Scientia Iranica E (2023) 30(2), 727–737

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Designing of a mat-heuristic algorithm for solving bi-level optimization problems

H. Shams Shemirani, R. Sahraeian*, and M. Bashiri

Department of Industrial Engineering, Faculty of Engineering, Shahed University, Tehran, Iran.

Received 1 January 2020; received in revised form 26 December 2020; accepted 13 January 2021

KEYWORDS
Bi-level optimization;
Mat-heuristic algorithm;
Meta-heuristic methods;
Evolution-based methods;
Bi-level programming.

Abstract. In this paper, a new algorithm for solving bi-level optimization problems is presented. This algorithm can obtain an optimal or near-optimal solution for any bi-level optimization problem. The decision variables of the first- and second-level models can be both integers and continuous. In this method, by solving a certain number of the bi-objective programming models and, then, solving the corresponding second-level model, a bi-level feasible solution that is either optimal or near-optimal is identified. To evaluate the efficiency of the algorithm, the value of the objective function and its computation time in different instances are compared with those associated with exact methods as well as evolution-based methods. The numerical results confirm the high efficiency of the proposed algorithm.

1. Introduction

The bi-level programming is a problem that involves two decision-makers. The first decision-maker is called 'leader' and the second decision-maker is called 'follower'. Some of the variables are identified by the leader and, then, the others are identified by the follower. The objective function and the leader's constraints may include the decision variables of the follower, and vice versa. In the bi-level programming process, the leader first determines the values of its decision variables. Then, the follower determines the values of its decision variables by solving a mathematical model considering the leader's decisions. The process of implementing the bi-level optimization problems is as follows:

Step 1: The leader considers values for first-level decision variables;

Step 2: The follower receives the values of the leader's variables and inserts them as parameters into the second-level model;

Step 3: The follower solves the second-level mathematical model and determines the optimal values for the second-level decision variables;

Step 4: The leader receives the values of the follower's variables;

Step 5: The leader calculates the value of the first-level objective function;

Step 6: The leader will check whether the first-level constraints are in place;

Step 7: The leader wants to determine the values of the first-level decision variables in such a way that due to the follower's reaction, the first-level constraints are satisfied and the first-level objective function is optimized.

*. *Corresponding author.*
E-mail addresses: shams@gut.ac.ir (H. Shams Shemirani); Sahraeian@shahed.ac.ir (R. Sahraeian); Bashiri@shahed.ac.ir (M. Bashiri)

doi: 10.24200/sci.2021.55147.4093

The bi-level optimization problems are expressed in general terms using mathematical and vector symbols in the following form:

$$\max_{X} : f_1(X,Y), \tag{1}$$

s.t.:

$$g_1(X,Y) \leq 0, \tag{2}$$

$$\max_{Y} : f_2(X,Y), \tag{3}$$

s.t.:

$$g_2(X,Y) \leq 0. \tag{4}$$

The following definitions are used in the above equations:

X	Vector of decision variables for the leader
Y	Vector of decision variables for the follower
$f_1(X,Y)$	Leader objective function
$g_1(X,Y) \leq 0$	Leader constraints set
$f_2(X,Y)$	Follower objective function
$g_2(X,Y) \leq 0$	Follower constraints set

Bi-level programming problems can be converted into an MIP problem. The problems simply involve linear mathematical models for the leader and the follower and their decision variables are continuous. In this case, using the Karush-Kuhn-Tucker (KKT) conditions, we can transform the follower model into a series of constraints. Then, by placing them into the constraints of the leader, a nonlinear model is created. This non-linearity is due to the constraints of the complementary slackness. By using binary variables, these terms can be linearized to create a (Mixed Integer Programming) MIP model. However, this model can only be solved in small sizes, and as the number of variables and constraints of followers increases, the number of binary variables will increase, too. Computational experience has indicated the need for longer computational time.

In a state where possible values of the leader's decision variables are discrete and the number of possible states for them is not high, we can identify an optimal solution using the complete enumeration method. If the number of possible solutions is high for the leader, then the computational time of this method will be very long and time-consuming. How to utilize the complete enumeration method and KKT and MIP method is described in the section of the general formulation.

Given that there has not been an exact method that can identify the optimal solution for a variety of bi-level optimization problems in an acceptable time, designing a method that can identify optimal or near-optimal solutions in a short time is crucial. In this paper, using fuzzy normalization, goal programming, and concepts of bi-level optimization, we introduce an algorithm that can find optimal or near-optimal solutions to a variety of bi-level optimization problems. In these problems, models of leader and follower can be solved using the following methods: LP (Linear Programming), NLP (Non-Linear Programming), MIP (Mixed Integer Programming), and MINLP (Mixed Integer Non-Linear Programming) in a relatively short time. Considering the mechanism of this method, we named it GPBLO (Goal Programming for Bi-level Optimization).

The rest of this paper is organized as follows. Section 2 discusses an overview of the published papers on various applications of bi-level programming, as well as the various methods raised for solving them. Section 3 presents a general formulation of bi-level optimization problems and the method of using the KKT conditions as well as the complete enumeration for the applicable cases. Section 4 introduces the proposed solution method, which is a mat-heuristic algorithm. Section 5 analyzes computational experiments conducted on 20 sample problems in different sizes that can be solved using the KKT method and, also, on 20 sample problems that are solved by complete enumeration method. These problems are also solved using the proposed method, and the results are compared in terms of the objective function and calculation time. The final section provides conclusions and future research directions.

