International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 03 - Issue 01 || January 2018 || PP. 35-43

A hybrid DE-PSO algorithm for the nonlinear bilevel programming with special structure

Zhang Tao

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

Abstract: Bilevel programming problem (BLP) is a nested optimization problem that contains one optimization task as a constraint to another optimization task. In this paper, a hybrid differential evolution and particle swarm (DE-PSO) is proposed for solving the nonlinear BLP with special structure, in which the hybrid strategy can efficiently prevent the premature convergence of the swarm. Finally, we use the unconstrained test problems from the reference to measure and evaluate the proposed algorithm.

Keywords: Nonlinear bilevel programming; particle swarm optimization; differential evolutionary; fitness

1. Introduction

The bilevel programming problem (BLP) is a nested optimizations problem with two levels in a hierarchy: the upper and lower level decision-makers. The upper level maker makes his decision firstly, followed by the lower level decision maker. The objective function and constraint of the upper level problem not only rely on their own decision variables but also depend on the optimal solution of the lower level problem. The lower level has to optimize its own objective function under the given parameters from the upper level and the upper lever selects the parameters which feedback from the lower level to optimize the whole problem. Since many practical problems, such as engineering design, management, economic policy and traffic problems, can be formulated as hierarchical problems, BLP has been studied and received increasing attention in the literatures. During the past decades, some surveys and bibliographic reviews were given by several authors [1–4]. Reference books on bilevel programming and related issues have emerged [5–8].

The bilevel programming problem is a nonconvex problem, which is extremely difficult to solve. As we know, BLP is a NP-Hard problem [9-11]. Vicente et al. [12] also showed that even the search for the local optima to the bilevel linear programming is NP-Hard. Even so, many researchers are devoted to develop the algorithms for solving BLP and propose many efficient algorithms. To date a few algorithms exist to solve BLP, it can be classified into four types: Karus-Kuhn-Tucker approach (KKT) [13-16], Branch-and-bound method [17], penalty function approach [18-21] and descent approach [22, 23]. The properties such as differentiation and continuity are necessary when proposing the traditional algorithms. Unfortunately, the bilevel programming problem is nonconvex. Thus, many researchers tend to propose the heuristic algorithms for solving BLP because of their key characteristics of minimal problem restrictions such as differentiation. Mathieu et al. [24] firstly developed a genetic algorithm (GA) for bilevel linear programming problem because of its good characteristics such as simplicity, minimal problem restrictions, global perspective and implicit parallelism. Motivated by the same reason, other kinds of genetic algorithm for solving bilevel programming were also proposed in [25–28].

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 [29]. Due to its high speed of convergence and relative simplicity, the PSO algorithm has been employed for solving BLP problems. For example, Li et al. [30] proposed a hierarchical PSO for solving BLP problem. Kuo and Huang [31] applied the PSO algorithm for solving bilevel linear programming problem. Jiang et al. [32] presented the PSO based on CHKS smoothing function for solving nonlinear bilevel programming problem. Gao et al. [33] presented a method to solve bilevel pricing problems in supply chains using PSO. Zhang et al. [34] presented a new strategic bidding optimization technique which applies bilevel programming and swarm intelligence. In addition, the hybrid algorithms based on PSO are also proposed to solve the bilevel programming problems [35-37]. Though the PSO algorithm has widely applications in optimization problems, the global convergence of the PSO cannot be guaranteed [38].

In this paper, a hybrid differential evolution and particle swarm (DE-PSO) is proposed for solving the BLP, in which the hybrid strategy can efficiently prevent the premature convergence of the swarm. The rest of this paper is organized as follows. Sect.2 introduces the definitions and properties of bilevel programming problems. Sect.3 proposes the DE-PSO algorithm for BLP. We use the unconstrained test problems from the reference to measure and evaluate the proposed algorithm in Sect.4. While the conclusion is reached in Sect.5.

