vol. 92, no. 2, pp. 7–8, 1999.
[39] H. Li and L. Zhang, "A discrete hybrid differential evolution algorithm for solving integer programming problems," *Engineering Optimization*, vol. 46, no. 9, pp. 1238–1268, 2014.
[40] A. W. Mohamed, "An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 1–19, 2015.
[41] T. Liao, "Two hybrid differential evolution algorithms for engineering design optimization," *Applied Soft Computing*, vol. 10, no. 4, pp. 1188–1199, 2010.
[42] Y. Luo, X. Yuan, and Y. Liu, "An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints," *Computers & Chemical Engineering*, vol. 31, no. 3, pp. 153–162, 2007.
[43] C. Mohan and H. T. Nguyen, "A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems," *Computational Optimization and Applications*, vol. 14, no. 1, pp. 103–132, 1999.
[44] Y. C. Lin, K. S. Hwang, and F. S. Wang, "Co-evolutionary hybrid differential evolution for mixed-integer optimization problems," *Engineering Optimization*, vol. 33, no. 6, pp. 663–682, 2001.
[45] J. P. Chiou and F. S. Wang, "A hybrid method of differential evolution with application to optimal control problems of a bioprocess system," *1998 IEEE International Conference on Evolutionary Computation Proceedings*, pp. 627–632, 1998.
[46] J. Wu, Y. Gao, and L. Yan, "An improved differential evolution algorithm for mixed integer programming problems," *2013 Ninth International Conference on Computational Intelligence and Security*, pp. 31–35, 2013.
[47] A. Ponsich and C. A. Coello Coello, "Differential evolution performances for the solution of mixed-integer constrained process engineering problems," *Applied Soft Computing*, vol. 11, no. 1, pp. 399–409, 2011.
[48] L. Sahoo, A. Banerjee, and A. K. Bhunia, "An efficient GA–PSO approach for solving mixed-integer nonlinear programming problem in reliability optimization," *Swarm and Evolutionary Computation*, vol. 19, pp. 43–51, 2014.
[49] V. H. Hinojosa and R. Araya, "Modeling a mixed-integer-binary small-population evolutionary particle swarm algorithm for solving the optimal power flow problem in electric power systems," *Applied Soft Computing*, vol. 13, no. 9, pp. 3839–3852, 2013.
[50] G. Jia, Y. Wang, Z. Cai, and Y. Jin, "An improved (μ+λ)-constrained differential evolution for constrained optimization," *Information Sciences*, vol. 222, no. 4, pp. 302–322, 2013.
[51] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 4104–4108, 1997.
[52] D. Datta and J. R. Figueira, "A real-integer-discrete-coded particle swarm optimization for design problems," *Applied Soft Computing*, vol. 11, no. 4, pp. 3625–3633, 2011.
[53] R. Li, M. T. Emmerich, J. Eggermont, T. Back, M. Schutz, J. Dijkstra, and J. Reiber, "Mixed integer evolution strategies for parameter optimization," *Evolutionary Computation*, vol. 21, no. 1, pp. 29–64, 2013.
[54] W. C. Wu and M. S. Tsai, "Application of enhanced integer coded particle swarm optimization for distribution system feeder reconfiguration," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1591–1599, 2011.
[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[56] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, no. 8, pp. 8–31, 2006.
[57] G. M. Ostrovsky and G. W. Mikhailow, "Discrete optimization of chemical processes," *Computers & Chemical Engineering*, vol. 14, no. 1, pp. 111–117, 1990.
[58] Y. Lin, Y. Liu, W. Chen, and J. Zhang, "A hybrid differential evolution algorithm for mixed-variable optimization problems," *Information Sciences*, vol. 466, pp. 170–188, 2018.

Jiao Liu received the B.S. degree in process equipment and control engineering and the M.S. degree in power engineering and engineering thermophysics both from the Taiyuan University of Technology, Taiyuan, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in control science and engineering, Central South University, Changsha, China.