2. Related literature

Many researchers have studied the application of bi-level programming to solve problems related to industries, services, transportation, military operations, and so on. The following can be mentioned:

1. Facility location problems [1-3];
2. Supply chain management [4-6];
3. Missile defense design [7];
4. Environmental pollution control policies [8];
5. Energy sources planning [9];
6. Operations scheduling [10];
7. Transportation systems [11];
8. Physical human-robot interaction [12];
9. Tomographic reconstruction [13].

Solving the bi-level optimization problems, even in its simplest form, is strongly NP-hard according to the methods known so far. Even the study of whether a given solution is optimal or not is also NP-hard [14]. For this reason, many researchers have tried to provide suitable procedures for such problems in an acceptable time using evolutionary algorithms. In [14], classical and evolutionary approaches proposed to solve bi-level programming problems by 2018 are under review and some uses for bi-level programming problems are raised. Several studies have been done to solve linear and nonlinear bi-level optimization problems using evolutionary methods such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [15–21]. The use of evolutionary methods for solving single-level and bi-level programming problems may offer optimal or near-optimal solutions. However, the application of these methods to solve bi-level problems requires, unlike the single-level problems, heavy and prolonged computations.

Several studies have also been carried out to solve linear and nonlinear bi-level optimization problems using such methods as branch and bound, branch and cut, Benders decomposition, and KKT conditions [22–26]. These methods are also commonly developed for specific models and cannot be used for solving a variety of bi-level problems, or they require a long computation time.

Considering the wide range of bi-level programming applications and that an algorithm is not able to achieve an optimal or near-optimal solution in an acceptable time, identification of heuristic and mat-heuristic methods that can bring us closer to this request is crucial. This study presents a fast and flexible mat-heuristic method that can obtain optimal or near-optimal solutions to various types of bi-level optimization problems in a relatively short time.

3. The general formulation

All bi-level optimization problems are expressed using vector symbols by Eqs. (1)-(4). If the mathematical relations in the problem are linear, the general formulation of the problem in the form of a matrix can be stated as follows:

$$\max_X : f_1 = C_1X + D_1Y, \tag{5}$$

$$A_1X + B_1Y \leq G_1, \tag{6}$$

$$X \geq 0, \tag{7}$$

$$\max_Y : f_2 = C_2X + D_2Y, \tag{8}$$

$$A_2X + B_2Y \leq G_2, \tag{9}$$

$$Y \geq 0. \tag{10}$$

Moreover, the bi-level linear programming problems in general and in algebraic form can be demonstrated as follows:

$$\max_X : f_1 = \sum_{j_1} C_1(j_1)X(j_1) + \sum_{j_2} D_1(j_2)Y(j_2). \tag{11}$$

s.t.:

$$\sum_{j_1} A_1(i_1, j_1)X(j_1) + \sum_{j_2} B_1(i_1, j_2)Y(j_2) \leq G_1(i_1) \qquad \forall i_1, \tag{12}$$

$$X(j_1) \geq 0 \qquad \forall j_1, \tag{13}$$

$$\max_Y : f_2 = \sum_{j_1} C_2(j_1)X(j_1) + \sum_{j_2} D_2(j_2)Y(j_2), \tag{14}$$

s.t.:

$$\sum_{j_1} A_2(i_2, j_1)X(j_1) + \sum_{j_2} B_2(i_2, j_2)Y(j_2) \leq G_2(i_2) \quad \forall i_2, \tag{15}$$

$$Y(j_2) \geq 0 \qquad \forall j_2. \tag{16}$$

3.1. Using the KKT

In linear bi-level programming problems, the second-level mathematical model is a convex programming model, because decision variables are continuous and all terms in the objective function, as well as constraints, are all linear. Therefore, by using the KKT conditions, it can be converted into a series of the following constraints:

$$D_2(j_2) = \sum_{i_2} [\lambda(i_2)B_2(i_2, j_2)] - \theta(j_2) \qquad \forall j_2, \tag{17}$$

$$\sum_{j_1} A_2(i_2, j_1)X(j_1) + \sum_{j_2} B_2(i_2, j_2)Y(j_2) + S(i_2) = G_2(i_2) \qquad \forall i_2, \tag{18}$$

$$\lambda(i_2)S(i_2) = 0 \qquad \forall i_2, \tag{19}$$

$$\theta(j_2)Y(j_2) = 0 \qquad \forall j_2, \tag{20}$$

$$Y(j_2) \geq 0 \qquad \forall j_2, \tag{21}$$

$$\lambda(i_2) \geq 0 \qquad \forall i_2, \tag{22}$$

$$\theta(j_2) \geq 0 \qquad \forall j_2, \tag{23}$$

$$S(i_2) \geq 0 \qquad \forall i_2. \tag{24}$$

Replacing these constraints, rather than the second-level model, turns the problem into a single-level model.

Of course, Eqs. (19) and (20) are non-linear and the problem is of NLP type. However, we can linearize them using the following method by adding binary variables $P(i_2)$ and $Q(j_2)$ and using a large enough numerical coefficient called M_{big}:

$$\lambda(i_2) \leq M_{big}P(i_2) \qquad \forall i_2, \tag{25}$$

$$S(i_2) \leq M_{big}\left[1 - P(i_2)\right] \qquad \forall i_2, \tag{26}$$

$$\theta(j_2) \leq M_{big}Q(j_2) \qquad \forall j_2, \tag{27}$$

$$Y(j_2) \leq M_{big}\left[1 - Q(j_2)\right] \qquad \forall j_2. \tag{28}$$

Therefore, any bi-level linear programming problem can be converted into an MIP problem as follows:

