

Research Article

Genetic Algorithm for Mixed Integer Nonlinear Bilevel Programming and Applications in Product Family Design

Chenlu Miao, Gang Du, Yi Xia, and Danping Wang

College of Management and Economics, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Yi Xia; xiayi@tju.edu.cn

Received 12 April 2016; Revised 4 July 2016; Accepted 21 July 2016

Academic Editor: José-Fernando Camacho-Vallejo

Copyright © 2016 Chenlu Miao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many leader-follower relationships exist in product family design engineering problems. We use bilevel programming (BLP) to reflect the leader-follower relationship and describe such problems. Product family design problems have unique characteristics; thus, mixed integer nonlinear BLP (MINLBP), which has both continuous and discrete variables and multiple independent lower-level problems, is widely used in product family optimization. However, BLP is difficult in theory and is an NP-hard problem. Consequently, using traditional methods to solve such problems is difficult. Genetic algorithms (GAs) have great value in solving BLP problems, and many studies have designed GAs to solve BLP problems; however, such GAs are typically designed for special cases that do not involve MINLBP with one or multiple followers. Therefore, we propose a bilevel GA to solve these particular MINLBP problems, which are widely used in product family problems. We give numerical examples to demonstrate the effectiveness of the proposed algorithm. In addition, a reducer family case study is examined to demonstrate practical applications of the proposed BLGA.

1. Introduction

With the evolution of the mass customization paradigm, product family has played an increasingly important role in modern production and has garnered significant attention. Product family optimization design includes product-design-related knowledge, the product family structure, and customization design based on the same product platform to meet customer needs. Many leader-follower relationships exist in product family optimization design problems, for example, between platform and customization design [1], between modular design and product family architecture [2], and between product family and supply design [3, 4]. Consequently, many researchers have used this model to reflect the optimization relationship in product family design. Du et al. [5] formulated a Stackelberg game-theoretic model for joint optimization of product family configuration and scaling design, wherein a bilevel decision structure reveals coupled decision-making between module configuration and parameter scaling. Kristianto et al. [6] demonstrated the potential of a two-stage, bilevel stochastic programming framework for tackling engineer-to-order product customization.

Du et al. [7] proposed a leader-follower joint optimization model of product family configuration and supply chain design, and Ji et al. [8] studied green design modularity using bilevel optimization. In addition, Jiao et al. [9] proposed an underpinning decision structure that distinguishes a de facto leader-follower model rather than a single-level, all-in-one optimization problem.

Due to the impact of Stackelberg game theory, which was proposed by Stackelberg when researching marketing economics, researchers have studied bilevel programming (BLP) since the 1970s. Bracken and Gill proposed BLP models in 1973 and 1977, respectively. Candler and Norton proposed a formal definition of BLP and multilevel programming in their technology reports [10]. BLP models are NP-hard problems. Prominent BLP results primarily include the following: the k th best method for special linear cases [11], replacing a lower-level problem with Karush-Kuhn-Tucker conditions to convert a bilevel problem to a single-level problem [12], and converting the problem to a single level by constructing a penalty function based on the duality gap [13]. The monographs of Bard [14] and Dempe [15] present systematic surveys of BLP models. In leader-follower joint optimization

for product family design, 0-1 mixed integer nonlinear bilevel BLP, which has multiple nonlinear and nonconvex low-level models, is involved. Therefore, traditional methods (e.g., the K-T method for convex bilevel BLP) are limited to solving specific types of the BLP models, which restrict their application.

The BLP used in product family design has unique characteristics. In this type of programming, both continuous and discrete variables such as 0-1 variables (the most commonly used) are employed. In product family design problems, more than one lower-level model, which are nonlinear and nonconvex to model, are required. In addition, in this type of BLP, the leader's constraints often contain follower variables. Mixed integer nonlinear BLP (MINLBP) contains the above characteristics when applied to product family optimization. Note that MINLBP combines integer programming and BLP. However, the discreteness of decision variables and multiple followers make problems more complex. Some researchers prefer to use intelligent algorithms to solve this problem. BLP with multiple lower-level models is more difficult to solve than with a model consisting of only one follower. Therefore, developing an intelligent algorithm to solve the MINLBP used in product family design problems has significant value.

A genetic algorithm (GA) is a method to search for an optimal solution by simulating biological evolution (survival of the fittest). GAs are popular intelligent algorithms that have seen increasingly wide utilization in many fields. GAs have many advantages such as convergence and robustness. Thus, GAs are effective in solving optimization problems. Using a GA to solve a bilevel problem reduces the limitations which traditional methods have, which has been extensively studied. Consequently, many GA monographs for BLP exist. In 1998, Liu designed a GA for multilevel programming with multiple followers, in which high-level models do not contain the low-level models' decision variables [16]. In 2002, Oduguwa and Roy used a GA to solve a BLP scheme that was developed to encourage limited asymmetric cooperation between two players [17]. Kuo and Han applied bilevel linear programming to a supply chain distribution problem and developed an efficient method based on a hybrid of a GA and particle swarm optimization [18], and Lin et al. presented a modified GA to solve a bilevel linear programming network design problem [19]. For a linear bilevel programming problem with interval coefficients in the upper-level objective, Fan and Li propose a genetic algorithm to obtain the best optimal solution and the worst one simultaneously [20]. Li et al. convert an integer linear bilevel programming problem into a single-level binary implicit program and then develop an orthogonal genetic algorithm for solving such problem based on the orthogonal design [21]. Based on a novel coding scheme, Li proposes a genetic algorithm with global convergence to solve nonlinear bilevel programming problems where the follower is a linear fractional program [22]. However, even though GAs have been successfully applied to BLP, it is difficult for such GAs to solve MINLBP in product family design problems. Such GAs focus on solving specific models for other problems. They can solve linear BLP, single-level programming, and BLP with multiple followers, in which leader constraints do

not contain follower decision variables. However, these GAs are not suitable for solving MINLBP, which contains 0-1 decision variables and multiple low-level models. A bilevel GA (BLGA) for MINLBP is proposed to solve product family design optimization problems.

The proposed BLGA can handle MINLBP with both continuous and discrete variables with one or multiple independent nonlinear and nonconvex lower-level models. The remainder of this paper is organized as follows. In Section 2, we pose the primary research question and derive MINLBP. In Section 3, we propose the BLGA and describe its procedures. Numerical examples are provided in Section 3 to illustrate the proposed algorithm. In Section 4, we provide a case study of product family design to demonstrate an application of the proposed method. Conclusions are given in Section 5.

2. MINLBP with One or Multiple Independent Followers

An MINLBP model probably has one or multiple followers. Based on the relationships among the multiple followers, such MINLBP problems can be classified as dependent or independent and, for each particular case, there exists a specific model to describe it [23–25]. The coupling relationships among the followers are not taken into account for most of the bilevel problems in product family design [26, 27]. Therefore, this paper focuses on the MINLBP with independent followers. In an MINLBP model with multiple independent followers, the followers are independent and have no coupling relationship, which means that the objective function and the set of constraints of each follower only include the leader's variables and its own variables [24].

Here, assume the leader first chooses its decision variables $X = \{x_1, x_2, \dots, x_i, \dots, x_m\}$, in which $x_1, x_2, \dots, x_i, \dots, x_m$ are continuous or discrete, and the followers determine their decision variables $Y_i = \{y_{i1}, y_{i2}, \dots, y_{ij}, \dots, y_{ip}\}$, in which $y_{i1}, y_{i2}, \dots, y_{ij}, \dots, y_{ip}$ are continuous or discrete. The MINLBP model with one or multiple independent followers should be formulated as follows:

$$\begin{aligned}
 & \max_X F(X, Y_1, \dots, Y_i, \dots, Y_n) \\
 & \text{s.t. } G(X, Y_1, \dots, Y_i, \dots, Y_n) \leq 0 \\
 & \quad H(X, Y_1, \dots, Y_i, \dots, Y_n) = 0, \\
 & \quad \text{where } Y_i \ (i = 1, \dots, n) \text{ are solved from} \quad (1) \\
 & \min_{Y_i} f_i(X, Y_i) \\
 & \text{s.t. } g_i(X, Y_i) \leq 0 \\
 & \quad h_i(X, Y_i) = 0,
 \end{aligned}$$

where n means the number of followers.