His current research interests include evolutionary computation, mixed-integer programming, and automotive lightweight design.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang is an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *Swarm and Evolutionary Computation*. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Bin Xin (Member, IEEE) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering, both from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He was an academic visitor at the Decision and Cognitive Sciences Research Centre, the University of Manchester, from 2011 to 2012.

He is currently a Professor with the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, evolutionary computation, unmanned systems, and multiagent systems. He serves as an Associate Editor of the *Unmanned Systems* and many other international journals.

Ling Wang received the B.Sc. in automation and Ph.D. in control theory and control engineering from Tsinghua University, Beijing, China, in 1995 and 1999, respectively. Since 1999, he has been with the Department of Automation, Tsinghua University, where he became a full professor in 2008.

His current research interests include intelligent optimization and production scheduling. He has authored five academic books and more than 300 refereed papers. He was a recipient of the National Natural Science Fund for Distinguished Young Scholars of China, the National Natural Science Award (Second Place) in 2014, the Science and Technology Award of Beijing City in 2008, and the Natural Science Award (First Place in 2003, and Second Place in 2007) nominated by the Ministry of Education of China. Professor Wang now is the Editor-in-Chief of *International Journal of Automation and Control*, and an Associate Editor of the *IEEE Transactions on Evolutionary Computation*, the *Swarm and Evolutionary Computation*, the *Expert Systems with Applications*, etc.

Supplementary File for "A Biobjective Perspective for Mixed-Integer Programming"

S-I. RESULTS

TABLE S-I
RESULTS OF BOTOP, BOTOP-C1, BOTOP-C1-C3, AND BOTOP-C1-C2 OVER 25 INDEPENDENT RUNS. *Ave* AND *Std Dev* INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE BEST FEASIBLE OBJECTIVE FUNCTION VALUES, RESPECTIVELY. *FR* AND *SR* INDICATE THE FEASIBLE RATE AND THE SUCCESSFUL RATE, RESPECTIVELY. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED BETWEEN BOTOP AND EACH OF BOTOP-C1, BOTOP-C1-C3, AND BOTOP-C1-C2.

Problem	Status	BOToP-C1	BOToP-C1-C3	BOToP-C1-C2	BOToP
F1	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	24%	36%	**100%**	**100%**
	Ave ± *Std Dev*	$16.0400 \pm 1.6330-$	$15.5600 \pm 1.9596-$	**13.0000 ± 0.0000** $\approx$	**13.0000 ± 0.0000**
F2	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**1.0000 ± 0.0000** $\approx$	**1.0000 ± 0.0000** $\approx$	**1.0000 ± 0.0000** $\approx$	**1.0000 ± 0.0000**
F3	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	44%	52%	88%	**100%**
	Ave ± *Std Dev*	$-3.7200 \pm 0.2533-$	$-3.7600 \pm 0.2550-$	$-3.9400 \pm 0.1685 \approx$	**-4.0000 ± 0.0000**
F4	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000**
F5	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	80%	60%	88%	**100%**
	Ave ± *Std Dev*	$0.5668 \pm 0.4714-$	$0.6460 \pm 0.4951-$	$0.3688 \pm 0.3284 \approx$	**0.2500 ± 0.0000**
F6	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**-6783.5818 ± 0.0000** $\approx$	**-6783.5818 ± 0.0000** $\approx$	**-6783.5818 ± 0.0000** $\approx$	**-6783.5818 ± 0.0000**
F7	*FR*	**84%**	76%	80%	**84%**
	SR	**24%**	16%	**24%**	20%
	Ave ± *Std Dev*	$NA \approx$	$NA \approx$	$NA \approx$	NA
F8	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	40%	68%	0%	**80%**
	Ave ± *Std Dev*	$7112.2720 \pm 82.0650-$	$7073.5611 \pm 26.5236 \approx$	$7112.3704 \pm 40.7376-$	**7070.3408 ± 26.8423**
F9	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	4%	20%	0%	**40%**
	Ave ± *Std Dev*	$7372.9783 \pm 216.8699-$	$7336.3361 \pm 191.7481-$	$7417.3329 \pm 217.9000-$	**7209.5545 ± 179.1296**
F10	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	12%	52%	12%	**64%**
	Ave ± *Std Dev*	$7683.0022 \pm 344.7027-$	$7629.3333 \pm 538.3727-$	$7847.9998 \pm 514.6091-$	**7373.3323 ± 369.6853**
F11	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	32%	40%	24%	**44%**
	Ave ± *Std Dev*	$33.9406 \pm 0.7684-$	$34.4502 \pm 1.3163-$	$33.8014 \pm 0.6402-$	**33.7144 ± 0.3114**
F12	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**60%**	12%	24%	**60%**
	Ave ± *Std Dev*	$42.2539 \pm 1.5335 \approx$	$53.1226 \pm 17.3967-$	$42.3356 \pm 1.5276 \approx$	**42.1461 ± 0.6974**
$-/+/\approx$		7/0/5	7/0/5	4/0/8	/