$$\max_X : f_1 = \sum_{j_1} C_1(j_1)X(j_1) + \sum_{j_2} D_1(j_2)Y(j_2), \tag{29}$$

s.t.:

$$\sum_{j_1} A_1(i_1, j_1)X(j_1) + \sum_{j_2} B_1(i_1, j_2)Y(j_2) \leq G_1(i_1) \qquad \forall i_1, \tag{30}$$

$$X(j_1) \geq 0 \qquad \forall j_1, \tag{31}$$

$$D_2(j_2) = \sum_{i_2}\left[\lambda(i_2)B_2(i_2, j_2)\right] - \theta(j_2) \qquad \forall j_2, \tag{32}$$

$$\sum_{j_1} A_2(i_2, j_1)X(j_1) + \sum_{j_2} B_2(i_2, j_2)Y(j_2) + S(i_2) = G_2(i_2) \qquad \forall i_2, \tag{33}$$

$$\lambda(i_2) \leq M_{big}P(i_2) \qquad \forall i_2, \tag{34}$$

$$S(i_2) \leq M_{big}\left[1 - P(i_2)\right] \qquad \forall i_2, \tag{35}$$

$$\theta(j_2) \leq M_{big}Q(j_2) \qquad \forall j_2, \tag{36}$$

$$Y(j_2) \leq M_{big}\left[1 - Q(j_2)\right] \qquad \forall j_2, \tag{37}$$

$$Y(j_2) \geq 0 \qquad \forall j_2, \tag{38}$$

$$\lambda(i_2) \geq 0 \qquad \forall i_2, \tag{39}$$

$$\theta(j_2) \geq 0 \qquad \forall j_2, \tag{40}$$

$$S(i_2) \geq 0 \qquad \forall i_2, \tag{41}$$

$$P(i_2) \in \{0, 1\} \qquad \forall i_2, \tag{42}$$

$$Q(j_2) \in \{0, 1\} \qquad \forall j_2. \tag{43}$$

The above problem can be solved using MIP solvers such as CPLEX. Of course, computational experiments done using this model indicate that if the number of binary variables in the model increases, the computation time is prolonged, as well. The number of binary variables required to linearize the resulting single-level model is equal to the number of limitations plus the number of variables in the second-level model. Therefore, this method can only be used for those bi-level models whose second-level model is linear programming, with the smaller number of variables and constraints. The steps needed to apply this procedure for solving Bi-Level Linear Programming (BLLP) are as follows:

Step 1: Convert the second-level model into a series of constraints using the KKT conditions;

Step 2: Linearize nonlinear terms arising from KKT conditions;

Step 3: Solve the resulting single-level model using MIP solvers such as CPLEX.

3.2. Using the complete enumeration

If the decision variables of the first-level model are discrete and the number of different states for them is not too large, we can utilize the Complete Enumeration (CE) method to obtain an optimal solution to the problem. In this method, for each of the possible solutions for the vector of first-level decision variables, the second-level mathematical model is solved and the value of the leader's objective function is calculated. Then, among all the solutions, the solution is the optimal one for which the value of the leader's objective function is the best. In this method, if the number of possible solutions for the leader is high, the calculation time will be very long. If the first-level model contains only binary variables and the number of these variables is n, then the number of possible solutions is $2n$. If there are only 30 binary variables, more than 1 billion possible solutions should be considered. For each of these solutions, the second-level model should be solved, which will be very time consuming. Table 1 shows the number of possible solutions for the first-level model in terms of the number of binary variables.

The steps needed for solving Bi-Level Integer Programming (BLIP) by Complete Enumeration (CE) are given as follows:

Step 1: Consider negative infinity for the initial candidate value of the first-level objective function;

Step 2: Identify a feasible solution for the first-level decision variables;

Step 3: Set the values for the first-level variables as parameters in the second-level model;

Step 4: Solve the second-level model;

Table 1. The number of possible solutions based on the number of binary variables.

n	Number of solutions	n	Number of solutions	n	Number of solutions
1	2	11	2,048	21	2,097,152
2	4	12	4,096	22	4,194,304
3	8	13	8,192	23	8,388,608
4	16	14	16,384	24	16,777,216
5	32	15	32,768	25	33,554,432
6	64	16	65,536	26	67,108,864
7	128	17	131,072	27	134,217,728
8	256	18	262,144	28	268,435,456
9	512	19	524,288	29	536,870,912
10	1,024	20	1,048,576	30	1,073,741,824

Step 5: If the optimal solution for the second-level model is obtained, calculate the value of the first-level objective function;

Step 6: If the value obtained for the first-level objective function is better than the candidate's value, it should be replaced with the previous one. Then, save the values of the first- and second-level variables;

Step 7: Repeat Steps 1 to 6 for all feasible solutions for the first-level model;

Step 8: The candidate's values are identified as the optimal solution to the bi-level problem.

The complete enumeration method can only be used for models whose binary variables in the first-level model are approximately not more than 20 variables in number.

4. GPBLO algorithm

Because the computational time for bi-level optimization problems is very long, even for cases in which KKT conditions can be used or the complete enumeration method is applied, a faster approach to solving such problems is needed. Therefore, a method that can identify an optimal or near-optimal solution to bi-level problems in a relatively short time is proposed. The result of this method can be used as a candidate solution for applying exact methods such as branch and bound, or branch and cut, to reduce their computational time significantly.