In this programming model, the follower solutions interact with the leader solution. Here, Discrete and continuous variables have upper and lower bounds. Bard [14] proposed the following basic definition for a BLP solution:

(1) Constraint region of the BLP problem:

$$\begin{aligned} S \triangleq \{ & X, Y_1, \dots, Y_n \mid G(X, Y_1, \dots, Y_n) \\ & \leq 0, H(X, Y_1, \dots, Y_n) = 0, g_i(X, Y_i) \\ & \leq 0, h_i(X, Y_i) = 0, i = 1, \dots, n\}. \end{aligned} \quad (2)$$

(2) Feasible set for the i th ($i = 1, \dots, n$) follower for each fixed: $x \in X$:

$$S_i(X) \triangleq \{Y_i \mid g_i(X, Y_i) \leq 0, h_i(X, Y_i) = 0\}. \quad (3)$$

(3) Leader's decision space:

$$\begin{aligned} M(X) \triangleq \{ & X \mid \exists Y_1, \dots, Y_n : G(X, Y_1, \dots, Y_n) \\ & \leq 0, H(X, Y_1, \dots, Y_n) = 0, g_i(X, Y_i) \\ & \leq 0, h_i(X, Y_i) = 0, i = 1, \dots, n\}. \end{aligned} \quad (4)$$

(4) i th ($i = 1, \dots, n$) follower's rational reaction set for $x \in M(x)$:

$$P_i(X) \triangleq \{Y_i \mid Y_i \in \operatorname{argmin} \{f_i(X, Y_i), Y_i \in S_i(X)\}\}. \quad (5)$$

(5) Inducible region of the BLP problem:

$$\begin{aligned} IR \triangleq \{ & (X, Y_1, \dots, Y_n) \mid (X, Y_1, \dots, Y_n) \in S, Y_i \\ & \in P_i(X), i = 1, \dots, n\}. \end{aligned} \quad (6)$$

(6) Optimal solution of BLP problem:

$(X^*, Y_1^*, \dots, Y_n^*)$ is considered the optimal solution of BLP if and only if there exists $(X^*, Y_1^*, \dots, Y_n^*) \in IR$ such that $F(X^*, Y_1^*, \dots, Y_n^*) \leq F(X, Y_1, \dots, Y_n)$ for all $(X, Y_1, \dots, Y_n) \in IR$.

3. BLGA Design

3.1. Assumption. Analysis of feasibility of the proposed algorithm illustrates that two assumptions must be satisfied to obtain good performance. These assumptions ensure the effectiveness of the proposed BLGA by restraining the algorithm's process.

The first assumption is that solutions must meet the requirements of BLP, which means that the solutions must satisfy the constraints of both the upper- and lower-level programs. Meaningful solutions can only be obtained if the algorithm is feasible.

According to the concepts about the solution, we know that if (X, Y_1, \dots, Y_n) is within the feasible region, it is denoted as follows:

$$(X, Y_1, \dots, Y_n) \in S. \quad (7)$$

We propose a BLGA that meets these constrain requirements. First, the binary code encodes the leader variables and initializes them according to their bound constraints. Second,

by taking the initial leader variables as parameters, each follower programming uses the lower-level GA to obtain the best solution in the last generation. Note that the best solution in the last generation provided by the lower-level GA is a good approximation for the optimal solution, which meets the lower-level constraints. Then the lower-level approximation solutions are brought back to the upper-level GA to solve the leader program. In the upper-level GA, the solutions should be verified whether they satisfy the leader's constraints.

The second assumption is that the algorithm has convergence. As the proposed BLGA meets the first condition, it can obtain the optimal solution only if it has convergence, which is an important measurement of computing capacity. Convergence of a GA means that the last cycle of a finite series of cycles can yield the optimal solution. This assumption ensures that, after comparing feasible solutions, the optimal solution can be found. An algorithm is only effective and feasible when it satisfies this condition.

Currently, many GAs have been designed to solve special types of problems that have convergence [28–30]. Markov chains are often used to evaluate convergence. If a GA whose mutation rate is fixed and greater than 0 takes an elitist strategy and the binary code strategy, it has convergence that cannot be influenced by the initial population. Using the elitist strategy allows the best chromosome to be removed from operations in order to avoid disappearing in the operations. Moreover, many factors influence convergence, such as the maximal number of generations, crossover rate, mutation rate, and crossover and mutation methods.

According to the above assumptions, even though the best solution provided by the BLGA does not guarantee to be the optimal solution, it may be a good approximation for the global optimal solution in reasonable time. We can calculate the problem several times, using different seeds, to find the closest approximation solutions. Numerical examples are given in Section 3.4 to verify the algorithm.

3.2. Design Process of BLGA. On the basis of the above assumptions, this study proposes a BLGA to solve MINLBP with one or multiple independent followers that can improve efficiency in generating solutions. The essential process is as follows. First, leader decision variables are initialized according to their bound constraints. Second, each lower-level GA takes leader decision variables $X = \{x_1, x_2, \dots, x_j, \dots, x_m\}$ as parameters to solve the i th follower problem to determine the best solution $Y_i^*(X)$ and the value of $f_i^*(X, Y_i^*(X))$, where $i = 1, \dots, n$. Third, the best solution $Y_i^*(X)$ for each follower and the leader decision variables X is returned to the leader problem to determine if they satisfy other leader constraints. Their fitness values are then evaluated. Fourth, selection, crossover, and mutation operations are performed for these leader decision variables. Fifth, this cycle continues until maximal iterations are reached. The best optimal solutions X^* and Y_1^*, \dots, Y_n^* are then recorded.

However, there is no guarantee that the optimal solutions will be the best solution for MINLBP with one or multiple independent followers because the GA can only ensure that solutions belong to the constraint region rather than the

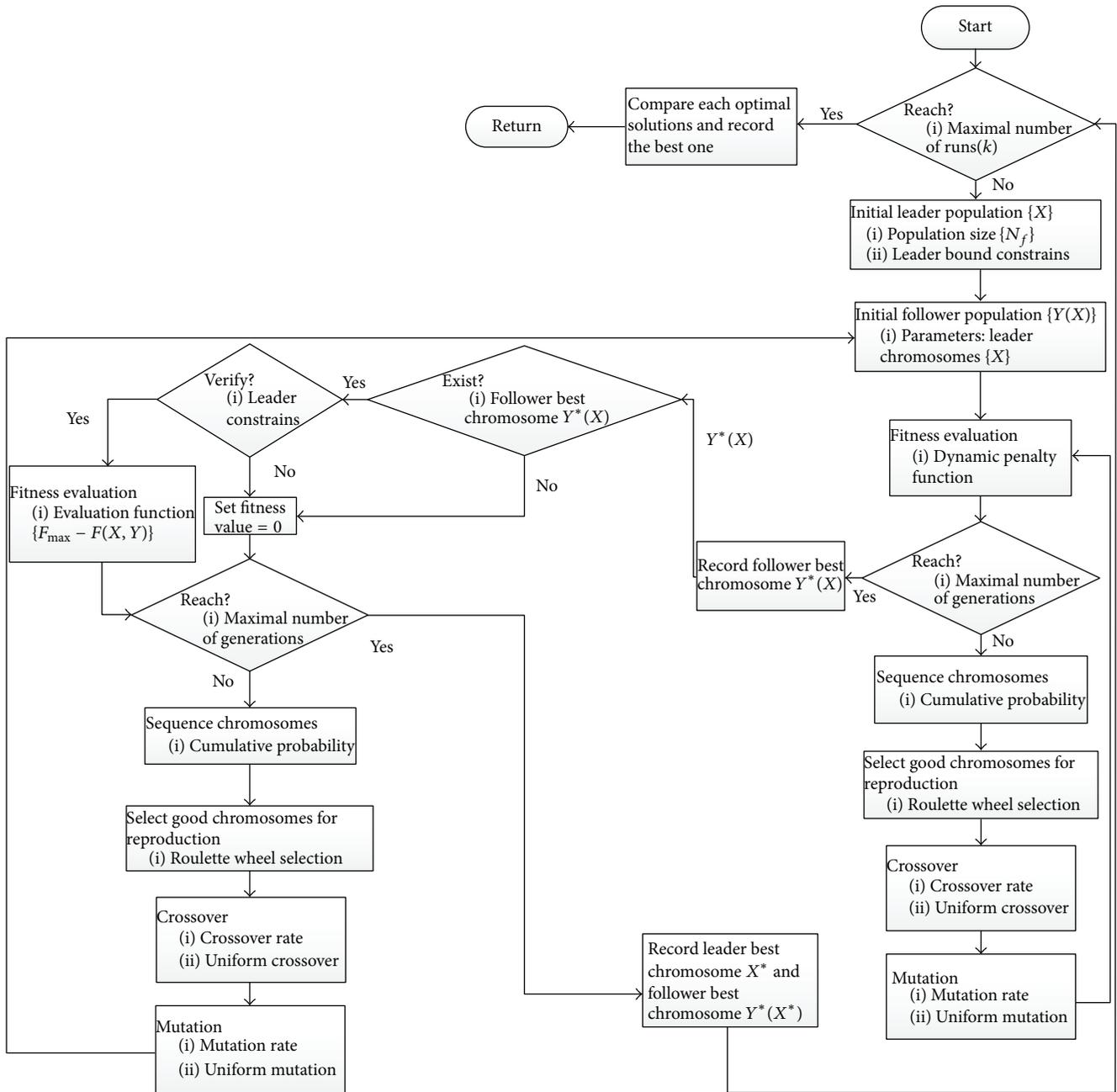


FIGURE 1: Process of the proposed BLGA.

inducible region. It is necessary to run this process several times to obtain several optimal solutions. These optimal solutions are then compared to determine the best solution. The process of the proposed BLGA is shown in Figure 1.

