

Research Article

An Improved Particle Swarm Optimization for Solving Bilevel Multiobjective Programming Problem

Tao Zhang,^{1,2} Tiesong Hu,¹ Yue Zheng,³ and Xuning Guo¹

¹ State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan 430072, China

² School of Information and Mathematics, Yangtze University, Jingzhou 434023, China

³ College of Mathematics and Computer Sciences, Huanggang Normal University, Huanggang 438000, China

Correspondence should be addressed to Tao Zhang, zt.sx981@126.com

Received 4 December 2011; Revised 21 January 2012; Accepted 5 February 2012

Academic Editor: Debasish Roy

Copyright © 2012 Tao Zhang 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.

An improved particle swarm optimization (PSO) algorithm is proposed for solving bilevel multiobjective programming problem (BLMPP). For such problems, the proposed algorithm directly simulates the decision process of bilevel programming, which is different from most traditional algorithms designed for specific versions or based on specific assumptions. The BLMPP is transformed to solve multiobjective optimization problems in the upper level and the lower level interactively by an improved PSO. And a set of approximate Pareto optimal solutions for BLMPP is obtained using the elite strategy. This interactive procedure is repeated until the accurate Pareto optimal solutions of the original problem are found. Finally, some numerical examples are given to illustrate the feasibility of the proposed algorithm.

1. Introduction

Bilevel programming problem (BLPP) arises in a wide variety of scientific and engineering applications including optimal control, process optimization, game-playing strategy development, and transportation problem. Thus, the BLPP has been developed and researched by many scholars. The reviews, monographs, and surveys on the BLPP can refer to [1–11]. Moreover, the evolutionary algorithms (EA) have been employed to address BLPP in papers [12–16].

However, the bilevel multiobjective programming problem (BLMPP) has seldom been studied. Shi and Xia [17, 18], Abo-Sinna and Baky [19], Nishizaki and Sakawa [20], and

Zheng et al. [21] presented an interactive algorithm for BLMPP. Eichfelder [22] presented a method for solving nonlinear bilevel multiobjective optimization problems with coupled upper level constraints. Thereafter, Eichfelder [23] developed a numerical method for solving nonlinear nonconvex bilevel multiobjective optimization problems. In recent years, the metaheuristic has attracted considerable attention as an alternative method for BLMPP. For example, Deb and Sinha [24–26] as well as Sinha and Deb [27] discussed BLMPP based on evolutionary multiobjective optimization principles. Based on those studies, Deb and Sinha [28] proposed a viable and hybrid evolutionary-local-search-based algorithm and presented challenging test problems. Sinha [29] presented a progressively interactive evolutionary multiobjective optimization method for BLMPP.

Particle swarm optimization (PSO) is a relatively novel heuristic algorithm inspired by the choreography of a bird flock, which has been found to be quite successful in a wide variety of optimization tasks [30]. Due to its high speed of convergence and relative simplicity, the PSO algorithm has been employed by many researchers for solving BLPPs. For example, Li et al. [31] proposed a hierarchical PSO for solving BLPP. Kuo and Huang [32] applied the PSO algorithm for solving bilevel linear programming problem. Gao et al. [33] presented a method to solve bilevel pricing problems in supply chains using PSO. However, it is worth noting that the papers mentioned above are only for bilevel single objective problems.

In this paper, an improved PSO is presented for solving BLMPP. The algorithm can be outlined as follows. The BLMPP is transformed to solve multiobjective optimization problems in the upper level and the lower level interactively by an improved PSO. And a set of approximate Pareto optimal solutions for BLMPP is obtained using the elite strategy. The above interactive procedure is repeated for a predefined count, and then the accurate Pareto optimal solutions of the BLMPP will be achieved. Towards these ends, the rest of the paper is organized as follows. In Section 2, the problem formulation is provided. The proposed algorithm for solving bilevel multiobjective problem is presented in Section 3. In Section 4, some numerical examples are given to demonstrate the proposed algorithm, while the conclusion is reached in Section 5.