2. Formulation and properties of BLP

Let $c_1, c_2 \in R^{n_1}, d_1, d_2 \in R^{n_2}, A \in R^{m\times n_1}, B \in R^{m\times n_2}, b \in R^m$. $Q \in R^{(n_1+n_2)\times(n_1+n_2)}$. The BLP can be formulated as follows:

$$\max_{(x,y)} F(x,y) = c_1^T x + d_1^T y + \frac{1}{2}(x^T, y^T)Q(x^T, y^T)^T$$

Where y solves the following problem: (1)

$$\max_{y} f(x,y) = c_2^T x + d_2^T y$$

$$\text{s.t.} \quad Ax + By \le b,$$

$$x, y \ge 0,$$

where $F(x,y), f(x,y)$ are the upper level object function and lower level object function. The definitions of the BLP is as following:

(1) Constraint region of BLPP:

$$S = \{(x,y): x \in X, y \in Y, A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2\}$$

(2) Feasible set of the lower lever for each fixed $x \in X$:

$$S = \{y \in Y : By \le b - Ax\}$$

(3) Projection of S onto the leader's decision space:

$$S(x) = \{(x,y): x \in X, \exists y \in Y, A_1 x + B_1 y \le b_1, A_2 x + B_2 y \le b_2\}$$

(4) Follower's rational reaction set for $x \in S(X)$

$$P(x) = \{y \in Y : y \in \arg\min[f(x,y^*) : y^* \in S(x)]\}$$

(5) Inducible region

$$IR = \{(x, y) \in S, y \in P(x)\}$$

Definition 2.1. A point (x, y) is feasible if $(x, y) \in IR$.

Definition 2.2. A feasible point (x^*, y^*) is an optimal solution if $(x^*, y^*) \in IR$ and

$F(x^*, y^*) \le F(\bar{x}, \bar{y})$, $\forall(\bar{x}, \bar{y}) \in IR$.

Definition 2.3. If (x^o, y^o) is the optimistic solution for problem (1), the (x^o, y^o) is given by:

$\min_{x,y}\{F(x, y) \mid y \in P(x), G(x, y) \le 0\}$.

3. The DE-PSO algorithm for the nonlinear bi-level programming with special structure

For the fixed $x \ge 0$, the optimal solution to the lower programming can be obtained by solving the following linear programming:

$$\max_y f(x, y) = c_2^T x + d_2^T y$$
$$\text{s.t. } By \le b - Ax, \tag{2}$$
$$y \ge 0,$$

Note that $c_2^T x$ is constant, we can assume $c_2 = 0$ when solving the lower problem. Thus, we can get the dual problem of problem (2) written as follows:

$$\min_u (b - Ax)^T u$$
$$\text{s.t } B^T u \ge d_2, \tag{3}$$
$$u \ge 0,$$

where $u \in R^m$ is the dual variable.

Theorem 1. (x^*, y^*) is the optimal solution to the problem (1) if and only if there exists u^* such that (x^*, y^*, u^*) is the solution to the following problem:

$$\max_{x,y,u} F(x, y) = c_1^T x + d_1^T y + \frac{1}{2}(x^T, y^T)Q(x^T, y^T)^T$$
$$\text{s.t. } Ax + By \le b,$$
$$B^T u \ge d_2, \tag{4}$$

$$d_2^T y-(r-Ax)^T u=0,$$

$$x, y, u \geq 0 .$$

Proof. According to the duality theorem of the linear programming, it is obvious that there exists x^*, y^*, u^* such that $d_2^T y^*-(b-Ax^*)^T u^*=0,$ if and only if y^* solve the problem (2) for the fixed x^*.

By Theorem 1, the original bilevel problem (1) can be equally transformed into the traditional programming (4). Thus, we can get the solution of (1) by solving the problem (4). Note that the constraints are all linear in problem (4) except the nonlinear constraint $d_2^T y-(b-Ax)^T u=0$.We can solve a series of nonlinear programming with only linear constraints by relaxing the nonlinear constraint to replace solving problem (4).