3.3. *BLGA Procedure.* The procedure of the proposed BLGA is as follows.

Step 1 (verify if the process is terminated). The termination condition is that the number of runs has reached the maximal

number. To determine if the termination condition has been satisfied, the maximal number of runs k must be input. If this condition is satisfied, each optimal solution is compared to determine the best solution. If the termination condition is not satisfied, the process proceeds to Step 2.

Step 2 (initialize the leader population with leader population size N_f). According to their bound constraints, the leader population is initialized and these variables are encoded. Two strategies are used for the encoding of variables. The proposed method uses the binary code strategy, which uses

binary vectors as chromosomes, to represent the values of continuous variables with unequal lengths. The variable encoding length l is expressed as follows:

$$l = \left\lceil \log_2 \left(\frac{u_{\max} - u_{\min}}{\delta} \right) \right\rceil + 1, \quad (8)$$

where u_{\max} and u_{\min} represent the upper and lower bounds of the variables, respectively, and δ is precision. The proposed method also uses the real code strategy for encoding, which uses ordered arrays to represent discrete variable values.

Step 3 (run lower-level GAs). Each follower programming takes the initial leader population $X = \{x_1, x_2, \dots, x_i, \dots, x_m\}$ as a parameter. Note that the processes for each lower-level GA are the same. First, the follower population size N_f is input, and the follower population is initialized according to their bound constraints. The two strategies described in Step 2 are used to encode the population. Second, verify that the follower chromosomes satisfy the follower constraints. If the follower chromosome does not satisfy the requirement, set the fitness values of it to zero. The others are evaluating the fitness value based on the follower objective function. Third, the maximal number of generations is input to verify whether the current number of generations is equivalent to the maximal number. If the current generation reaches the maximal number, the best follower chromosome $Y_i^*(X)$, $i = 1, \dots, n$, is recorded and the solution's feasibility is verified. If the solution meets the lower-level constraints, it is returned to the leader program or else null is returned. If the current population does not reach the maximal number, selection, crossover, and mutation operations are performed and the process continues.

Step 4 (calculate the leader fitness evaluation). The leader programming takes the best chromosome of each follower $Y_i^*(X)$, $i = 1, \dots, n$, as a parameter. Then, verify if $Y_i^*(X)$, $i = 1, \dots, n$, are null values. If at least one is null, the fitness value is set to zero. If no null values are found, $(X^*, Y_1^*(X), \dots, Y_n^*(X))$ are taken into the leader constraints. If it satisfies the constraints, the objective function $F(X^*, Y_1^*, \dots, Y_n^*)$ is calculated and all $F(X^*, Y_1^*, \dots, Y_n^*)$ are sorted in descending order to obtain the largest value, that is, F_{\max} . The current F_{\max} is compared with the previous F_{\max} , and the larger value is recorded as the new F_{\max} . The fitness function $F_{\max} - F(X^*, Y_1^*, \dots, Y_n^*)$, which can ensure nonnegative fitness values, is used to evaluate the individuals and record their fitness values. If the constraints are not satisfied, the fitness value is set to zero.

Step 5 (verify if the cycle is terminated). The termination condition is that the number of generations has reached the maximal number. If the maximal number is reached, the chromosome with the greatest fitness value is considered the best solution. The best leader chromosome X^* and each best follower chromosome $Y_i^*(X)$, $i = 1, \dots, n$, for each generation are recorded and the process returns to Step 1. If the maximal number has not been reached, the process continues to Step 6. Note that inputting a correct maximal number of generations is very important to an algorithm's

convergence. If the number is too small, the algorithm may not find the best solution after termination of the cycle. In contrast, if the number is too large, efficiency may be reduced. A careful and deliberate choice is required.

Step 6 (selection operation). The roulette wheel selection method is used to select $N_F - 1$ good chromosomes as parents for the crossover operation. Simultaneously, another good chromosome is selected as the best chromosome within the current population according to the elitist strategy.

Step 7 (crossover operation). Input the crossover rate p_c . Each of the parents selected in Step 6 is picked randomly, and each pair uses a uniform crossover method to undergo crossover with the given probability. Note that the new population includes these new chromosomes.

Step 8 (mutation operation). The proposed method uses two mutation methods. For continuous variables that use the binary code strategy for encoding, the mutation rate p_m is input and the uniform mutation method with the given probability is used. For discrete variables that use the real code strategy, new chromosomes are formulated as follows:

$$x_i^\wedge = [(x_i - lb_i + 1) \bmod (ub_i - lb_i)] + lb_i, \quad (9)$$

where x_i and x_i^\wedge are the individuals before and after mutating, respectively, and lb_i and ub_i are the lower and upper bounds of the integer variables, respectively. After completion of Step 8, the process returns to Step 3.

3.4. Numerical Examples. The proposed BLGA for solving MINLBP with one or multiple independent followers is realized using MATLAB. To verify the effectiveness and robustness of the proposed BLGA, we give numerical examples obtained with a personal computer with the following parameters: population size is 50; maximal number of generations is 200; precision of binary code is 0.01; crossover rate is 0.8; mutation rate is 0.01; number of experiments is fifteen. The results were compared with the exact solutions obtained from traditional methods for solving MINLBP to test the quality of the computed solution. The best, worst, average, and median results and standard deviation value are presented.

Example 1 (leader and follower decision variables are discrete).

$$\begin{aligned} \min_X \quad & F(X, Y) = -x_1^2 - 3x_2 - 4y_1^2 + y_2^2 \\ & x_1^2 + 2x_2^2 - 4 \leq 0 \\ & x_1 + x_2 - y_1 - y_2 \leq 0 \\ & x_1 \in \{0, 1, 2\} \\ & x_2 \in \{0, 1, 2\} \end{aligned}$$

TABLE 1: Results of Example 1.

	$F^*(X^*, Y^*)$	X^*	$f^*(X^*, Y^*)$	Y^*
Best	0	(1, 1)	-8	(0, 2)
Worst	0	(2, 0)	-2	(0, 2)
Average	0	—	-5.60	—
Median	0	—	-8	—
Standard deviation value	0	—	3.04	—
Enumeration method	0	(1, 1)	-8	(0, 2)

$$\begin{aligned}
\min_Y \quad & F(X, Y) = 2x_1^2 + y_1^2 - 5y_2 \\
& -2y_1 + y_2 \leq 3 + x_1^2 - 2x_1 + x_2^2 \\
& y_1 \in \{0, 1, 2\} \\
& y_2 \in \{0, 1, 2\}.
\end{aligned} \tag{10}$$

Example 1 is a simple integer nonlinear BLP problem that can be solved using the enumeration method. The best, worst, average, and median results and standard deviation value are shown in Table 1.

Table 1 shows the results of Example 1. The standard deviation value of $F^*(X^*, Y^*)$ is zero, which means that each run of this BLGA in this test finds the same upper objective function value. However, the standard deviation of $f^*(X^*, Y^*)$ is 3.04, which means that there exists more than one optimal solution found by the proposed BLGA in this example. From the result of each run, we find that the solutions $X = (2, 0)$ and $X = (1, 1)$ are the optimal solutions. However, even though both of these solutions have the same upper optimal solution, their lower optimal solutions are different. When the upper-level solution is $X = (2, 0)$, the lower-level result is -2 . When the upper-level solution is $X = (1, 1)$, the lower-level result is -8 . The upper solution $X = (1, 1)$ provides a better value for the lower-level objective function. Therefore, the best solution for this test is $X = (1, 1)$ and $Y = (0, 2)$. The iterative process is shown in Figure 2. As seen in Figure 2(a), the maximal number of generations does not need to be 200 because the optimal solution has been determined in the first generation. Figure 2(b) shows the BLGA evolution processes for the upper-level and lower-level optimizations.