2. Problem Formulation

Let $x \in R^{n_1}$, $y \in R^{n_2}$, $F : R^{n_1} \times R^{n_2} \rightarrow R^{m_1}$, $f : R^{n_1} \times R^{n_2} \rightarrow R^{m_2}$, $G : R^{n_1} \times R^{n_2} \rightarrow R^p$, and $g : R^{n_1} \times R^{n_2} \rightarrow R^q$. The general model of the BLMPP can be written as follows:

$$\begin{aligned} & \min_x F(x, y) \\ & \text{s.t. } G(x, y) \geq 0, \\ & \min_y f(x, y) \\ & \text{s.t. } g(x, y) \geq 0, \end{aligned} \tag{2.1}$$

where $F(x, y)$ and $f(x, y)$ are the upper level and the lower level objective functions, respectively. $G(x, y)$ and $g(x, y)$ denote the upper level and the lower level constraints, respectively. Let $S = \{(x, y) | G(x, y) \geq 0, g(x, y) \geq 0\}$, $X = \{x | \exists y, G(x, y) \geq 0, g(x, y) \geq 0\}$, $S(x) = \{y | g(x, y) \geq 0\}$, and for the fixed $x \in X$, let $\bar{S}(X)$ denote the weak efficiency set of solutions to the lower level problem, the feasible solution set of problem (2.1) is denoted as $IR = \{(x, y) | (x, y) \in S, y \in \bar{S}(X)\}$.

Definition 2.1. For a fixed $x \in X$, if y is a Pareto optimal solution to the lower level problem, then (x, y) is a feasible solution to the problem (2.1).

Definition 2.2. If (x^*, y^*) is a feasible solution to the problem (2.1) and there are no $(x, y) \in \mathbb{R}$, such that $F(x, y) < F(x^*, y^*)$, then (x^*, y^*) is a Pareto optimal solution to the problem (2.1), where " $<$ " denotes Pareto preference.

For problem (2.1), it is noted that a solution (x^*, y^*) is feasible for the upper level problem if and only if y^* is an optimal solution for the lower level problem with $x = x^*$. In practice, we often make the approximate Pareto optimal solutions of the lower level problem as the optimal response feedback to the upper level problem, and this point of view is accepted usually. Based on this fact, the PSO algorithm may have a great potential for solving BLMPP. On the other hand, unlike the traditional point-by-point approach mentioned in Section 1, the PSO algorithm uses a group of points in its operation thus, the PSO can be developed as a new way for solving BLMPP. In the following, we present an improved PSO algorithm for solving problem (2.1).

3. The Algorithm

The process of the proposed algorithm is an interactive coevolutionary process for both the upper level and the lower level. We first initialize population and then solve multiobjective optimization problems in the upper level and the lower level interactively using an improved PSO. Afterwards, a set of approximate Pareto optimal solutions for problem 1 is obtained by the elite strategy which was adopted in Deb et al. [34]. This interactive procedure is repeated until the accurate Pareto optimal solutions of problem (2.1) are found. The details of the proposed algorithm are given as follows:

3.1. Algorithm

Step 1. Initialize.

Substep 1.1. Initialize the population P_0 with N_u particles which is composed by $n_s = N_u/N_l$ subswarms of size N_l each. The particle's position of the k th ($k = 1, 2, \dots, n_s$) subswarm is presented as $z_j = (x_j, y_j)$ ($j = 1, 2, \dots, n_l$), and the corresponding velocity is presented as: $v_j = (v_{x_j}, v_{y_j})$ ($j = 1, 2, \dots, n_l$), z_j and v_j are sampled randomly in the feasible space, respectively.

Substep 1.2. Initialize the external loop counter $t := 0$.

Step 2. For the k th subswarm ($k = 1, 2, \dots, n_s$), each particle is assigned a nondomination rank ND_l and a crowding value CD_l in f space. Then, all resulting subswarms are combined into one population which is named as the P_t . Afterwards, each particle is assigned a nondomination rank ND_u and a crowding value CD_u in F space.

Step 3. The nondomination particles assigned both $ND_u = 1$ and $ND_l = 1$ from P_t are saved in the elite set A_t .

Step 4. For the k th subswarm ($k = 1, 2, \dots, n_s$), update the lower level decision variables.

Substep 4.1. Initialize the lower level loop counter $t_l := 0$.

Substep 4.2. Update the j th ($j = 1, 2, \dots, N_l$) particle's position and velocity with the fixed x_j and the fixed v_j using

$$\begin{aligned} v_{y_j}^{t_l+1} &= \omega_l v_{y_j}^{t_l} + c_{1l} r_{1l} (p_{y_j}^{pbest} - z_j^{t_l}) + c_{2l} r_{2l} (p_l^{gbest} - z_j^{t_l}), \\ z_j^{t_l+1} &= z_j^{t_l} + v_{y_j}^{t_l+1}. \end{aligned} \quad (3.1)$$

Substep 4.3. Consider $t_l := t_l + 1$.