TABLE S-II

RESULTS OF MDE, MDE-LS, MDE-IHS, EMDE, DE_{MV}, AND BOTOP OVER 25 INDEPENDENT RUNS. *Ave* AND *Std Dev* INDICATE THE AVERAGE AND STANDARD DEVIATION OF THE BEST FEASIBLE OBJECTIVE FUNCTION VALUES, RESPECTIVELY. *FR* AND *SR* INDICATE THE FEASIBLE RATE AND THE SUCCESSFUL RATE, RESPECTIVELY. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED BETWEEN BOTOP AND EACH OF MDE, MDE-LS, MDE-IHS, EMDE, AND DE_{MV}.

Problem	Status	MDE	MDE-LS	MDE-IHS	EMDE	DE_{MV}	BOToP
F1	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	0%	0%	0%	0%	64%	**100%**
	Ave $\pm$ *Std Dev*	$17.0000 \pm 0.0000-$	$17.0000 \pm 0.0000-$	$17.0000 \pm 0.0000-$	$17.0000 \pm 0.0000-$	$14.4400 \pm 1.9596-$	**13.0000 ± 0.0000**
F2	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	0%	16%	**100%**
	Ave $\pm$ *Std Dev*	**1.0000 ± 0.0000** $\approx$	**1.0000 ± 0.0000** $\approx$	**1.0000 ± 0.0000** $\approx$	$2.0000 \pm 0.0000-$	$1.9200 \pm 0.5715-$	**1.0000 ± 0.0000**
F3	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	16%	32%	60%	0%	88%	**100%**
	Ave $\pm$ *Std Dev*	$-3.5800 \pm 0.1871-$	$-3.6600 \pm 0.2380-$	$-3.8000 \pm 0.2499-$	$-3.5000 \pm 0.0000-$	$-3.9400 \pm 0.1658 \approx$	**-4.0000 ± 0.0000**
F4	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**	48%	**100%**
	Ave $\pm$ *Std Dev*	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000** $\approx$	**-6.0000 ± 0.0000** $\approx$	$-4.7200 \pm 1.7205-$	**-6.0000 ± 0.0000**
F5	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	4%	4%	24%	0%	0%	**100%**
	Ave $\pm$ *Std Dev*	$1.2005 \pm 0.1980-$	$1.2005 \pm 0.1980-$	$1.0025 \pm 0.4316-$	$1.2500 \pm 0.0000-$	$1.2500 \pm 0.0000-$	**0.2500 ± 0.0000**
F6	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	92%	92%	92%	**100%**	**100%**	**100%**
	Ave $\pm$ *Std Dev*	$-6699.7579 \pm 290.1222 \approx$	$-6699.7579 \pm 290.1222 \approx$	$-6699.3903 \pm 290.0129 \approx$	**-6783.5818 ± 0.0000** $\approx$	**-6783.5818 ± 0.0000** $\approx$	**-6783.5818 ± 0.0000**
F7	*FR*	32%	20%	**84%**	72%	12%	**84%**
	SR	0%	0%	8%	0%	0%	**20%**
	Ave $\pm$ *Std Dev*	$NA \approx$	$NA \approx$	$NA \approx$	$NA \approx$	$NA \approx$	$NA \approx$