Example 2 (continuous leader and follower decision variables).

$$\begin{aligned}
\min_x \quad & F(x, y) = x + 3y \\
& y - x \leq 0 \\
& 1 \leq x \leq 6 \\
\min_y \quad & F(x, y) = -y \\
& x + y \leq 8 \\
& -x - 4y \leq -8 \\
& x + 2y \leq 13.
\end{aligned} \tag{11}$$

TABLE 2: Results of Example 2.

	$F^*(x^*, y^*)$	x^*	$f^*(x^*, y^*)$	y^*
Best	12.0030	6.0000	-2.0010	2.0010
Worst	12.1596	5.9217	-2.0793	2.0793
Average	12.0134	—	-2.0010	—
Median	12.0030	—	-2.0070	—
Standard deviation value	0.0404	—	0.0217	—
Graphic method	12.0000	6	-2.0000	2.0000

Example 2 is simple continuous BLP that can be solved by the graphic method. The best, worst, average, and median results and standard deviation value are shown in Table 2.

As can be seen, the difference between the values of upper-level objective functions of the best result calculated by the proposed BLGA and the exact method is 0.0030, which is a negligible difference. This demonstrates the effectiveness of the proposed BLGA. In these 15 independent runs, except the worst result, the other results are the same. The iterative process of the best result is shown in Figure 3(a). As can be seen, the maximal number of generations does not need to be 200 because the optimal solution has been determined in the fifth generation. Figure 3(b) shows the BLGA evolution processes for the upper-level and lower-level optimizations.

Example 3 (leader decision variables are discrete and follower decision variables are continuous).

$$\begin{aligned}
\min_x \quad & F(X, Y) = -\frac{(x_1 + y_1)(x_2 + y_2)}{1 + x_1y_1 + x_2y_2} \\
& x_1^2 + x_2^2 \leq 100 \\
& x_1^2 - x_2^2 \leq y_1^2 - y_2^2 \\
& x_1 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\
& x_2 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\
\min_Y \quad & F(X, Y) = \frac{(x_1 + y_1)(x_2 + y_2)}{1 + x_1y_1 + x_2y_2} \\
& 0 \leq y_1 \leq x_1 \\
& 0 \leq y_2 \leq x_2.
\end{aligned} \tag{12}$$

The results calculated by the proposed BLGA and the exact method are shown in Table 3.

Table 3 shows the best and the worst results among these 15 runs, whose objective function values are the same as that calculated by the exact method, even though the solutions are different. The standard deviation value is near zero, which means that each independent run draws the similar objective function values. Figure 4(a) shows the iterative process of a best run, and the evolution processes for the upper-level and lower-level GAs are shown in Figure 4(b).

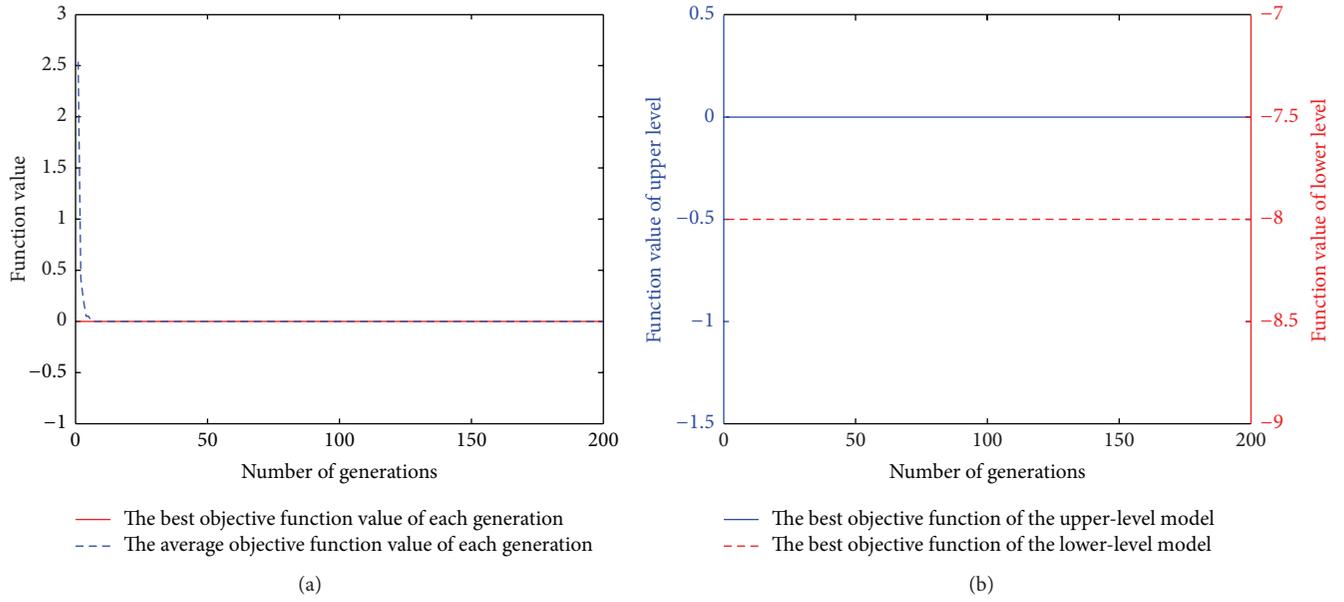


FIGURE 2: Iterative process of Example 1.

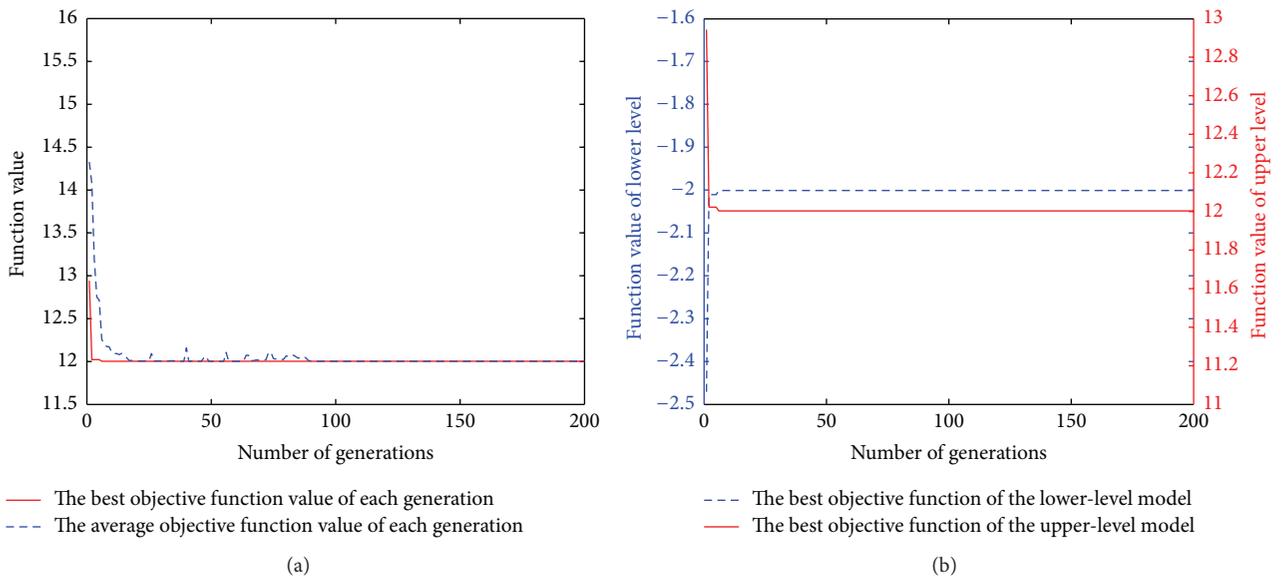


FIGURE 3: Iterative process of Example 2.

TABLE 3: Results of Example 3.