Substep 4.4. If $t_l \geq T_l$, go to Substep 4.5. Otherwise, go to Substep 4.2.

Substep 4.5. Each particle of the i th subswarm is reassigned a nondomination rank ND_l and a crowding value CD_l in F space. Then, all resulting subswarms are combined into one population which is renamed as the Q_t . Afterwards, each particle is reassigned a nondomination rank ND_u and a crowding value CD_u in F space.

Step 5. Combine population P_t and Q_t to form R_t . The combined population R_t is reassigned a nondomination rank ND_u , and the particles within an identical nondomination rank are assigned a crowding distance value CD_u in the F space.

Step 6. Choose half particles from R_t . The particles of rank $ND_u = 1$ are considered first. From the particles of rank $ND_u = 1$, the particles with $ND_l = 1$ are noted one by one in the order of reducing crowding distance CD_u , for each such particle the corresponding subswarm from its source population (either P_t or Q_t) is copied in an intermediate population S_t . If a subswarm is already copied in S_t and a future particle from the same subswarm is found to have $ND_u = ND_l = 1$, the subswarm is not copied again. When all particles of $ND_u = 1$ are considered, a similar consideration is continued with $ND_u = 2$ and so on till exactly n_s subswarms are copied in S_t .

Step 7. Update the elite set A_t . The nondomination particles assigned both $ND_u = 1$ and $ND_l = 1$ from S_t are saved in the elite set A_t .

Step 8. Update the upper level decision variables in S_t .

Substep 8.1. Initiate the upper level loop counter $t_u := 0$.

Substep 8.2. Update the i th ($i = 1, 2, \dots, N_u$) particle's position and velocity with the fixed y_i and the fixed v_i using

$$\begin{aligned} v_{x_i}^{t_u+1} &= \omega_u v_{x_i}^{t_u} + c_{1u} r_{1u} (p_{x_i}^{pbest} - z_i^{t_u}) + c_{2u} r_{2u} (p_u^{gbest} - z_i^{t_u}), \\ z_i^{t_u+1} &= z_i^{t_u} + v_{x_i}^{t_u+1}. \end{aligned} \quad (3.2)$$

Substep 8.3. Consider $t_u := t_u + 1$.

Substep 8.4. If $t_u \geq T_u$, go to Substep 8.5. Otherwise, go to Substep 8.2.

Substep 8.5. Every member is then assigned a nondomination rank ND_u and a crowding distance value CD_u in F space.

Step 9. Consider $t := t + 1$.

Step 10. If $t \geq T$, output the elite set A_t . Otherwise, go to Step 2.

In Steps 4 and 8, the global best position is chosen at random from the elite set A_t . The criterion of personal best position choice is that if the current position is dominated by the previous position, then the previous position is kept; otherwise, the current position replaces the previous one; if neither of them is dominated by the other, then we select one of them randomly. A relatively simple scheme is used to handle constraints. Whenever two individuals are compared, their constraints are checked. If both are feasible, nondomination sorting technology is directly applied to decide which one is selected. If one is feasible and the other is infeasible, the feasible dominates. If both are infeasible, then the one with the lowest amount of constraint violation dominates the other. Notations used in the proposed algorithm are detailed in Table 1.

4. Numerical Examples

In this section, three examples will be considered to illustrate the feasibility of the proposed algorithm for problem (2.1). In order to evaluate the closeness between the obtained Pareto optimal front and the theoretical Pareto optimal front, as well as the diversity of the obtained Pareto optimal solutions along the theoretical Pareto optimal front, we adopted the following evaluation metrics.

4.1. Generational Distance (GD)

This metric used by Deb [35] is employed in this paper as a way of evaluating the closeness between the obtained Pareto optimal front and the theoretical Pareto optimal front. The GD metric denotes the average distance between the obtained Pareto optimal front and the theoretical Pareto optimal front:

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad (4.1)$$

where n is the number of the obtained Pareto optimal solutions by the proposed algorithm and d_i is the Euclidean distance between each obtained Pareto optimal solution and the nearest member of the theoretical Pareto optimal set.