To solve a variety of bi-level models, an algorithm that can identify either the optimal solution or the near-optimal solution is presented by solving a limited number of single-level mathematical models. In this method, several goal programming problems are solved to find an appropriate solution to the bi-level optimization, which is called GPBLO. In this method, first, the leader and follower objective functions are normalized by identifying the best and worst possible logical values for them according to fuzzy normalization. Then, by considering the importance coefficient of w for the leader's objective function and the importance coefficient $(1-w)$ for the follower's objective function, they are aggregated into a single form.

$$f=(w)\frac{f1-f1_{\min}}{f1_{\max}-f1_{\min}}+(1-w)\frac{f2-f2_{\min}}{f2_{\max}-f2_{\min}}. \quad (44)$$

To identify the best possible value for $f1$, the bi-level model from the objective function of the follower is released. Then, by solving a resulting single-level model, we obtain an upper bound for $f1$, which we call $f1_{\max}$. With the values obtained for variables X and Y, we calculate the value of $f2$. This value is the lower bound value for $f2$, which we call $f2_{\min}$.

To identify the upper bound value for $f2$, we release the bi-level model from the leader's objective function. Then, by solving a single-level model, we obtain an upper bound for $f2$, called $f2_{\max}$. With the values obtained for variables X and Y, we calculate the value of $f1$. This is the lower bound value for $f1$, which we call $f1_{\min}$. The coefficient w is a parameter whose value changes from 0 to 1. The increment in the value of this parameter can be any arbitrary number between 0 and 1. With smaller values for w, the better solutions may be detected, but the amount and time of calculations will increase. According to the computational experiments indicated in this paper, we consider this step to be 0.1. In this case, we need to solve 11 bi-objective models, for each of which the second-level model should be solved by fixing the values of the first-level variables. Moreover, two mathematical models must be solved to identify the best and worst rational values for $f1$ and $f2$. Therefore, in this case, we need to solve only 24 mathematical models. If the first- and second-level models are linear and the decision variables are also continuous, the goal programming model will also be linear. Besides, the goal programming model type is the same as the more complicated model type between leader and follower.

The goal programming model derived from the relaxed problem from the second-level objective function and, also, the replacement of the function f with the objective function of the first level can be stated as follows:

$$f=(w)\frac{f1-f1_{\min}}{f1_{\max}-f1_{\min}}+(1-w)\frac{f2-f2_{\min}}{f2_{\max}-f2_{\min}}, \quad (45)$$

s.t.:

$$\sum_{j_1} A_1(i_1,j_1)X(j_1)+\sum_{j_2} B_1(i_1,j_2)Y(j_2) \leq G_1(i_1) \qquad \forall i_1, \quad (46)$$

$$\sum_{j_1} A_2(i_2, j_1)X(j_1) + \sum_{j_2} B_2(i_2, j_2)Y(j_2) \leq G_2(i_2) \qquad \forall i_2, \tag{47}$$

$$X(j_1) \geq 0 \qquad \forall j_1, \tag{48}$$

$$Y(j_2) \geq 0 \qquad \forall j_2. \tag{49}$$

The steps to solve Bi-Level Programming (BLP) by the GPBLO are given below:

Step 1: Normalize the first- and second-level objective functions with a fuzzy normalization method;

Step 2: Consider the numerical coefficient w for the first-level normalized objective function and the numerical coefficient $(1 - w)$ for the second-level normalized objective function;

Step 3: Integrate two objectives in a new single-objective function for the problem;

Step 4: Ignore first- and second-level objective functions and consider negative infinity for the initial candidate value of the first-level objective function;

Step 5: Set the value of w to 0;

Step 6: Solve the resulting single-level model;

Step 7: Consider the values obtained for the first-level variables as parameters for the second-level model;

Step 8: Solve the second-level model;

Step 9: If the solution to the second-level model is optimal, then calculate the value of the first-level objective function;

Step 10: If the value obtained for the first-level objective function is better than the candidate's value, update the best found and save the values of the first- and second-level variables;

Step 11: Increase the value of w by 0.1 and repeat Steps 6 to 10 until w reaches 1;

Step 12: The solution given to the candidate's value and variables is the solution for the bi-level problem.

5. Computational analysis

In this section, 20 sample bi-level problems in different sizes that can be solved using the KKT method are randomly generated in the first stage. These problems are linear-linear and continuous-continuous. Then, we solve them using the CPLEX solver. In addition, we solve these problems using the GPBLO algorithm and a metaheuristic method based on Particle Swarm Optimization (PSO). Bi-level PSO has been employed in many published papers to solve bi-level models. Results of these methods are presented in Table 2. The values of the problem parameters are generated using the uniform distribution. These values are presented in Table 3.

Table 2. Comparison of the KKT method with the GPBLO.