	$F^*(X^*, Y^*)$	X^*	$f^*(X^*, Y^*)$	Y^*
Best	-1.9600	(7, 7)	1.9600	(7.0000, 0.0008)
Worst	-1.9600	(7, 7)	1.9600	(0.0009, 7.0000)
Average	-1.9600	—	1.9600	—
Median	-1.9600	—	1.9600	—
Standard deviation value	6.8951×10^{-16}	—	6.8951×10^{-16}	—
Exact method	-1.9600	(7, 7)	1.9600	(7.0000, 0.0000)

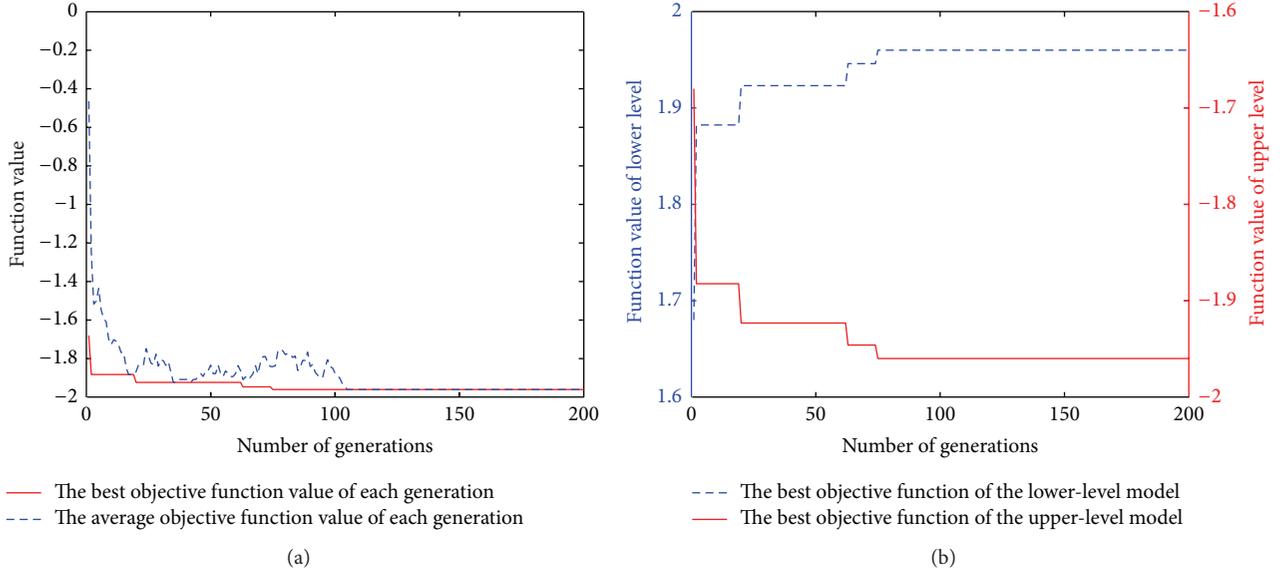


FIGURE 4: Iterative process of Example 3.

TABLE 4: Results of Example 4.

	$F^*(X^*, Y^*)$	x^*	$f^*(x^*, y^*)$	y^*
Best	1.0000	1.0039	1.5319×10^{-5}	1
Worst	1.0333	1.0011	1.1000×10^{-3}	1
Average	1.0045	—	1.8139×10^{-4}	—
Median	1.0000	—	1.5319×10^{-5}	—
Standard deviation value	0.1171	—	0.0004	—
Exact method	1.0000	1.0000	0.0000	1

Example 4 (leader decision variables are continuous and follower decision variables are discrete).

$$\begin{aligned}
 \min_x \quad & F(x, y) = (x - 1)^2 + (y - 2)^2 \\
 & y - x \leq 0 \\
 & 1 \leq x \leq 3 \\
 \min_y \quad & F(x, y) = (x + y - 2)^2 \\
 & x + y \leq 5 \\
 & y \in \{0, 1, 2, 3\}.
 \end{aligned} \tag{13}$$

The results are shown in Table 4.

In these 15 runs, there are 6 times in which the proposed BLGAs draw the different solutions with the best result. However, these differences are so small that can be negligible. Moreover, the best result is rather similar to the exact solution, which demonstrates the effectiveness of the proposed BLGA. The iterative process is shown in Figure 5(a), and the BLGA evolution processes for the upper-level and lower-level optimizations are shown in Figure 5(b).

Example 5 (MINLBP with multiple independent followers).

$$\begin{aligned}
 \min_x \quad & F(x, Y_1, Y_2) \\
 & = (y_{11} + y_{21} - 200)(y_{11} + y_{21}) \\
 & \quad + (y_{12} + y_{22} - 160)(y_{12} + y_{22}) \\
 & x_1 + x_2 + x_3 + x_4 \leq 40 \\
 & y_{11} + y_{22} \leq x_3 \\
 & 1 \leq x_1 \leq 10 \\
 & 1 \leq x_2 \leq 5 \\
 & 1 \leq x_3 \leq 15 \\
 & 1 \leq x_4 \leq 20 \\
 \min_{Y_1} \quad & f(x, Y_1) = (y_{11} - 4)^2 + (y_{12} - 4)^2 \\
 & 0.4y_{11} + 0.7y_{12} \leq x_1 \\
 & 0.6y_{11} + 0.3y_{12} \leq x_2 \\
 & 0 \leq y_{11} \leq 20 \\
 & y_{12} \in \{0, 1, 2, \dots, 20\} \\
 \min_{Y_2} \quad & f(x, Y_2) = (y_{21} - 35)^2 + (y_{22} - 2)^2 \\
 & 0.4y_{21} + 0.7y_{22} \leq x_3 \\
 & 0.6y_{21} + 0.3y_{22} \leq x_4 \\
 & 0 \leq y_{21} \leq 40 \\
 & y_{22} \in \{0, 1, 2, \dots, 40\}.
 \end{aligned} \tag{14}$$

The results are shown in Table 5.

TABLE 5: Results of Example 5.

	F^*	X^*	f_1^*	Y_1^*	f_2^*	Y_2^*
Best	-6598.6	(7.61, 4.66, 11.09, 16.64)	16.5284	(3.27, 9)	56.8328	(27.73, 0)
Worst	-6512.8	(4.82, 2.90, 12.80, 19.38)	54.8902	(1.57, 6)	12.94	(32.01, 0)
Average	-6568.81	—	25.0162	—	52.3811	—
Median	6572.40	—	9.8871	—	72.1481	—
Standard deviation value	22.0338	—	15.1987	—	22.4542	—
Exact method	-6600	(7.91, 4.37, 11.09, 16.63)	11.9241	(2.29, 10)	64.6841	(27.21, 0)

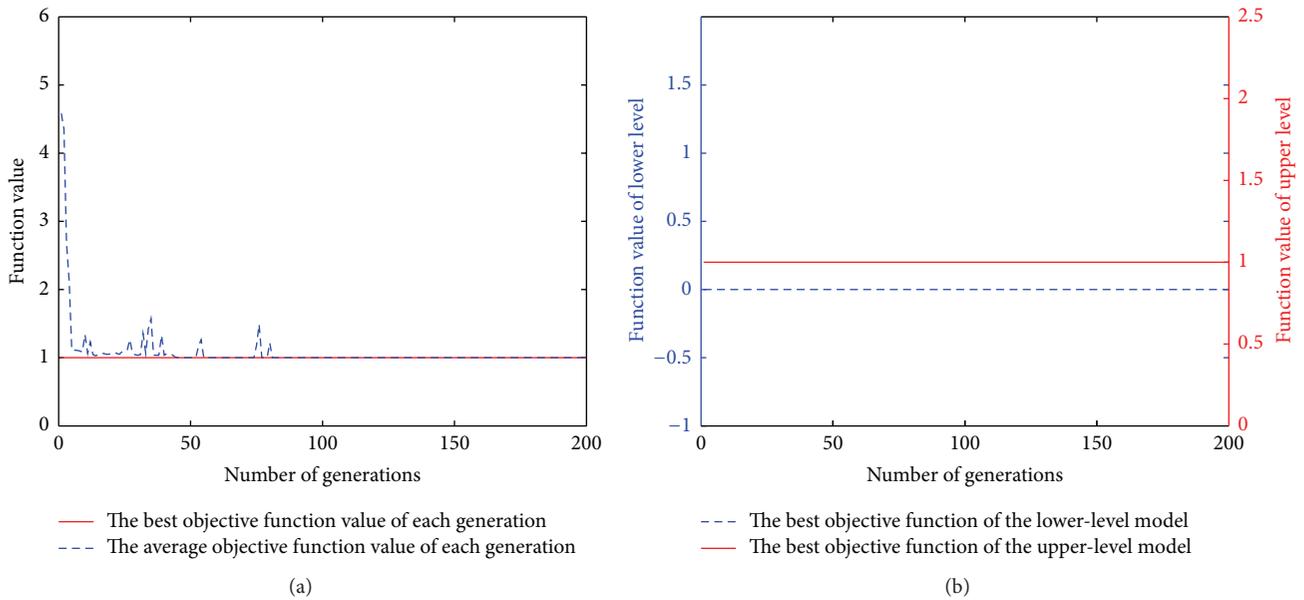


FIGURE 5: Iterative process of Example 4.