Table 1: The notations of the algorithm.

x_i	The i th particle's position of the upper level problem.
v_{x_i}	The velocity of x_i .
y_j	The j th particle's position of the lower level problem.
v_{y_j}	The velocity of y_j .
z_j	The j th particle's position of BLMPP.
$p_{y_j}^{pbest}$	The j th particle's personal best position for the lower level problem.
$p_{x_i}^{pbest}$	The i th particle's personal best position for the upper level problem.
p_l^{gbest}	The particle's global best position for the lower level problem.
p_u^{gbest}	The particle's global best position for the upper level problem.
N_u	The population size of the upper level problem.
N_l	The subswarm size of the lower level problem.
t	Current iteration number for the overall problem.
T	The predefined max iteration number for t .
t_u	Current iteration number for the upper level problem.
t_l	Current iteration number for the lower level problem.
T_u	The predefined max iteration number for t_u .
T_l	The predefined max iteration number for t_l .
w_u	Inertia weights for the upper level problem.
w_l	Inertia weights the lower level problem.
c_{1u}	The cognitive learning rate for the upper level problem.
c_{2u}	The social learning rate for the upper level problem.
c_{1l}	The cognitive learning rate for the lower level problem.
c_{2l}	The social learning rate for the lower level problem.
ND_u	Nondomination sorting rank of the upper level problem.
CD_u	Crowding distance value of the upper level problem.
ND_l	Nondomination sorting rank of the lower level problem.
CD_l	Crowding distance value of the lower level problem.
P_t	The t th iteration population.
Q_t	The offspring of P_t .
S_t	Intermediate population.

4.2. Spacing (SP)

This metric is used to evaluate the diversity of the obtained Pareto optimal solutions by comparing the uniform distribution and the deviation of solutions as described by Deb [35]:

$$SP = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^n (\bar{d} - d_i)^2}{\sum_{m=1}^M d_m^e + n\bar{d}}, \quad (4.2)$$

where $d_i = \min_j (|F_1^i(x, y) - F_1^j(x, y)| + |F_2^i(x, y) - F_2^j(x, y)|)$, $i, j = 1, 2, \dots, n$, \bar{d} is the mean of all d_i , d_m^e is the Euclidean distance between the extreme solutions in obtained Pareto optimal solution set and the theoretical Pareto optimal solution set on the m th objective, M is the

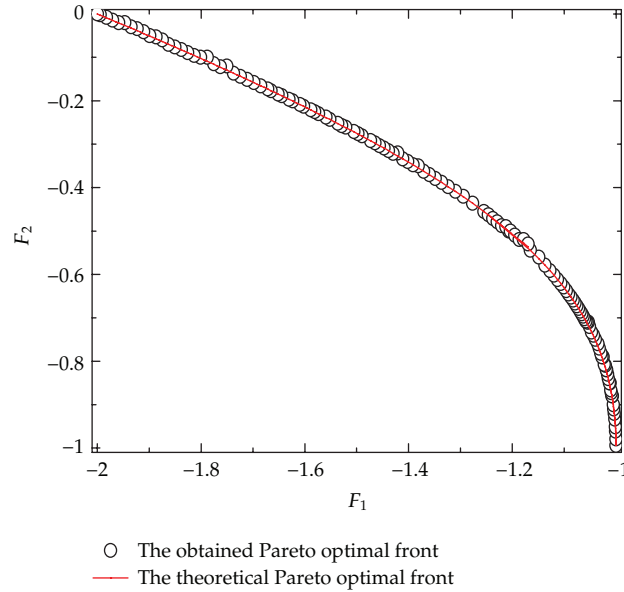


Figure 1: The obtained Pareto optimal front of Example 4.1.

number of the upper level objective function, n is the number of the obtained solutions by the proposed algorithm.

The PSO parameters are set as follows: $r_{1u}, r_{2u}, r_{1l}, r_{2l} \in \text{random}(0, 1)$, the inertia weight $w_u = w_l = 0.7298$, and acceleration coefficients with $c_{1u} = c_{2u} = c_{1l} = c_{2l} = 1.49618$. All results presented in this paper have been obtained on a personal computer (CPU: AMD Phenom II X6 1055T 2.80 GHz; RAM: 3.25 GB) using a C# implementation of the proposed algorithm, and the figures were obtained using the origin 8.0.