Let $U=\{u \mid B^T u \geq d_2, u \geq 0\}$ denote the feasible region to linear programming (3). Then the following conclusions are listed by the theory of the linear programming [39]:

Conclusion 1. The feasible region of the linear programming has at least one vertex and at most finite vertexes if it is not empty.

Conclusion 2. If there exist an optimal solution to the linear programming, it must be one vertex of the feasible region.

By the conclusions, there are finite vertexes in the feasible region U and u^* is one of the vertexes of U.Therefore, we can transform the problem (4) into a series of following nonlinear programming problems by obtaining all vertexes of U, denoted by $U^E=\{u^1, u^2, \ldots u^t\}$, according to the method in linear programming [40].

$$\max_{x,y} F(x, y)=c_1^T x+d_1^T y+\frac{1}{2}(x^T, y^T) Q(x^T, y^T)^T$$

$$\text{s.t.} \quad Ax+By \leq b, \tag{5}$$

$$d_2^T y-(b-Ax)^T u^i=0,$$

$$x, y \geq 0 .$$

It is clear that solving the above problem is easier than the problem (4).

The quadratic programming problem (5) should be written as the nonlinear programming without constraint because it is difficult to deal with the constraints in the DE-PSO algorithm.

Firstly, the problem (5) is transformed into the following formulation:

$$\max_{x,y} \quad c_1^T x + d_1^T y + \frac{1}{2}(x^T, y^T)Q(x^T, y^T)^T$$

$$\text{s.t.} \quad (A \quad B)\begin{pmatrix} x \\ y \end{pmatrix} \le b, \tag{6}$$

$$\begin{pmatrix} -(u^i)^T & d_2^T \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \le b^T u^i,$$

$$\begin{pmatrix} -(u^i)^T & -d_2^T \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \le -b^T u^i,$$

$$\begin{pmatrix} -I_{n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & -I_{n_2} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \le 0,$$

Namely,

$$\max_{x,y} \quad c_1^T x + d_1^T y + \frac{1}{2}(x^T, y^T)Q(x^T, y^T)^T$$

$$\text{s.t.} \begin{pmatrix} A & B \\ -(u^i)^T & d_2^T \\ (u^i)^T & -d_2^T \\ -I_{n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & -I_{n_2} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} r \\ b^T u^i \\ -b^T u^i \\ 0_{(n_1+n_2)\times 1} \end{pmatrix}$$

By the duality theory of the nonlinear programming, the quadratic programming problem (6) can be written as the following nonlinear programming without constraint:

$$\max_{\lambda\ge 0} \quad -\frac{1}{2}\lambda^T M\lambda + d^T\lambda + \frac{1}{2}(c_1^T, d_1^T)Q^{-1}(c_1^T, d_1^T)^T$$

Where

$$M = -\begin{pmatrix} A & B \\ -(u^i)^T & d_2^T \\ (u^i)^T & -d_2^T \\ -I_{n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & -I_{n_2} \end{pmatrix} Q^{-1} \begin{pmatrix} A & B \\ -(u^i)^T & d_2^T \\ (u^i)^T & -d_2^T \\ -I_{n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & -I_{n_2} \end{pmatrix}^T, \tag{7}$$

$$d = -\left[\begin{pmatrix} r \\ b^T u^i \\ -b^T u^i \\ 0_{(n_1+n_2)\times 1} \end{pmatrix} - \begin{pmatrix} A & B \\ -(u^i)^T & d_2^T \\ (u^i)^T & -d_2^T \\ -I_{n_1} & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & -I_{n_2} \end{pmatrix} Q^{-1}\begin{pmatrix} c_1 \\ d_1 \end{pmatrix}\right].$$

The DE-PSO algorithm for nonlinear bilevel programming with special structure:

Step 1: Generate all vertexes $U^E = \{u^1, u^2, ... u^t\}$ of U by the method of linear programming.