Table 5 shows the basic information about these 15 independent runs. Note that the differences between the best objective function values calculated by the proposed BLGA and the exact method are negligible. The iterative process is shown in Figure 6(a), and the BLGA evolution processes for the upper-level and lower-level optimizations are shown in Figure 6(b).

3.5. *Algorithm Evaluation.* Based on the characteristic of MINLBP with one or multiple independent followers in product family design (PFD), we have proposed a BLGA to solved this problem. We use five numerical examples to demonstrate the effectiveness and robustness of the proposed BLGA. We summarize the advantages of the proposed BLGA as follows.

First, MINLBP with one or multiple independent followers has generality, which can illustrate many practical management and engineering problems, such as PFD. Many studies have examined using GAs to solve special BLP problems. Some algorithms require that leader constraints include no follower decision variables, which cannot solve the target MINLBP problems. In addition, some algorithms require only continuous decision variables, which limit practical application. In theory, the proposed BLGA has no limitations on constraints and variables.

Second, the optimal solutions calculated by the proposed algorithm belong to the constraint region. Note that to ensure the feasibility of the optimal solutions, we have made the following assumptions prior to designing the proposed BLGA. In the BLGA, the fitness values of individuals that do not satisfy constraints are set to zero to avoid being selected as parents. This ensures that the optimal solutions satisfy all constrains, which belong to the constraint region.

Third, the proposed BLGA has good effectiveness and strong robustness, which has been demonstrated through the five numerical examples. Note that we use the binary code strategy to encode so that we can set the precision according to practical requirements. This strategy suits the mechanism of GAs, which can search for optimal solutions carefully.

4. Case Study

4.1. *Problem Description.* A case study of a simplified reducer family design is shown to demonstrate the proposed BLGA. A single-stage gear reducer includes three types of customized reducers, whose transmission ratios are 2, 3, and 4, respectively. To reduce cost and produce the three reducers simultaneously, we design platform parts and different customization parts, respectively. In this case, platform parts include a shell,

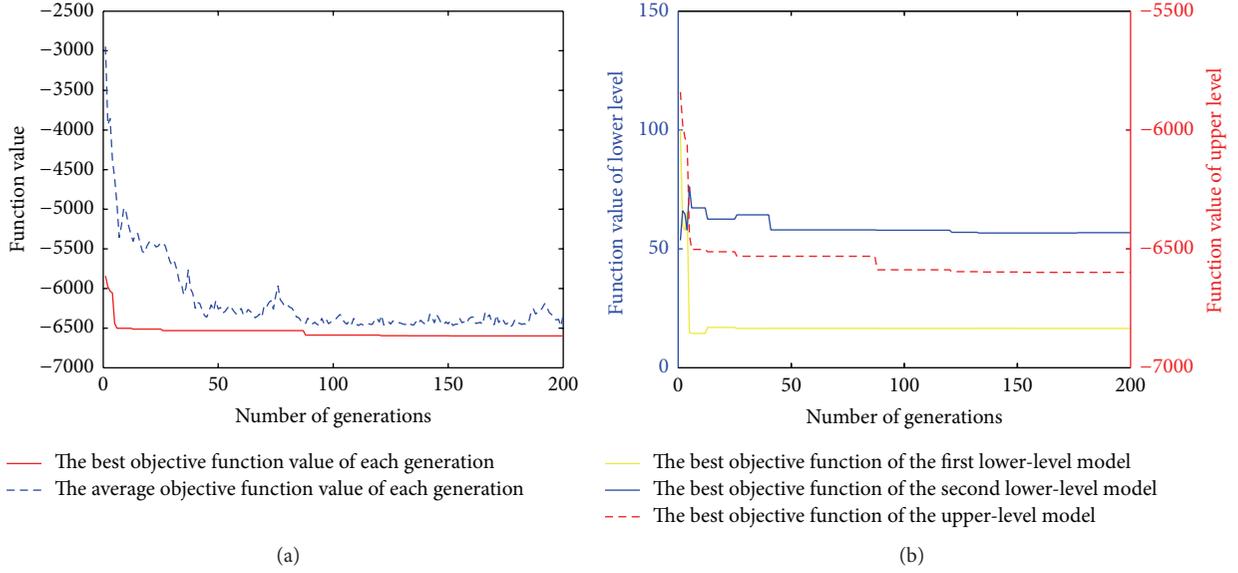


FIGURE 6: Iterative process of Example 5.

a high-speed shaft, and a low-speed shaft, whose critical parameters are taken as the platform variables. The different customization parts include the high-speed gear and the low-speed gear, whose critical parameters are taken as the customization variables. The critical parameter of platform parts is shell size, which is described by the center distance and diameter of high-speed shaft d_H . The center distance is proportional to the total number of gears when the module is determined; thus, we take the total number of gears z_s as a platform variable. To optimize customization parts, we must optimize the selection of materials of gears y_{ij} (a 0-1 variable) and the number of teeth of high-speed gear z_{H_i} ($i = 1, 2, 3$; the total number of gears is the sum of the number of low-speed gear and high-speed gear teeth). Here, $y_{ij} = 1$ indicates that the i th reducer uses the j th material.

In the design process of the reducer family, we cannot consider platform optimization and customization optimization simultaneously. The platform designer who optimizes the platform aims to leverage enterprise profitability, which includes maximizing the performance of all reducers and minimizing total cost. The customization designers design customization variables to maximize the performance of each reducer. Note that these designers have different goals and decision variables; however, their decisions affect each other. This is a joint optimization problem. Compared with the customization designers, platform designers act as leaders because the platform is the basis of a product family, which is more important. The customization designers act as followers. Consequently, this case can be considered a Stackelberg game-based decision-making problem.

4.2. Model Development. In this reducer family design problem, the leader is the platform design, whose decision variables are the total number of gears z_s and the diameter of high-speed shaft d_H . The objective is to maximize the

performance of all reducers and minimize total cost. The constraints are the ranges of the decision variables.

In this problem, the performance of the reducer family is the sum of the performance of each reducer, which is influenced by bending resistance and abrasion resistance. We use the bending fatigue stress of high-speed gear σ_{FH_i} to define the bending resistance of the i th reducer because bending resistance affects the bending and fracture of a gear and, compared with a low-speed gear, the high-speed gear is easier to break. Because it is influenced by contact fatigue stress, the abrasion resistance is defined by the contact fatigue stress σ_{H_i} . In this simplified case, we describe the evolution of fatigue stress of the i th reducer by the product of two types of fatigue stress. The evolution of fatigue stress of the reducer family is $\sum \sigma_{H_i} \sigma_{FH_i}$. This has an inverse relationship with the performance of the reducer; thus, the goal of maximizing the performance of all reducers can be replaced by minimizing the evolution of fatigue stress of the reducer family.

Here, the total cost is the sum of the costs of each reducer, which is influenced by manufacturing materials. Assume that different types of reducers that use the same type of material have the same production cost. The customization designer can select the manufacturing materials from metals A, B, and C. The production costs C_j ($j = 1, 2, 3$) are 50, 70, and 100, respectively. We define y_{ij} as 0-1 variables that show whether the i th reducer uses the j th material. The production cost of the i th reducer is $\sum C_j y_{ij}$, and the production cost of the reducer family is $\sum \sum C_j y_{ij}$.

We can describe the leader's objective of the upper-level model as minimizing the product between the evolution of fatigue stress and the total cost, which is given as follows:

$$\min \sum_{i=1}^3 \sigma_{H_i} \sigma_{FH_i} \sum_{j=1}^3 C_j y_{ij}. \quad (15)$$

TABLE 6: Material coefficients of metals.

	Metal A ($j = 1$)	Metal B ($j = 2$)	Metal C ($j = 3$)
Elastic coefficient Z_{E_j}	200 MPa ^{0.5}	190 MPa ^{0.5}	180 MPa ^{0.5}
Contact fatigue stress $[\sigma]_{H_j}$	300 MPa	400 MPa	500 MPa
Bending fatigue stress $[\sigma]_{FH_j}$	100 MPa	200 MPa	300 MPa

The lower-level models attempt to optimize the three types of reducers. For the i th reducer, the decision variables are the diameter of low-speed shaft d_{L_i} , the number of teeth of high-speed gear z_{H_i} , and the materials selection variable y_{ij} ($i, j = 1, 2, 3$). The objectives of the lower-level models are to minimize the cost of reducer $C_j y_{ij}$. The constraints are the restrictions of fatigue stress, dimensions, ranges of indicators, ranges of variables, and material selections.