Ins.	$\#X$[a]	$\#Y$[b]	$\#C1$[c]	$\#C2$[d]	KKT		GPBLO			PSO		
					$F1$[e]	Time[f]	$F1$	Time	Gap	$F1$	Time	Gap
1	2	2	2	2	411.7	0.1	411.7	0.2	0%	411.7	21.4	0%
2	3	3	3	3	357.9	0.1	357.9	0.2	0%	357.9	29.9	0%
3	4	4	4	4	439.0	0.2	439.0	0.2	0%	439.0	37.6	0%
4	5	5	5	5	514.2	0.1	514.2	0.2	0%	514.2	42.5	0%
5	8	8	8	8	519.0	0.2	518.9	0.9	0%	513.8	47.6	1%
6	10	40	10	20	457.0	0.5	449.7	0.2	2%	443.3	50.8	3%
7	20	50	10	30	399.2	1.6	384.6	0.2	4%	383.2	58.1	4%
8	30	60	20	40	384.8	2.0	377.0	0.2	2%	369.4	56.7	4%
9	40	70	30	50	364.1	2.8	357.0	0.4	2%	345.9	65.4	5%
10	50	80	40	60	364.6	3.5	356.3	0.3	2%	342.7	63.9	6%
11	60	90	50	70	345.9	9.8	319.3	0.4	8%	318.2	68.9	8%
12	70	100	60	80	358.5	9.6	329.5	0.4	8%	329.8	73.9	8%
13	80	110	70	90	353.1	8.8	323.5	0.4	8%	324.9	78.7	8%
14	90	120	80	100	345.8	16.1	328.8	0.4	5%	321.6	79.7	7%
15	100	130	90	110	358.5	21.3	351.6	0.5	2%	329.8	83.3	8%
16	110	140	100	120	334.6	20.5	323.4	0.5	3%	307.8	86.7	8%
17	120	150	110	130	346.4	67.0	330.1	0.8	5%	318.7	87.2	8%
18	130	160	120	140	351.9	75.1	330.5	0.7	6%	323.7	88.0	8%
19	140	170	130	150	354.1	90.3	324.5	0.8	8%	322.2	90.0	9%
20	150	180	140	160	349.9	131.7	339.3	0.9	3%	318.4	92.5	9%
							Mean		3%	Mean		5%

a: The number of variables in the first-level model; b: The number of variables in the second-level model;
c: The number of constraints in the first-level model; d: The number of constraints in the second-level model;
e: The value of the objective function of the first-level model; f: Calculation time in seconds.

Table 3. The values of problem parameters.

$C1(j_1)$ = Uniform (10,20)	$C2(j_1)$ = Uniform (10,20)
$D1(j_2)$ = Uniform (10,20)	$D2(j_2)$ = Uniform (10,20)
$A1(i_1, j_1)$ = Uniform (1,5)	$A2(i_2, j_1)$= Uniform (1,5)
$B1(i_1, j_2) = 0$	$B2(i_2, j_2)$ = Uniform (1,5)
$G1(i_1)$ = Uniform (50,90)	$G2(i_2)$ = Uniform (50,90)

As the problem size increases, the computational time for the MIP method becomes quite long. It can be observed that in these 20 instances, the value of the objective function approximately has a 3% gap with the optimal objective value. This is while the computational time, on average, is much shorter. Then, 20 bi-level problems were generated in different sizes. These problems are linear-linear and discrete-discrete. We consider the follower model as an integer linear programming type. Due to the non-convexity of this type of the model, the condition of KKT is no longer a necessary and sufficient condition for its optimality. Therefore, these problems cannot be solved by the KKT method. However, because the first-level variables are binary, we can obtain their optimal solutions by CE method. Each iteration will be solved by the CPLEX solver and, then, the best solution among the options is identified as the optimal solution. We also solve these problems using the GPBLO method to compare the results. The values of problem parameters are generated similar to Table 3 except for the parameter $G1(i1)$, which is generated randomly by a uniform (5,9) distribution. The results are shown in Table 4. It can be observed that in these 20 cases, the value of the objective

Table 4. Comparison of the CE method with the GPBLO.

Ins.	$\#X$[a]	$\#Y$[b]	$\#C1$[c]	$\#C2$[d]	CE[f]		GPBLO			PSO		
					$F1$[g]	Time[h]	$F1$	Time	Gap	$F1$	Time	Gap
21	5	5	1	5	284	2.1	284	1.5	0%	284	7.9	0%
22	6	10	1	5	476	4.1	474	1.6	0%	476	8.6	0%
23	7	15	1	10	431	5.6	426	1.8	1%	431	9.6	1%
24	8	20	1	15	360	12.1	356	1.8	1%	356	10.5	1%
25	9	25	1	20	336	7.9	336	3.3	0%	329	16.9	2%
26	10	30	1	25	325	39.6	314	3.5	3%	315	20.0	3%
27	11	35	1	30	313	22.0	313	2.7	0%	300	16.2	4%
28	12	40	1	35	336	27.9	333	5.9	1%	323	32.2	4%
29	13	45	1	40	301	15.2	298	6.3	1%	289	37.4	4%
30	14	50	1	45	295	42.9	282	13.4	4%	280	69.2	5%
31	15	55	1	50	313	112.2	313	11.6	0%	300	60.9	4%
32	16	60	1	55	303	287.0	294	20.2	3%	285	115.1	6%
33	17	65	1	60	279	483.9	279	48.0	0%	262	246.7	6%
34	18	70	1	65	270	375.6	250	34.9	7%	254	178.1	6%
35	19	75	1	70	293	248.6	293	42.5	0%	278	231.8	5%
36	20	80	1	75	278	515.7	272	44.1	2%	261	227.1	6%
37	21	85	1	80	319	590.4	315	66.3	1%	297	396.3	7%
38	22	90	1	85	—*	—*	301	275.1	–	286	813.6	–
39	23	95	1	90	—*	—*	263	345.6	–	250	955.3	–
40	24	100	1	95	—*	—*	282	134.2	–	268	758.6	–
							Mean		1%	Mean		4%

a: The number of variables in the first level model; b: The number of variables in the second level model;

c: The number of constraints in the first level model; d: The number of constraints in the second level model;

f: Complete Enumerations; g: The value of the objective function of the first level model;

h: Calculation time in seconds; * The algorithm was not performed due to the long calculation time.