F8	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	12%	12%	0%	0%	0%	**80%**
	Ave $\pm$ *Std Dev*	$7123.4498 \pm 55.4274-$	$7118.5641 \pm 46.0454-$	$7393.4986 \pm 376.7446-$	$7512.6569 \pm 317.7695-$	$11181.4292 \pm 2281.8714-$	**7070.3408 ± 26.8423**
F9	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	4%	0%	0%	0%	0%	**40%**
	Ave $\pm$ *Std Dev*	$7550.2372 \pm 140.8965-$	$7375.6312 \pm 217.3625-$	$8062.9040 \pm 741.0874-$	$7810.6665 \pm 437.0184-$	$11901.9253 \pm 3186.3343-$	**7209.5545 ± 179.1296**
F10	*FR*	**100%**	**100%**	**100%**	**100%**	72%	**100%**
	SR	12%	8%	0%	32%	0%	**64%**
	Ave $\pm$ *Std Dev*	$7883.9998 \pm 491.0423-$	$7654.6665 \pm 156.9033-$	$8380.9085 \pm 1760.7817-$	$7998.6661 \pm 788.4285-$	$NA-$	**7373.3323 ± 369.6853**
F11	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	0%	0%	0%	16%	0%	**44%**
	Ave $\pm$ *Std Dev*	$38.9841 \pm 6.3478-$	$50.3621 \pm 8.5924-$	$98.7962 \pm 22.4980-$	$39.3675 \pm 7.0430-$	$78.4360 \pm 13.9930-$	**33.7144 ± 0.3114**
F12	*FR*	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
	SR	4%	0%	0%	0%	0%	**60%**
	Ave $\pm$ *Std Dev*	$85.8413 \pm 22.1436-$	$94.9773 \pm 15.3228-$	$142.6916 \pm 73.6463-$	$98.4292 \pm 47.8409-$	$65.9677 \pm 6.7737-$	**42.1461 ± 0.6974**
$-/+/\approx$		8/0/4	8/0/4	8/0/4	9/0/3	9/0/3	/

S-II. Test Problems F1-F12

$$\textbf{F1}: \ minimize: \ f(x,y) = (x-1)^2 + (y-3)^2$$
$$subject\ to:$$
$$g(x,y) = (x+1)^2 + (y+1)^2 - 1 \leq 0$$
$$x \in [-3, 1]$$
$$y \in \{-3, -2, -1, 0, 1\}$$

The optimal solution is $x^* = -1$ and $y^* = 0$, and $f(x^*, y^*) = 13.0000$.

$$\textbf{F2}: \ minimize: \ f(x,\mathbf{y}) = x^2 + (y_1-1)^2 + (y_2-2)^2$$
$$subject\ to:$$
$$g(x,\mathbf{y}) = x^2 + y_1^2 + 0.5y_2^2 - 1.5 \leq 0$$
$$x \in [-1, 100]$$
$$y_1, y_2 \in \{-1, 0, \ldots, 100\}$$

The optimal solution is $x^* = 0$ and $\mathbf{y}^* = (1, 1)$, and $f(x^*, \mathbf{y}^*) = 1.0000$.

$$\textbf{F3}: \ minimize: \ f(x,y) = -x - y$$
$$subject\ to:$$
$$g_1(x,y) = -x + y - 2.005 \leq 0$$
$$g_2(x,y) = x - y + 0.5 \leq 0$$
$$g_3(x,y) = 0.505x + y - 3.505 \leq 0$$
$$x \in [-1, 100]$$
$$y \in \{-1, 0, \ldots, 100\}$$

The optimal solution is $x^* = 1$ and $y^* = 3$, and $f(x^*, y^*) = -4.0000$.