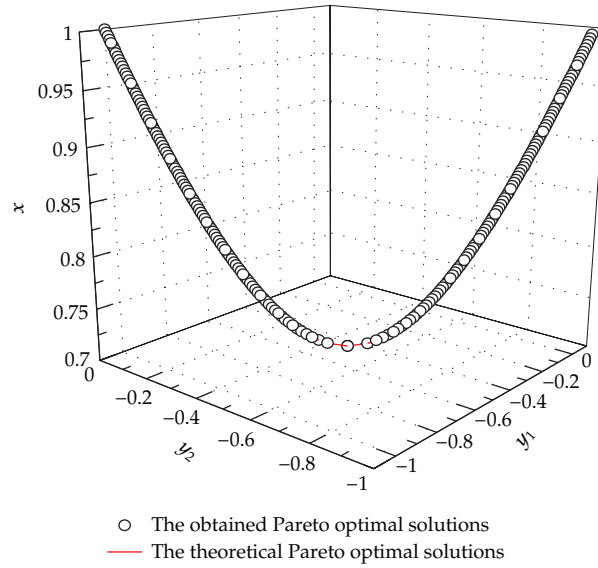
Example 4.1. Example 4.1 is taken from [22]. Here $x \in R^1$, $y \in R^2$. In this example, the population size and iteration times are set as follows: $N_u = 200$, $T_u = 200$, $N_l = 40$, $T_l = 40$, and $T = 40$:

$$\begin{aligned}
 \min_x \quad & F(x, y) = (y_1 - x, y_2) \\
 \text{s.t.} \quad & G_1(y) = 1 + y_1 + y_2 \geq 0 \\
 \min_y \quad & f(x, y) = (y_1, y_2) \\
 \text{s.t.} \quad & g_1(x, y) = x^2 - y_1^2 - y_2^2 \geq 0, \\
 & -1 \leq y_1, \quad y_2 \leq 1, \quad 0 \leq x \leq 1.
 \end{aligned} \tag{4.3}$$

Figure 1 shows the obtained Pareto front of this example by the proposed algorithm. From Figure 1, it can be seen that the obtained Pareto front is very close to the theoretical Pareto optimal front, and the average distance between the obtained Pareto optimal front and the theoretical Pareto optimal front is 0.00026, that is, $GD = 0.00026$ (see Table 2). Moreover, the lower SP value ($SP = 0.17569$, see Table 2) shows that the proposed algorithm is able to obtain

Table 2: Results of the Generation Distance (GD) and Spacing (SP) metrics for Examples 4.1 and 4.2.

Example	GD	SP
Example 4.1	0.00026	0.17569
Example 4.2	0.00004	0.00173

**Figure 2:** The obtained solutions of Example 4.1.

a good distribution of solutions on the entire range of the theoretical Pareto optimal front. Figure 2 shows the obtained solutions of this example, which follow the relationship, that is, $y_1 = -1 - y_2$, $y_2 = -1/2 \pm (1/4)\sqrt{8x^2 - 4}$ and $x \in (1/\sqrt{2}, 1)$. It is also obvious that all obtained solutions are close to being on the upper level constraint $G(x)$ boundary ($1 + y_1 + y_2 = 0$).

Example 4.2. Example 4.2 is taken from [36]. Here $x \in R^1$, $y \in R^2$. In this example, the population size and iteration times are set as follows: $N_u = 200$, $T_u = 50$, $N_l = 40$, $T_l = 20$, and $T = 40$.

$$\begin{aligned}
 \min_x F(x, y) &= (x^2 + (y_1 - 1)^2 + y_2^2, (x - 1)^2 + (y_1 - 1)^2 + y_2^2), \\
 \min_y f(x, y) &= (y_1^2 + y_2^2, (y_1 - x)^2 + y_2^2), \\
 & -1 \leq x, \quad y_1, y_2 \leq 2.
 \end{aligned} \tag{4.4}$$

Figure 3 shows the obtained Pareto optimal front of this example by the proposed algorithm. From Figure 3, it is obvious that the obtained Pareto optimal front is very close to the theoretical Pareto optimal front, the average distance between the obtained Pareto optimal front and the theoretical Pareto optimal front is 0.00004 (see Table 2). On the other hand, the

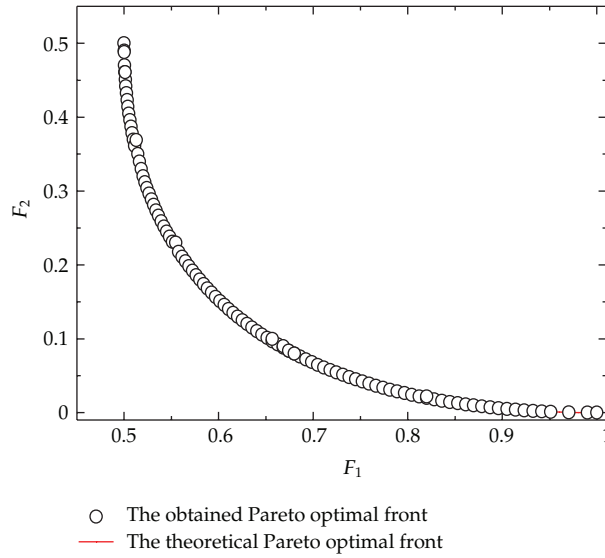


Figure 3: The obtained Pareto optimal front of Example 4.2.