The constraints of fatigue stress are that bending fatigue stress and the contact fatigue stress are less than allowable stress, respectively, which are given as follows:

$$\begin{aligned} \sigma_{H_i} &= Z_H \sum_{j=1}^3 Z_{E_j} y_{ij} \sqrt{\frac{2KT_{H_i}}{b_i m^2 z_{H_i}^2} \cdot \frac{u'_i + 1}{u'_i}} \\ &\leq \sum_{j=1}^3 [\sigma_{H_i}] y_{ij}, \end{aligned} \quad (16)$$

$$\sigma_{FH_i} = \frac{2KT_{H_i} Y_{FSH_i}}{b_i z_{H_i} m^2} \leq \sum_{j=1}^3 [\sigma_{FH_i}] y_{ij},$$

where loading coefficient K is 1.4, region coefficient Z_H is 2.5, tooth width b_i is 80 mm, and the torque of the high-speed shaft is 1.27×10^5 N·mm. The different material coefficients of metal are given in Table 6. The dimension constraints dictate that the diameters of the reference circles of low-speed and high-speed gears are greater than 20 mm longer than the diameters of the low-speed shaft and high-speed shaft, respectively, which are given as follows:

$$\begin{aligned} D_{H_i} &= m z_{H_i} \geq d_{H_i} + 20, \\ D_{L_i} &= m z_{L_i} \geq d_{L_i} + 20. \end{aligned} \quad (17)$$

Suppose the margin of error for the actual transmission ratio of the reducers is 10% above or below the nominal transmission ratio. The constraints of the indicator ranges include the ranges of the actual transmission ratio, which is expressed as follows:

$$u_i^L \leq u_i = \frac{z_S - z_{H_i}}{z_{H_i}} \leq u_i^H. \quad (18)$$

Each customization designer can only select a single material. This constraint is expressed as follows:

$$\sum_{j=1}^3 y_{ij} = 1. \quad (19)$$

In addition, there are also constraints for the ranges of variables. These constraints are given in Table 7. Note that

TABLE 7: Ranges of variables and indicators.

Variables and indicators	Range
Diameters of high-speed shaft d_H	40–60 mm
Total number of gears z_S	140~160
Diameters of low-speed shaft d_{L_i}	60–100 mm
Number of teeth of high-speed gear z_{H_i}	25–60
Transmission ratio of reducer 1 u_1	1.8–2.2
Transmission ratio of reducer 2 u_2	2.7–3.3
Transmission ratio of reducer 3 u_3	3.6–4.4

the upper and lower bounds are represented by superscripts U and L .

According to our analysis, an MINLBP model with multiple independent followers of the leader-follower joint optimization for the reducer family design can be expressed as follows:

$$\begin{aligned} \min_{d_H, z_S} \quad & \sum_{i=1}^3 \sigma_{H_i} \sigma_{FH_i} \sum_{j=1}^3 C_j y_{ij} \\ \text{s.t.} \quad & d_H^L \leq d_H \leq d_H^U \\ & z_S^L \leq z_S \leq z_S^U \\ \min_{d_{L_i}, z_{H_i}, y_{ij}} \quad & \sum_{j=1}^3 C_j y_{ij} \\ \text{s.t.} \quad & \sigma_{H_i} = Z_H \sum_{j=1}^3 Z_{E_j} y_{ij} \sqrt{\frac{2KT_{H_i}}{b_i m^2 z_{H_i}^2} \cdot \frac{u'_i + 1}{u'_i}} \\ & \sigma_{FH_i} = \frac{2KT_{H_i} Y_{FSH_i}}{b_i z_{H_i} m^2} \\ & \sigma_{H_i} \leq \sum_{j=1}^3 [\sigma_{H_i}] y_{ij} \\ & \sigma_{FH_i} \leq \sum_{j=1}^3 [\sigma_{FH_i}] y_{ij} \\ & m z_{H_i} \geq d_{H_i} + 20 \\ & m (z_S - z_{H_i}) \geq d_{L_i} + 20 \\ & u_i^L \leq \frac{z_S - z_{H_i}}{z_{H_i}} \leq u_i^H \end{aligned}$$

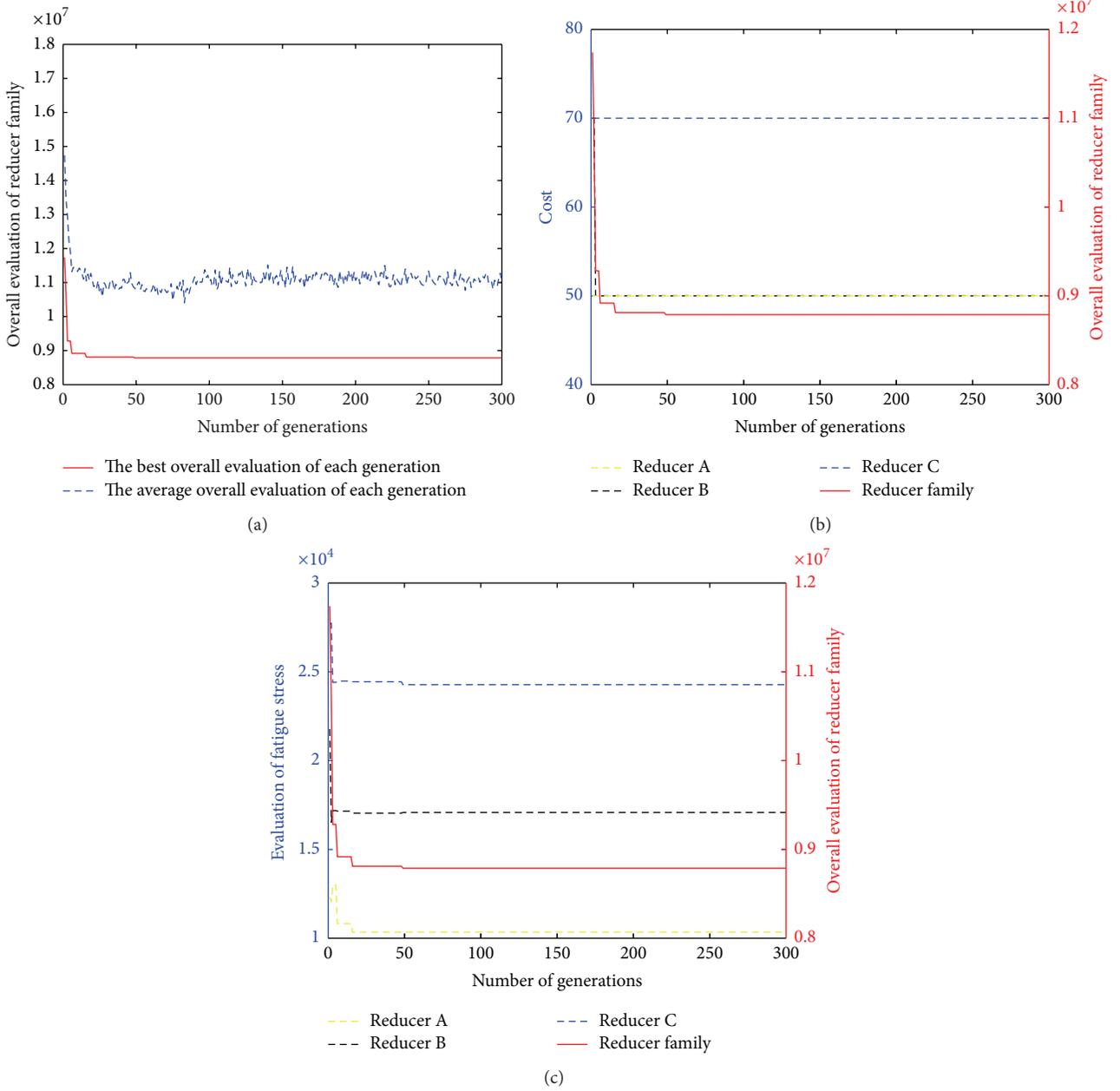


FIGURE 7: Iterative process.

$$\sum_{j=1}^3 y_{ij} = 1, \quad y_{ij} = 1 \text{ or } 0$$

$$d_{L_i}^L \leq d_{L_i} \leq d_{L_i}^U$$

$$z_{H_i}^L \leq z_{H_i} \leq z_{H_i}^U$$

$$i = 1, 2, 3.$$

(20)

The material coefficient of metal and the ranges of variables and indicators are shown in Tables 6 and 7, respectively.