$$\textbf{F4}: \ minimize: \ f(x,y) = -x - y$$
$$subject\ to:$$
$$g_1(x,y) = y - 3.4 \leq 0$$
$$g_2(x,y) = x - y \leq 0$$
$$x \in [-1, 100]$$
$$y \in \{-1, 0, \ldots, 100\}$$

The optimal solution is $x^* = 3$ and $y^* = 3$, and $f(x^*, y^*) = -6.0000$.

$$\textbf{F5}: \ minimize: \ f(x,y) = (x-0.5)^2 + (y-1)^2$$
$$subject\ to:$$
$$h(x,y) = -x^2 + y = 0$$
$$x \in [-1, 3.1]$$
$$y \in \{-1, 0, \ldots, 4\}$$

The best known optimal solution is $x^* = 1$ and $y^* = 1$, and $f(x^*, y^*) = 0.2500$.

$$\textbf{F6}: \ minimize: \ f(x,y) = (x-10)^3 + (y-20)^3$$
$$subject\ to:$$
$$g_1(x,y) = -(x-5)^2 - (y-4.86)^2 + 100 \leq 0$$
$$g_2(x,y) = (x-8)^2 + (y-5.48)^2 - 60 \leq 0$$
$$x \in [-1, 100]$$
$$y \in \{-1, 0, \ldots, 100\}$$

The best known optimal solution is $x^* = 14.22498780$ and $y^* = 1$, and $f(x^*, y^*) = -6783.5818$.

$$
\begin{aligned}
\textbf{F7}: \ & minimize: \ f(\mathbf{x},\mathbf{y}) = exp(x_1x_2x_3y_1y_2) \\
& subject\ to: \\
& \qquad h_1(\mathbf{x},\mathbf{y}) = x_1^2 + x_2^2 + x_3^2 + y_1^2 + y_2^2 - 10 = 0 \\
& \qquad h_2(\mathbf{x},\mathbf{y}) = x_2y_1 - 5x_3y_2 = 0 \\
& \qquad h_3(\mathbf{x},\mathbf{y}) = x_1^3 + y_1^3 + 1 = 0 \\
& \qquad x_1 \in [-2.3, 2.3] \\
& \qquad x_2, x_3 \in [-3.2, 3.2] \\
& \qquad y_1 \in \{-2, -1, 0, 1, 2\} \\
& \qquad y_2 \in \{-3, -2, -1, 0, 1, 2, 3\}
\end{aligned}
$$

The best known optimal solution is $\mathbf{x}^* = (-1.25994205, -2.48314049, 0.496648098)$ and $\mathbf{y}^* = (1, -1)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 0.2114$.

$$
\begin{aligned}
\textbf{F8}: \ & minimize: f(\mathbf{x},\mathbf{y}) = x_1 + x_2 + y_1 \\
& subject\ to: \\
& \qquad g_1(\mathbf{x},\mathbf{y}) = -1 + 0.0025(x_3 + x_4) \leq 0 \\
& \qquad g_2(\mathbf{x},\mathbf{y}) = -1 + 0.0025(-x_3 + y_2 + y_3) \leq 0 \\
& \qquad g_3(\mathbf{x},\mathbf{y}) = -1 + 0.01(x_5 - y_2) \leq 0 \\
& \qquad g_4(\mathbf{x},\mathbf{y}) = -x_1x_4 + 833.33252x_3 + 100x_1 - 83333.333 \leq 0 \\
& \qquad g_5(\mathbf{x},\mathbf{y}) = -y_1y_3 + 1250y_2 + x_3y_1 - 1250x_3 \leq 0 \\
& \qquad g_6(\mathbf{x},\mathbf{y}) = -x_2x_5 + 1250000 + x_2y_2 - 2500y_2 \leq 0 \\
& \qquad x_1 \in [100, 10000] \\
& \qquad x_2 \in [1000, 10000] \\
& \qquad x_3, x_4, x_5 \in [10, 1000] \\
& \qquad y_1 \in \{1000, 1020, \ldots, 10000\} \\
& \qquad y_2, y_3 \in \{20, 40, \ldots, 1000\}
\end{aligned}
$$