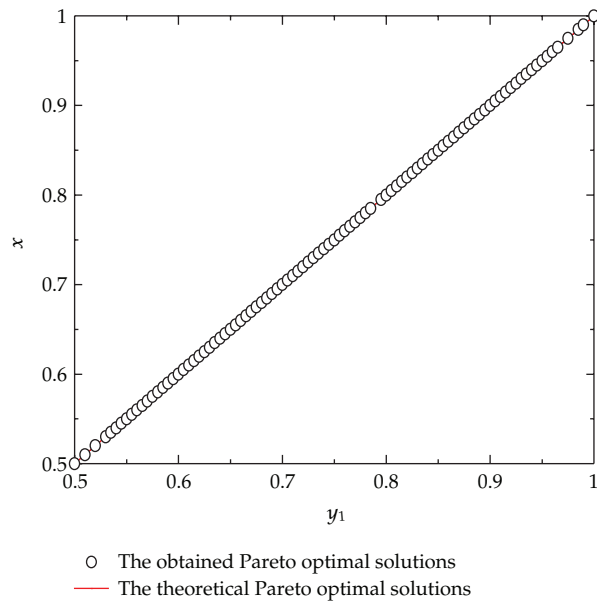


Figure 4: The obtained solutions of Example 4.2.

obtained Pareto optimal solutions can be distributed uniformly on entire range of theoretical Pareto optimal front based on the fact that the SP value is lower ($SP = 0.00173$, see Table 2). Figure 4 shows the obtained Pareto optimal solutions; they follow the relationship, that is, $x = y_1, y_1 \in [0.5, 1]$ and $y_2 = 0$.

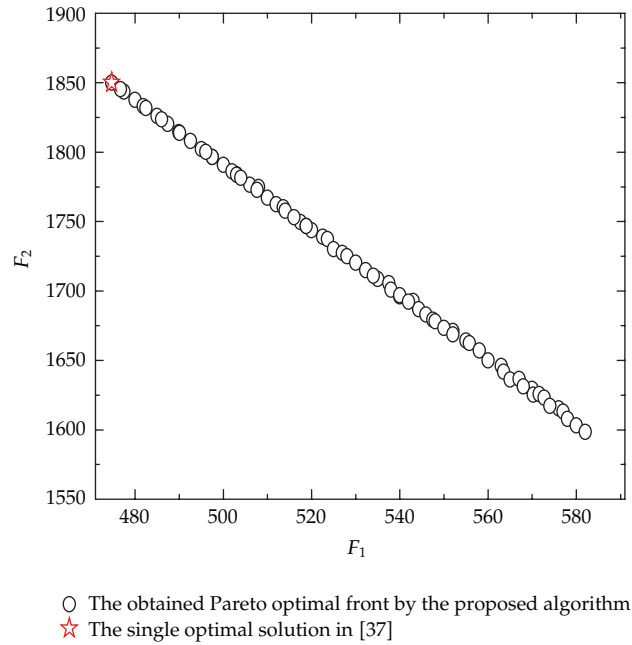


Figure 5: The obtained Pareto optimal front of Example 4.3.

Example 4.3. Example 4.3 is taken from [37], in which the theoretical Pareto optimal front is not given. Here $x \in R^2$, $y \in R^3$. In this example, the population size and iteration times are set as follows: $N_u = 100$, $T_u = 50$, $N_l = 20$, $T_l = 10$, and $T = 40$:

$$\begin{aligned}
 & \max_x F(x, y) = (x_1 + 9x_2 + 10y_1 + y_2 + 3x_3, 9x_1 + 2x_2 + 2y_1 + 7y_2 + 4x_3), \\
 & \text{s.t. } G_1(x, y) = 3x_1 + 9x_2 + 9y_1 + 5y_2 + 3y_3 \leq 1039, \\
 & \quad G_2(x, y) = -4x_1 - x_2 + 3y_1 - 3y_2 + 2y_3 \leq 94, \\
 & \min_y f(x, y) = (4x_1 + 6x_2 + 7y_1 + 4y_2 + 8y_3, 6x_1 + 4x_2 + 8y_1 + 7y_2 + 4y_3), \\
 & \text{s.t. } g_1(x, y) = 3x_1 - 9x_2 - 9y_1 - 4y_2 \leq 61, \\
 & \quad g_2(x, y) = 5x_1 + 9x_2 + 10y_1 - y_2 - 2y_3 \leq 924, \\
 & \quad g_3(x, y) = 3x_1 - 3x_2 + y_2 + 5y_3 \leq 420, \\
 & \quad x_1, x_2, y_1, y_2, y_3 \geq 0.
 \end{aligned} \tag{4.5}$$