Step 2: Solve the problem (7) with $u^i \ (i = 1,2,...t)$ by DE-PSO algorithm. If there is none feasible solution, then let $(0,0)$ and $F_k = -\infty$. Otherwise, let (x^k, y^k) be the optimal solution and $F_k = F(x^k, y^k)$.

Step 3: Compare $F_k \ (k = 1,2,...t)$, let $F^* = \max\{F_k, k = 1,2,...,t\}$, and the corresponding (x^k, y^k) be the optimal solution (x^*, y^*).

Step 4: If $F_k = -\infty$, then there is no feasible solution to problem (1), otherwise (x^*, y^*) is the solution to the problem (1) and F^* is the upper level optimal value of problem (1).

The DE-PSO algorithm:

Step 1. Initialize the population P with N particles. $p_i(t)$ is the t-th generation particle, p_{ibest} is the i-th history best position. $p_{i,nbest}$ is the best neighbor of i-th particle. Let $t = 0$.

Step 2. Calculate the fitness of each particle according to the fitness function.

Step 3. Search the neighborhood of $p_{i,nbest}$ using the random moving strategy:

If ($p_{ibest} == p_{i,nbest}$) then

$$x_i[t+1] = p_{i,nbest} + \delta * rand[a_j, b_j]^n;$$

$$v_i[t+1] = x_i[t+1] - x_i[t];$$

then, go to step3.

Step 4. Update the particle's personal best position.

If fitness ($x_i[t+1]$) < fitness (p_{ibest}), then $p_{ibest} = x_i[t+1]$;

Step 5. Update the particle's global best position.

Step 6. Update particle's position and velocity

$$v_i[t+1] = wv_i[t] + c_1 r_1 (gbest - x_i[t]) + c_2 r_2 (p_{i,nbest} - x_i[t]);$$

$$x_i[t+1] = x_i[t] + v_i[t+1];$$

Step 7. Stopping criterion. If $t = T$, stop. Otherwise, go to step 2.

where rand $[a_i, a_j]^n$ is a uniform random vector with its $j-th$ component in $[a_i, a_j]$; δ is a scale parameter for adjusting the perturbation to $p_{i,nbest}$; a_j and b_j are the lower and upper bounds of the component of vector $p_{i,nbest}$; n is the dimensionality. The inertia weight w controls the momentum of the particle by weighing the contribution of the previous velocity; c_1 and c_2 denote acceleration coefficients which are two positive constants. $r_1, r_2 \in rand(0,1)$, $w = 0.7298$, $c_1 = c_2 = 1.49618$.

4. Numerical experience

In this section we will present a nonlinear bilevel programming to illustrate the validity of the differential evolution and particle swarm optimization approach for the nonlinear bilevel programming.
Example[41]:

$$\min_x F(x, y) = -x^2 - y^2 + 16x + 5xy$$
$$s.t. \quad 0 \le x \le 20,$$
$$\min_y f(x, y) = y$$
$$s.t. \; x + y - 20 \le 0$$
$$0 \le y \le 10.$$

Table1 numerical experience result

i	u^i	(x^T, y^T)	F_i	Time consumed (s)
1	(0,1)	(10,10)	460	35.7082
2	(1,0)	(11.14286,8.85714)	468.99986	36.5246

We make program with Microsoft Visual C# and use a personal computer (CPU: Intel Pentium 1.7 GHz, RAM: 256MB) to execute the program. From table 1, it spend an average time of 35. 7082s for 5 iterations. it spend an average time of 36.5246 for 100 iterations. Table 1 is the final result of the numerical experiments, the relative error between the result and the optimal solution is in the 0.001 or less after 100 iterations. It shows our method is effective.

5. Conclusion

In this paper, a hybrid differential evolution and particle swarm (DE-PSO) is proposed for solving the nonlinear BLP with special structure, in which the hybrid strategy can efficiently prevent the premature convergence of the swarm. Finally, we use the unconstrained test problems from the reference to measure and evaluate the proposed algorithm.