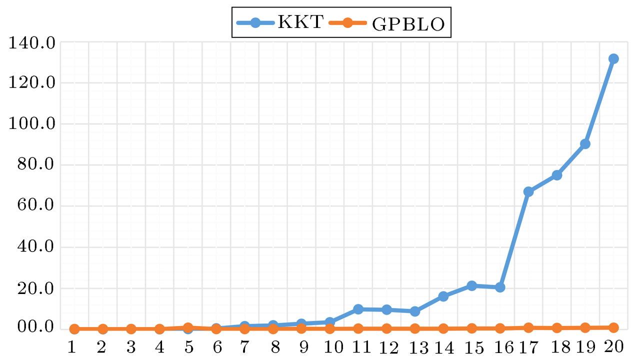

Figure 1. Comparison of KKT with GPBLO in terms of computation times (in seconds).

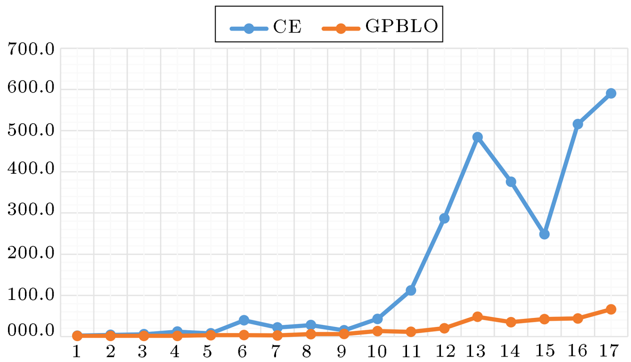

Figure 2. Comparison of CE with GPBLO in terms of computation times (in seconds).

function is 1% less than the optimal objective value with much shorter computational time. Figure 1 illustrates the comparison of KKT with GPBLO in terms of computation times (in seconds), and Figure 2 illustrates the comparison of CE with GPBLO in terms of computation times (in seconds).

In the following, several sample problems designed to test bi-level optimization algorithms are considered. These problems were presented in [18,21,27]. These problems are solved using the GPBLO algorithm in conjunction with the model obtained from KKT conditions, or by the CE method.

Instance 1 [16]:

$$\max_{x_1,x_2}: F_1 = -18x_1 + 10x_2 + 11y_1 - 11y_2 + 23y_3 + 40y_4$$

s.t.:

$$\max_{y_1,y_2,y_3,y_4}: F_2 = -35x_1 - 9x_2 + 20y_1 - 44y_2 + 10y_3 + 7y_4$$

s.t.:

$$47x_1 - 14x_2 - y_1 + 4y_2 + y_3 - 49y_4 \leq 1.5,$$

$$-23x_1 + 2x_2 + 45y_1 - 35y_2 + 12y_3 + 41y_4 \leq 13.5,$$

$$-9x_1 - 18x_2 + 12y_1 + 13y_2 + 37y_3 - 11y_4 \leq 5.5,$$

$$6x_1 - 19x_2 - y_1 - 2y_2 - 49y_3 - 11y_4 \leq -43.5,$$

$$-31x_1 - 8x_2 + 2y_1 + 17y_2 + 47y_3 - 25y_4 \leq 6.3,$$

$$46x_1 + 3x_2 - 28y_1 + 17y_2 - 36y_3 - 3y_4 \leq 22.5,$$

$$-45x_1 + 34x_2 - 44y_1 + 44y_2 + 16y_3 - 2y_4 \leq 17,$$

$$29x_1 - 13x_2 + 38y_1 + 19y_2 - 2y_3 + 7y_4 \leq 39,$$

$$13x_1 + 10x_2 + 27y_1 - 29y_2 - 49y_3 - 38y_4 \leq -38,$$

$$x_1, x_2, y_1, y_2, y_3, y_4 \geq 0.$$

Zhao et al. [16] provided three identified solutions to this problem that have been obtained using three different metaheuristic methods. Given that the second-level model in this problem is linear programming, this model can be converted into a single-level model and its global optimal solution is obtained using the KKT conditions, as previously stated. This problem was also solved using the GPBLO method. The results are presented in Table 5. As can be seen, the GPBLO solution is the same as the KKT solution and significantly outperforms the previously identified solutions.

Instance 2 [16]:

$$\max_{x_1,x_2}: F_1 = 5x_1 + 2x_2 + 4y_1$$

s.t.:

$$\max_{y_1,y_2}: F_2 = 3x_2 + 5y_1 - 2y_2$$

s.t.:

Table 5. The best solutions of Instance 1 for different methods.

Method	x^*	y^*	$F1$	$F2$
Method 1: Zhao et al., 2014 [16]	(0.897, 1.128)	(0.000, 0.075, 1.048, 0.534)	39.789	−30.561
Method 2: Zhao et al., 2015 [16]	(0.890, 1.125)	(0.000, 0.071, 1.045, 0.529)	39.613	−30.625
Method 3: Zhao et al., 2017 [16]	(0.894, 1.127)	(0.000, 0.074, 1.047, 0.532)	39.724	−30.463
KKT	(1.326, 1.289)	(0.000, 0.332, 1.257, 0.926)	51.311	−53.582
GPBLO	(1.326, 1.289)	(0.000, 0.332, 1.257, 0.926)	51.311	−53.582

Table 6. The best solutions of Instance 2 for different methods.

Method	x^*	y^*	$F1$	$F2$
Method 1: Zhao et al., 2014 [16]	(0.000, 0.000)	(2.000, 0.000)	8.000	10.000
Method 2: Zhao et al., 2015 [16]	(1.000, 0.000)	(1.000, 0.000)	9.000	5.000
Method 3: Zhao et al., 2016 [16]	(0.438, 0.000)	(1.563, 0.000)	8.437	7.813
KKT	(2.000, 0.000)	(0.000, 0.000)	10.000	6.000
GPBLO	(2.000, 0.000)	(0.000, 0.000)	10.000	6.000

Table 7. The best solutions of Instance 3 for different methods.