Figure 5 shows the obtained Pareto optimal front of Example 4.3 by the proposed algorithm. Figure 6 shows all five constrains for all obtained Pareto optimal solutions and it can be seen that the G_1 , g_2 and g_3 are active constrains. Note that, Zhang et al. [37] only obtained a single optimal solution $x = (146.2955, 28.9394)$, and $y = (0, 67.9318, 0)$ which lies on the maximum of the F_2 using weighted sum method. In contrast, a set of Pareto optimal solutions is obtained by the proposed algorithm. However, the fact that the single optimal solution in [37] is included in the obtained Pareto optimal solutions illustrates the feasibility of proposed algorithm.

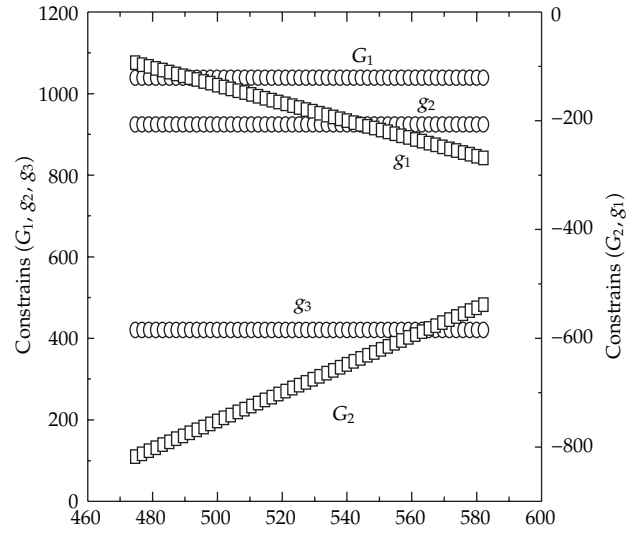


Figure 6: The constrains of Example 4.3.

5. Conclusion

In this paper, an improved PSO is presented for BLMPP. The BLMPP is transformed to solve the multiobjective optimization problems in the upper level and the lower level interactively using the proposed algorithm for a predefined count. And a set of accurate Pareto optimal solutions for BLMPP is obtained by the elite strategy. The experimental results illustrate that the obtained Pareto front by the proposed algorithm is very close to the theoretical Pareto optimal front, and the solutions are also distributed uniformly on entire range of the theoretical Pareto optimal front. Furthermore, the proposed algorithm is simple and easy to implement. It also provides another appealing method for further study on BLMPP.

Acknowledgment

This work is supported by the National Science Foundation of China (71171151, 50979073).

References

- [1] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, pp. 235–256, 2007.
- [2] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: a bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, 1994.
- [3] J. F. Bard, *Practical Bilevel Optimization: Algorithms and Applications*, vol. 30, Kluwer Academic, Dordrecht, The Netherlands, 1998.
- [4] S. Dempe, *Foundations of Bilevel Programming*, vol. 61 of *Nonconvex Optimization and its Applications*, Kluwer Academic, Dordrecht, The Netherlands, 2002.
- [5] S. Dempe, "Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints," *Optimization*, vol. 52, no. 3, pp. 333–359, 2003.
- [6] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK, 1996.