The best known optimal solution is $\mathbf{x}^* = (555.55433833, 5000, 180, 220, 400)$ and $\mathbf{y}^* = (1500, 300, 280)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 7055.5544$.

$$
\begin{aligned}
\textbf{F9}: \ & minimize: f(\mathbf{x},\mathbf{y}) = x_1 + x_2 + y_1 \\
& subject\ to: \\
& \qquad g_1(\mathbf{x},\mathbf{y}) = -1 + 0.0025(x_3 + x_4) \leq 0 \\
& \qquad g_2(\mathbf{x},\mathbf{y}) = -1 + 0.0025(-x_3 + y_2 + y_3) \leq 0 \\
& \qquad g_3(\mathbf{x},\mathbf{y}) = -1 + 0.01(x_5 - y_2) \leq 0 \\
& \qquad g_4(\mathbf{x},\mathbf{y}) = -x_1x_4 + 833.33252x_3 + 100x_1 - 83333.333 \leq 0 \\
& \qquad g_5(\mathbf{x},\mathbf{y}) = -y_1y_3 + 1250y_2 + x_3y_1 - 1250x_3 \leq 0 \\
& \qquad g_6(\mathbf{x},\mathbf{y}) = -x_2x_5 + 1250000 + x_2y_2 - 2500y_2 \leq 0 \\
& \qquad x_1 \in [100, 10000] \\
& \qquad x_2 \in [1000, 10000] \\
& \qquad x_3, x_4, x_5 \in [10, 1000] \\
& \qquad y_1 \in \{1000, 1050, \ldots, 10000\} \\
& \qquad y_2, y_3 \in \{50, 100, \ldots, 1000\}
\end{aligned}
$$

The best known optimal solution is $\mathbf{x}^* = (833.33171, 5000, 200, 200, 400)$ and $\mathbf{y}^* = (1250, 300, 300)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 7083.3317$.

$$
\begin{aligned}
\textbf{F10}: \ & \mathit{minimize}: f(\mathbf{x},\mathbf{y}) = x_1 + x_2 + y_1 \\
& \mathit{subject\ to}: \\
& \qquad g_1(\mathbf{x},\mathbf{y}) = -1 + 0.0025(x_3 + x_4) \le 0 \\
& \qquad g_2(\mathbf{x},\mathbf{y}) = -1 + 0.0025(-x_3 + y_2 + y_3) \le 0 \\
& \qquad g_3(\mathbf{x},\mathbf{y}) = -1 + 0.01(x_5 - y_2) \le 0 \\
& \qquad g_4(\mathbf{x},\mathbf{y}) = -x_1 x_4 + 833.33252 x_3 + 100 x_1 - 83333.333 \le 0 \\
& \qquad g_5(\mathbf{x},\mathbf{y}) = -y_1 y_3 + 1250 y_2 + x_3 y_1 - 1250 x_3 \le 0 \\
& \qquad g_6(\mathbf{x},\mathbf{y}) = -x_2 x_5 + 1250000 + x_2 y_2 - 2500 y_2 \le 0 \\
& \qquad x_1 \in [100, 10000] \\
& \qquad x_2 \in [1000, 10000] \\
& \qquad x_3, x_4, x_5 \in [10, 1000] \\
& \qquad y_1 \in \{1000, 1100, \ldots, 10000\} \\
& \qquad y_2, y_3 \in \{100, 200, \ldots, 1000\}
\end{aligned}
$$

The best known optimal solution is $\mathbf{x}^* = (833.33171, 5000, 200, 200, 400)$ and $\mathbf{y}^* = (1300, 300, 300)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 7133.3317$.