Method	x^*	y^*	$F1$	$F2$
HGAPSO-1 (Kuo and Han 2011) [27]	(999.265, 496.392)	(999.265, 496.392)	104229.91	201325.49
HGAPSO-2 (Kuo and Han 2011) [27]	(999.470, 499.503)	(999.470, 499.503)	104775.37	202175.61
HGAPSO-3 (Kuo and Han 2011) [27]	(999.932, 499.933)	(999.932, 499.933)	104990.20	202481.01
KKT	(1000.000, 500.000)	(1000.000, 500.000)	105000.00	202500.00
GPBLO	(1000.000, 500.000)	(1000.000, 500.000)	105000.00	202500.00

$$2x_1 + 2x_2 + 2y_1 + 4y_2 \leq 8,$$

$$x_1 + x_2 + y_1 \leq 2,$$

$$x_2 + x_3 + y_2 \leq 3,$$

$$x_1 \leq 4, \quad x_2 \leq 4, \quad y_1 \leq 2, \quad y_2 \leq 2,$$

$$x_1, x_2, y_1, y_2 \geq 0.$$

For this problem, Zhao et al. [16] provided three identified solutions obtained from three different metaheuristic methods. Given that the second-level model in this problem is linear programming, this model can be converted into a single-level model and its global optimal solution is obtained using the KKT conditions, as previously stated. This problem was also solved using the GPBLO method. The results are presented in Table 6. As can be seen, the GPBLO solution is the same as the KKT solution and is significantly better than the previously identified solutions.

Instance 3 [27]:

$$\max_{x_1,x_2} : F_1 = 110x_1 + 120x_2 - 40y_1 - 50y_2$$

s.t.:

$$\min_{y_1,y_2} : F_2 = 130x_1 + 145x_2$$

s.t.:

$$x_1 \leq y_1, \quad x_2 \leq y_2, \quad y_1 \leq 1000, \quad y_2 \leq 500,$$

$$y_1 + y_2 \geq 750, \quad x_1, x_2, y_1, y_2 \geq 0.$$

Kuo and Han [27] provided three identified solutions to this problem that were obtained from three different metaheuristic methods. Given that the second-level model in this problem is linear programming, this model can be converted as a single-level model and its global optimal solution is obtained using the KKT conditions, as previously stated. This problem was also solved using the GPBLO method. The results are presented in Table 7. As can be seen, the GPBLO solution is the same as the KKT solution and is better than the previously identified solutions.

Instance 4 [25]:

$$\max_{x} : F_1 = x + 10y,$$

s.t.:

$$\min_{y} : F_2 = y,$$

s.t.:

$$-25x + 20y \leq 30, \quad x + 2y \leq 10, \quad 2x - y \leq 15,$$

$$2x + 10y \geq 15, \quad x, y \in Z_+.$$

Tahernejad et al. [25] solved this problem using a metaheuristic method. Given that the second-level model is integer programming in this problem, the KKT method cannot be used to solve it. However, considering that the decision variables of the first-level model are integers, the global optimal solution can be obtained by using the CE method. This problem was also solved using the GPBLO method. The results are presented in Table 8. As can be seen, the results of these three methods are equal to each other.

Table 8. The best solutions of Instance 4 for different methods.

Method	x^*	y^*	$F1$	$F2$
Tahernejad et al., 2020 [25]	2.000	2.000	22.00	2.00
CE	2.000	2.000	22.00	2.00
GPBLO	2.000	2.000	22.00	2.00

6. Conclusions and future research

In this paper, a mat-heuristic algorithm called GPBLO for solving bi-level optimization problems was presented. Numerical experiments applied to small-, medium-, and large-sized instances whose optimal solution was obtained by solving a single-level model using KKT conditions or by a CE method indicated the efficiency of the algorithm. Several sample problems designed to test bi-level optimization algorithms were considered. This study solved these problems using the GPBLO algorithm under KKT conditions or CE methods. In some instances of these problems, the solution obtained from the GPBLO algorithm was the same as the global optimal solution and outperformed the previously identified solutions. Therefore, this method could be used to identify the solutions that might be optimal or near-optimal. For future research, it is possible to design exact algorithms that can identify the optimal solution in an acceptable time span.