- [7] K. Shimizu, Y. Ishizuka, and J. F. Bard, *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic, Dordrecht, The Netherlands, 1997.
- [8] B. Colson, P. Marcotte, and G. Savard, "Bilevel programming: a survey," *Quarterly Journal of Operations Research*, vol. 3, no. 2, pp. 87–107, 2005.
- [9] 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 (Chinese).
- [10] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: a bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, 1994.
- [11] U. P. Wen and S. T. Hsu, "Linear bi-level programming problems. A review," *Journal of the Operational Research Society*, vol. 42, no. 2, pp. 125–133, 1991.
- [12] A. Koh, "Solving transportation bi-level programs with differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 2243–2250, September 2007.
- [13] R. Mathieu, L. Pittard, and G. Anandalingam, "Genetic algorithm based approach to bilevel linear programming," *RAIRO Recherche Opérationnelle*, vol. 28, no. 1, pp. 1–21, 1994.
- [14] V. Oduguwa and R. Roy, "Bi-level optimization using genetic algorithm," in *Proceedings of the IEEE International Conference on Artificial Intelligence Systems*, pp. 322–327, 2002.
- [15] Y. Wang, Y. C. Jiao, and H. Li, "An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 35, no. 2, pp. 221–232, 2005.
- [16] Y. Yin, "Genetic-algorithms-based approach for bilevel programming models," *Journal of Transportation Engineering*, vol. 126, no. 2, pp. 115–119, 2000.
- [17] X. Shi and H. Xia, "Interactive bilevel multi-objective decision making," *Journal of the Operational Research Society*, vol. 48, no. 9, pp. 943–949, 1997.
- [18] X. Shi and H. Xia, "Model and interactive algorithm of bi-level multi-objective decision-making with multiple interconnected decision makers," *Journal of Multi-Criteria Decision Analysis*, vol. 10, pp. 27–34, 2001.
- [19] M. A. Abo-Sinna and I. A. Baky, "Interactive balance space approach for solving multi-level multi-objective programming problems," *Information Sciences*, vol. 177, no. 16, pp. 3397–3410, 2007.
- [20] I. Nishizaki and M. Sakawa, "Stackelberg solutions to multiobjective two-level linear programming problems," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 161–182, 1999.
- [21] Y. Zheng, Z. Wan, and G. Wang, "A fuzzy interactive method for a class of bilevel multiobjective programming problem," *Expert Systems with Applications*, vol. 38, pp. 10384–10388, 2011.
- [22] G. Eichfelder, "Solving nonlinear multi-objective bi-level optimization problems with coupled upper level constraints," Technical Report 320, Institute of Applied Mathematics, University Erlangen-Nrnberg, Germany, 2007.
- [23] G. Eichfelder, "Multiobjective bilevel optimization," *Mathematical Programming. A Publication of the Mathematical Programming Society*, vol. 123, no. 2, pp. 419–449, 2010.
- [24] K. Deb and A. Sinha, "Constructing test problems for bilevel evolutionary multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 1153–1160, May 2009.
- [25] K. Deb and A. Sinha, "Solving bilevel multiobjective optimization problems using evolutionary algorithms," in *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO '09)*, vol. 5467 of *Lecture Notes in Computer Science*, pp. 110–124, 2009.
- [26] K. Deb and A. Sinha, "An evolutionary approach for bilevel multiobjective problems," in *Cutting-Edge Research Topics on Multiple Criteria Decision Making*, vol. 35 of *Communications in Computer and Information Science*, pp. 17–24, Berlin, Germany, 2009.
- [27] A. Sinha and K. Deb, "Towards understanding evolutionary bilevel multiobjective optimization algorithm," Technical Report 2008006, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur, India, 2008.
- [28] K. Deb and A. Sinha, "An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary local-search algorithm," Technical Report 2009001, 2009.
- [29] A. Sinha, "Bilevel multi-objective optimization problem solving using progressively interactive EMO," in *Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization (EMO '11)*, vol. 6576, pp. 269–284, 2011.
- [30] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

- [31] X. Li, P. Tian, and X. Min, "A hierarchical particle swarm optimization for solving bilevel programming problems," in *Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC '06)*, vol. 4029 of *Lecture Notes in Computer Science*, pp. 1169–1178, 2006.
- [32] R. J. Kuo and C. C. Huang, "Application of particle swarm optimization algorithm for solving bi-level linear programming problem," *Computers & Mathematics with Applications*, vol. 58, no. 4, pp. 678–685, 2009.
- [33] Y. Gao, G. Zhang, J. Lu, and H. M. Wee, "Particle swarm optimization for bi-level pricing problems in supply chains," *Journal of Global Optimization*, vol. 51, pp. 245–254, 2011.
- [34] 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.
- [35] K. Deb, "Multi-objective optimization using evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.
- [36] K. Deb and A. Sinha, "Solving bilevel multi-objective optimization problems using evolutionary algorithms," Kangal Report, 2008.
- [37] G. Zhang, J. Liu, and T. Dillon, "Decentralized multi-objective bilevel decision making with fuzzy demands," *Knowledge-Based Systems*, vol. 20, pp. 495–507, 2007.