4.3. Solution and Analysis of Results. This model is an MINLBP problem with three independent followers. We use the proposed BLGA to solve this problem with the following parameters: population size is 50; maximal number of generations is 300; precision of binary code is 0.01; and crossover rate is 0.8. Figure 7(a) shows that the best optimal solutions are found in the fiftieth generation. Figure 7(b) shows the evolution processes of the overall evaluation and cost. Figure 7(c) shows the overall evolution processes of the fatigue stress of the family and each reducer. The optimal results are given in Table 8, which shows the design of each reducer.

TABLE 8: Results.

	Platform variables		Customization variables			Upper-level goal		Lower-level goal	
	Diameter of high-speed shaft d_H/mm	Total number of gears z_S	Diameter of low-speed shaft d_{L_i}/mm	Number of teeth of high-speed gear z_{H_i}	Selections in metal y_{ij}	Overall evaluation of reducer family	Cost of reducer	Evaluation of fatigue stress	
Reducer A	52.2627	159.6643	77.4652	57.0171	A	8.7881×10^6	50	1.0337×10^5	
Reducer B			64.0830	43.0684	A		50	1.7082×10^5	
Reducer C			79.6435	34.6496	B		79	2.4275×10^5	

5. Conclusion and Further Study

MINLBP can accurately model product family design, which has one or multiple nonlinear and nonconvex independent low-level models. Based on the characteristics of the model in product family optimization, the primary purpose of this study has been to propose a bilevel BLGA to solve product family optimization models. The proposed BLGA applies a GA to both the leader and followers and uses appropriate encoding strategies to handle continuous and discrete decision variables, which are common in product family optimization. The proposed BLGA overcomes many limitations of previously proposed GAs. For example, such GAs cannot solve a model in which leader constraints do not contain follower decision variables, which is common in product family optimization. The proposed BLGA overcomes this limitation. This paper models the joint optimization between product platform and product family designs by an MINLBP approach in detail and presents a case study of a simplified reducer family design to demonstrate the feasibility and effectiveness of this model and BLGA.

Currently, BLP is widely used in product family design optimization, which is a hot topic in engineering. GAs can solve BLP problems effectively. This paper only presents a specific application of a simplified reducer family design which describes the joint optimization between product platform design and product family design. In the future, we can do more research on other leader-follower relationships in the product family design problems and improve the BLGA efficiency by adjusting the encoding strategies and operation strategies.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This project is supported by the National Natural Science Foundation of China (nos. 71071104 and 71371132).

References

- [1] R. U. Nayak, W. Chen, and T. W. Simpson, "A variation-based method for product family design," *Engineering Optimization*, vol. 34, no. 1, pp. 65–81, 2002.
- [2] S.-W. Hsiao, Y.-C. Ko, C.-H. Lo, and S.-H. Chen, "An ISM, DEL, and ANP based approach for product family development," *Advanced Engineering Informatics*, vol. 27, no. 1, pp. 131–148, 2013.
- [3] B. Nepal, L. Monplaisir, and O. Famuyiwa, "Matching product architecture with supply chain design," *European Journal of Operational Research*, vol. 216, no. 2, pp. 312–325, 2012.
- [4] X. Zhang and G. Q. Huang, "Game-theoretic approach to simultaneous configuration of platform products and supply chains with one manufacturing firm and multiple cooperative suppliers," *International Journal of Production Economics*, vol. 124, no. 1, pp. 121–136, 2010.
- [5] G. Du, R. J. Jiao, and M. Chen, "Joint optimization of product family configuration and scaling design by Stackelberg game," *European Journal of Operational Research*, vol. 232, no. 2, pp. 330–341, 2014.
- [6] Y. Kristianto, P. Helo, and R. J. Jiao, "Mass customization design of engineer-to-order products using Benders' decomposition and bi-level stochastic programming," *Journal of Intelligent Manufacturing*, vol. 24, no. 5, pp. 961–975, 2013.
- [7] G. Du, J. Yu, R. Y. Wu et al., "Leader-followers joint optimization of product family configuration and supply chain design," *International Journal of Knowledge, Innovation and Entrepreneurship*, vol. 1, no. 1-2, pp. 6–23, 2013.
- [8] Y. Ji, R. J. Jiao, L. Chen, and C. Wu, "Green modular design for material efficiency: a leader-follower joint optimization model," *Journal of Cleaner Production*, vol. 41, pp. 187–201, 2013.
- [9] R. J. Jiao and M. M. Tseng, "On equilibrium solutions to joint optimization problems in engineering design," *CIRP Annals—Manufacturing Technology*, vol. 62, no. 1, pp. 155–158, 2013.
- [10] G. M. Wang, Z. P. Wan, and X. J. Wang, "Bibliography on bilevel programming," *Advances in Mathematics*, vol. 36, no. 5, pp. 513–529, 2007.
- [11] W. F. Bialas and M. H. Karwan, "Two-level linear programming," *Management Science*, vol. 30, no. 8, pp. 1004–1020, 1984.
- [12] J. Fortuny-Amat and B. McCarl, "A representation and economic interpretation of a two-level programming problem," *The Journal of the Operational Research Society*, vol. 32, no. 9, pp. 783–792, 1981.
- [13] G. Anandalingam and D. J. White, "A solution method for the linear static Stackelberg problem using penalty functions," *IEEE Transactions on Automatic Control*, vol. 35, no. 10, pp. 1170–1173, 1990.
- [14] J. F. Bard, *Practical Bilevel Optimization: Algorithms and Applications*, Kluwer Academic Publishers, Amsterdam, The Netherlands, 1998.
- [15] S. Dempe, "Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints," *Optimization*, vol. 52, no. 3, pp. 333–359, 2003.
- [16] B. Liu, "Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms," *Computers & Mathematics with Applications*, vol. 36, no. 7, pp. 79–89, 1998.
- [17] V. Oduguwa and R. Roy, "Bi-level optimisation using genetic algorithm," in *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS '02)*, pp. 322–327, Divnomorskoe, Russia, 2002.
- [18] R. J. Kuo and Y. S. Han, "A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem—a case study on supply chain model," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 3905–3917, 2011.
- [19] D.-Y. Lin, A. Unnikrishnan, and S. T. Waller, "A genetic algorithm for bi-level linear programming dynamic network design problem," *Transportation Letters*, vol. 1, no. 4, pp. 281–294, 2009.
- [20] Y. Fan and H. Li, "Genetic algorithm for solving linear bilevel programming with interval coefficients," *Journal of Computer Applications*, vol. 34, no. 1, pp. 185–188, 2014.
- [21] H. Li, L. Zhang, and Y. Jiao, "Solution for integer linear bilevel programming problems using orthogonal genetic algorithm," *Journal of Systems Engineering and Electronics*, vol. 25, no. 3, pp. 443–451, 2014.

- [22] H. Li, "A genetic algorithm using a finite search space for solving nonlinear/linear fractional bilevel programming problems," *Annals of Operations Research*, vol. 235, pp. 543–558, 2015.
- [23] J. Lu, C. Shi, and G. Zhang, "On bilevel multi-follower decision making: general framework and solutions," *Information Sciences*, vol. 176, no. 11, pp. 1607–1627, 2006.
- [24] H. I. Calvete and C. Galé, "Linear bilevel multi-follower programming with independent followers," *Journal of Global Optimization*, vol. 39, no. 3, pp. 409–417, 2007.
- [25] J. S. Angelo and H. J. Barbosa, "Differential evolution to find Stackelberg-Nash equilibrium in bilevel problems with multiple followers," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC '15)*, pp. 1675–1682, IEEE, Sendai, Japan, May 2015.
- [26] G. Du, R. J. Jiao, and M. Chen, "Joint optimization of product family configuration and scaling design by Stackelberg game," *European Journal of Operational Research*, vol. 232, no. 2, pp. 330–341, 2014.
- [27] D. Wang, G. Du, R. J. Jiao, R. Wu, J. Yu, and D. Yang, "A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration," *International Journal of Production Economics*, vol. 172, pp. 1–18, 2016.
- [28] M.-Y. Li, Z.-X. Cai, and G.-R. Sun, "An adaptive genetic algorithm with diversity-guided mutation and its global convergence property," *Journal of Central South University of Technology*, vol. 11, no. 3, pp. 323–327, 2004.
- [29] B. Pavez-Lazo and J. Soto-Cartes, "A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6523–6529, 2011.
- [30] L.-Y. Zhuang, H.-B. Dong, J.-Q. Jiang, and C.-Y. Song, "A genetic algorithm using a mixed crossover strategy," *Lecture Notes in Computer Science*, vol. 5263, no. 1, pp. 854–863, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