References

1. Zhang, Y., Berman, O., Marcotte, P., et al. "A bilevel model for preventive healthcare facility network design with congestion", *IIE Transactions*, **42**(12), pp. 865–880 (2010).
2. Zhou, L., Lin, Y., Wang, X., et al. "Model and algorithm for bilevel multisized terminal location-routing problem for the last mile delivery", *International Transactions in Operational Research*, **26**(1), pp. 131–156 (2019).
3. Beresnev, V. and Melnikov, A. "Exact method for the capacitated competitive facility location problem", *Computers & Operations Research*, **95**, pp. 73–82 (2018).
4. Cao, D. and Chen, M. "Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach", *European Journal of Operational Research*, **169**(1), pp. 97–110 (2006).
5. Calvete, H.I., Galé, C., and Iranzo, J.A. "Planning of a decentralized distribution network using bilevel optimization", *Omega*, **49**, pp. 30-41 (2014).
6. Zaheri, F., Zandieh, M., and Taghavifard, M.T. "Bi-level programming for supplier selection under quantity discount policy", *Scientia Iranica*, **24**(4), pp. 2095–2104 (2017).
7. Brown, G., Carlyle, M., Diehl, D., et al. "A two-sided optimization for theater ballistic missile defense", *Operations Research*, **53**(5), pp. 745–763 (2005).
8. Amouzegar, M.A. and Moshirvaziri, K. "Determining optimal pollution control policies: An application of bilevel programming", *European Journal of Operational Research*, **119**(1), pp. 100–120 (1999).
9. Kaheh, Z., Baradaran Kazemzadeh, R., and Sheikh-El-Eslami, M.K. "A solution based on fuzzy max-min approach to the bi-level programming model of energy and exiramp procurement in day-ahead market", *Scientia Iranica*, **27**(2), pp. 846–861 (2020).
10. Sadati, S.M.B., Moshtagh, J., Shafie-khah, M., et al. "Operational scheduling of a smart distribution system considering electric vehicles parking lot: A bi-level approach", *Inter. J. Elec. Pow. & Ener. Sys.*, **105**, pp. 159–178 (2019).
11. Basciftci, B. and Van Hentenryck, P. "Bilevel optimization for on-demand multimodal transit systems", *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, Springer, pp. 52–68 (Sept. 2020).
12. Stouraitis, T., Chatzinikolaidis, I., Gienger, M., et al. "Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization", *IEEE Transactions on Robotics*, **36**(5), pp. 1452–1471 (2020).
13. Antil, H., Di, Z.W., and Khatri, R. "Bilevel optimization, deep learning and fractional Laplacian regularizatin with applications in tomography", *Inverse Problems*, **36**(6), 064001 (2020).
14. Sinha, A., Malo, P., and Deb, K. "A review on bilevel optimization: From classical to evolutionary approaches and applications", *IEEE Transactions on Evolutionary Computation*, **22**(2), pp. 276–295 (2017).
15. Bagloee, S.A., Asadi, M., Sarvi, M., et al. "A hybrid machine-learning and optimization method to solve bi-level problems", *Expert Systems with Applications*, **95**, pp. 142–152 (2018).
16. Zhao, X., Zheng, Y., and Wan, Z. "Interactive intuitionistic fuzzy methods for multilevel programming problems", *Expert Systems with Applications*, **72**, pp. 258–268 (2017).
17. Sinha, A., Lu, Z., Deb, K., et al. "Bilevel optimization based on iterative approximation of multiple mappings", *Journal of Heuristics*, **26**(2), pp. 151–185 (2020).
18. Hejazi, S.R., Memariani, A., Jahanshahloo, G., et al. "Linear bilevel programming solution by genetic algorithm", *Computers & Operations Research*, **29**(13), pp. 1913–1925 (2002).
19. Sinha, A., Malo, P., and Deb, K. "Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping", *European Journal of Operational Research*, **257**(2), pp. 395–411 (2017).
20. Wang, Y., Li, H., and Dang, C. "A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence", *INFORMS Journal on Computing*, **23**(4), pp. 618–629 (2011).

21. Sharma, A. "Optimistic variants of single-objective bilevel optimization for evolutionary algorithms", *International Journal of Computational Intelligence and Applications*, **19**(03), p. 2050020 (2020).

22. Shi, C., Lu, J., and Zhang, G. "An extended Kuhn-Tucker approach for linear bilevel programming", *Applied Mathematics and Computation*, **162**(1), pp. 51–63 (2005).

23. Zare, M.H., Borrero, J.S., Zeng, B., et al. "A note on linearized reformulations for a class of bilevel linear integer problems", *Annals of Operations Research*, **272**(1-2), pp. 99–117 (2019).

24. Dempe, S. and Zemkoho, A.B. "KKT reformulation and necessary conditions for optimality in nonsmooth bilevel optimization", *SIAM Journal on Optimization*, **24**(4), pp. 1639–1669 (2014).

25. Tahernejad, S., Ralphs, T.K., and DeNegre, S.T. "A branch-and-cut algorithm for mixed integer bilevel linear optimization problems and its implementation", *Mathematical Programming Computation*, **12**(4), pp. 529–568 (2020).

26. Aazami, A. and Saidi-Mehrabad, M. "Benders decomposition algorithm for robust aggregate production planning considering pricing decisions in competitive environment: A case study", *Scientia Iranica*, **26**(5), pp. 3007–3031 (2019).

27. Kuo, R.J. and Han, Y.S. "A hybrid of genetic algorithm and particle swarm optimization for solving bilevel linear programming problem – A case study on supply chain model", *Applied Mathematical Modelling*, **35**(8), pp. 3905–3917 (2011).

Biographies

Hossein Shams Shemirani received his BSc in Industrial Engineering from the Sharif University of Technology, Tehran, Iran. He also received his MSc in the same field from the Sharif University of Technology. He is currently a PhD student and a faculty of the Industrial Engineering Group, Golpayegan University of Technology, Isfahan, Iran. His research interests include operations research, maintenance management, and management information systems.

Rashed Sahraeian is a Professor of Industrial Engineering at Shahed University. He holds a BSc in Industrial Engineering from Amirkabir University of Technology, MSc, and PhD from Tarbiat Modares University. His research interests include operations research, supply chain management, and scheduling. He has published some books and a large number of papers in high-impact international journals and presented many papers in research-indexed journals and international conferences.

Mahdi Bashiri is a Professor of Industrial Engineering at Shahed University. He holds a BSc in Industrial Engineering from Iran University of Science and Technology and received MSc and PhD from Tarbiat Modares University. His research interests are facility design, meta-heuristics, and multi-response optimization. He has published some books and a large number of papers in high-impact international journals and many papers in research-indexed journals and international conferences.