$$
\begin{aligned}
\textbf{F11}: \ & \mathit{minimize}: \ f(\mathbf{x}) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij} x_{(10+i)} x_{(10+j)} + 2\sum_{j=1}^{5} d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i \\
& \mathit{subject\ to}: \\
& \qquad g_j(\mathbf{x}) = -2\sum_{i=1}^{5} c_{ij} x_{(10+i)} - 3 d_j x_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij} x_i \le 0, j = 1, \ldots, 5 \\
& \qquad x_1, x_2, x_4, x_6, \ldots, x_{11}, x_{13}, x_{14}, x_{15} \in [0, 10] \\
& \qquad x_3, x_5, x_{12} \in \{0, 1, \ldots, 10\}
\end{aligned}
$$

where $\mathbf{b} = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the remaining data is in Table S-III. The best known optimal solution is $\mathbf{x}^* = (x_1, x_2, x_4, x_6, \ldots, x_{11}, x_{13}, x_{14}, x_{15}) = (0, 0, 0, 9.99999985, 0, 0, 0, 0, 0.28879805, 0.43951302, 0.31935496, 0.44885950)$

TABLE S-III
DATASET FOR TEST PROBLEMS F11 AND F12

j	1	2	3	4	5
e_j	-15	-27	-36	-18	-12
c_{1j}	30	-20	-10	32	-10
c_{2j}	-20	39	-6	-31	32
c_{3j}	-10	-6	10	-6	-10
c_{4j}	32	-31	-6	39	-20
c_{5j}	-10	32	-10	-20	30
d_j	4	8	10	6	2
a_{1j}	-16	2	0	1	0
a_{2j}	0	-2	0	0.4	2
a_{3j}	-3.5	0	2	0	0
a_{4j}	0	-2	0	-4	-1
a_{5j}	0	-9	-2	1	-2.8
a_{6j}	2	0	-4	0	0
a_{7j}	-1	-1	-1	-1	-1
a_{8j}	-1	-2	-3	-2	-1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

and $\mathbf{y}^* = (x_3, x_5, x_{12}) = (4, 4, 0)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 33.5066$.

$$\textbf{F12}: \ minimize: \ f(\mathbf{x}) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{(10+i)}x_{(10+j)} + 2\sum_{j=1}^{5} d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i$$

$$subject\ to:$$

$$g_j(\mathbf{x}) = -2\sum_{i=1}^{5} c_{ij}x_{(10+i)} - 3d_j x_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij}x_i \leq 0, j = 1, \ldots, 5$$

$$x_1, x_2, x_4, x_6, \ldots, x_9, x_{11}, x_{13}, x_{14} \in [0, 10]$$

$$x_3, x_5, x_{10}, x_{12}, x_{15} \in \{0, 1, \ldots, 10\}$$

where $\mathbf{b} = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the remaining data is in Table S-III. The best known optimal solution is $\mathbf{x}^* = (x_1, x_2, x_4, x_6, \ldots, x_9, x_{11}, x_{13}, x_{14}) = (0, 0, 0, 9.99999999, 0, 0, 2.96750117, 0.39963905, 0.82151768, 0.64848398)$ and $\mathbf{y}^* = (x_3, x_5, x_{10}, x_{12}, x_{15}) = (2, 4, 0, 0, 1)$, and $f(\mathbf{x}^*, \mathbf{y}^*) = 41.7399$.

S-III. Influence of the Form of $m(\mathbf{y})$

In this section, we investigated the influence of the form of $m(\mathbf{y})$. Four different forms were considered:

$$L_1 : m(\mathbf{y}) = |y_1 - round(y_1)| + \cdots + |y_{n_2} - round(y_{n_2})|$$

$$L_2 : m(\mathbf{y}) = \sqrt{|y_1 - round(y_1)|^2 + \cdots + |y_{n_2} - round(y_{n_2})|^2}$$

$$L_{ave} : m(\mathbf{y}) = avg(|y_1 - round(y_1)|, \ldots, |y_{n_2} - round(y_{n_2})|)$$

$$L_\infty : m(\mathbf{y}) = max(|y_1 - round(y_1)|, \ldots, |y_{n_2} - round(y_{n_2})|)$$

Based on these four forms, four variants of BOToP were devised, which were named as BOToP-L_1, BOToP-L_2, BOToP-L_{ave}, and BOToP-L_∞. Note that BOToP-L_∞ is the original BOToP. The results provided by these four variants are given in Table S-IV. From Table S-IV, we can observe that these four variants provide similar results. Therefore, BOToP is insensitive to the form of $m(\mathbf{x})$.

TABLE S-IV

Results of BOToP-L_1, BOToP-L_2, BOToP-L_{ave}, and BOToP-L_∞ over 25 Independent Runs. *Ave* and *Std Dev* Indicate the Average and Standard Deviation of the Best Feasible Objective Function Values, Respectively. *FR* and *SR* Indicate the Feasible Rate and the Successful Rate, Respectively. The Wilcoxon's Rank-sum Test at a 0.05 Significance Level Is Performed between BOToP-L_∞ and Each of BOToP-L_1, BOToP-L_2, and BOToP-L_{ave}.

Problem	Status	BOToP-L_1	BOToP-L_2	BOToP-L_{ave}	BOToP-L_∞
F1	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**13.0000 ± 0.0000** ≈	**13.0000 ± 0.0000** ≈	**13.0000 ± 0.0000** ≈	**13.0000 ± 0.0000**
F2	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**1.0000 ± 0.0000** ≈	**1.0000 ± 0.0000** ≈	**1.0000 ± 0.0000** ≈	**1.0000 ± 0.0000**
F3	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**-4.0000 ± 0.0000** ≈	**-4.0000 ± 0.0000** ≈	**-4.0000 ± 0.0000**	**-4.0000 ± 0.0000**
F4	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**-6.0000 ± 0.0000** ≈	**-6.0000 ± 0.0000** ≈	**-6.0000 ± 0.0000** ≈	**-6.0000 ± 0.0000**
F5	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	100%	100%	100%	**100%**
	Ave ± *Std Dev*	**0.2500 ± 0.0000** ≈	**0.2500 ± 0.0000** ≈	**0.2500 ± 0.0000** ≈	**0.2500 ± 0.0000**
F6	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**100%**	**100%**	**100%**	**100%**
	Ave ± *Std Dev*	**-6783.5818 ± 0.0000** ≈	**-6783.5818 ± 0.0000** ≈	**-6783.5818 ± 0.0000** ≈	**-6783.5818 ± 0.0000**
F7	*FR*	84%	**88%**	80%	84%
	SR	**24%**	20%	**24%**	20%
	Ave ± *Std Dev*	NA ≈	NA ≈	NA ≈	NA
F8	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**84%**	76%	80%	80%
	Ave ± *Std Dev*	**7062.5760 ± 25.0670** ≈	7092.5611 ± 26.8736 ≈	7083.3704 ± 29.0367 ≈	7070.3408 ± 26.8423
F9	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	**40%**	32%	32%	**40%**
	Ave ± *Std Dev*	7233.9783 ± 190.9721 ≈	7213.3461 ± 192.4831 ≈	7297.0925 ± 182.8655 ≈	**7209.5545 ± 179.1296**
F10	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	32%	36%	32%	**64%**
	Ave ± *Std Dev*	7453.0022 ± 378.5025 ≈	7414.9998 ± 334.5341 ≈	**7289.3443 ± 327.9877** ≈	7373.3323 ± 369.6853
F11	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	32%	40%	32%	**44%**
	Ave ± *Std Dev*	33.7406 ± 0.3676 ≈	33.7702 ± 0.3222 ≈	33.7210 ± 0.3900 ≈	**33.7144 ± 0.3114**
F12	*FR*	**100%**	**100%**	**100%**	**100%**
	SR	56%	**64%**	52%	60%
	Ave ± *Std Dev*	42.2539 ± 0.7347 ≈	**42.1426 ± 0.8567** ≈	42.3452 ± 0.7296 ≈	42.1461 ± 0.6974
$-/+/\approx$		0/0/12	0/0/12	0/0/12	/