Reference

[1]. Vicente L, Calamai P: Bilevel and multilevel programming: A bibliography review. Journal of Global Optimization. 5, 291-306(1994)

[2]. Dempe S: Annotated bibliography on bilevel programming and mathematical problems with

equilibrium constraints. Optimization. 52, 333-359(2003)

[3]. Colson B, Marcotte P, Savard G: Bilevel programming: A survey. 4OR: A Quarterly Journal of Operations Research. 3, 87-107(2005)

[4]. Colson B, Marcotte P, Savard G: An overview of bilevel optimization. Annals of Operations Research. 153, 235-256(2007)

[5]. Shimizu K, Ishizuka Y, Bard J: Non-differentiable and two-level mathematical programming. Kluwer Academic, Dordrecht (1997)

[6]. Bard J: Practical Bilevel Optimization: Algorithm and Applications. Kluwer Academic, Dordrecht (1998)

[7]. Migdalas A, Pardalos P, Varbrand P: Multilevel Optimization: Algorithms and Applications. Kluwer Academic, Dordrecht (1998)

[8]. Dempe S: Foundation of Bilevel Programming. Kluwer Academic, London (2002)

[9]. Jeroslow R: The polynomial hierarchy and a simple model for competitive analysis. Mathematical Programming. 32,146–164(1985)

[10]. Ben-Ayed O, Blair O: Computational difficulty of bilevel linear programming. Operations Research. 38, 556–560(1990)

[11]. Bard J:Some properties of the bilevel linear programming,. Journal of Optimization Theory and Applications. 68, 371–378(1991)

[12]. Vicente L, Savard G, Judice J: Decent approaches for quadratic bilevel programming. Journal of Optimization Theory and Applications. 81, 379–399 (1994)

[13]. Bard J: An algorithm for solving the general bilevel programming problem. Mathematics of Operations Research. 8, 260-272(1983)

[14]. Edmunds T, Bard J: Algorithm for nonlinear bilevel mathematical programs. IEEE Transactions on Systems, Men, and Cybernetics. 21, 83–89(1991).

[15]. Amouzegar M: A global optimization method for nonlinear bilevel programming problems. IEEE Transactions on Systems, Men, and Cybernetics. 29, 771-777(1999)

[16]. Etoa J: Solving quadratic convex bilevel programming problems using a smoothing method. Applied Mathematics and Computation. 217, 6680-6690(2011)

[17]. Bard J, Falk J: An explicit solution to the multi-level programming problem. Computers & Operations Research. 9, 77-100(1982)

[18]. Ishizuka Y, Aiyoshi E: A new computational method for Stackelberg and min-max problems by use of a penalty method. IEEE Transactions on Automatic Control. 26, 460-466(1981)

[19]. Aiyoshi E , Shimuzu K: A solution method for the static constrained Stackelberg problem via penalty method. IEEE Transactions on Automatic Control. 29, 1112-1114(1984).

[20]. Ishizuka Y, Aiyoshi E: Double penalty method for bilevel optimization problems. Annals of Operations Research. 34, 73-88(1992)

[21]. Lv Y, Hu T, Wang G , Wan Z: A penalty function method based on KuhnCTucker condition for solving linear bilevel programming. Applied Mathematics and Computation.188, 808-813(2007)

[22]. Savard G, Gauvin J: The steepest descent direction for the nonlinear bilevel programming problem. Operations Research Letters. 15, 265-272(1994)

[23]. Falk J, Liu J: On bilevel programming, Part I: general nonlinear cases. Mathematical Programming. 70, 47-72(1995)

[24]. Mathieu R, Pittard L, Anandalingam G: Genetic algorithm based approach to bi-level linear

programming. Operations Research. 28,1-21(1994)

[25]. Hejazi S, Memariani A, Jahanshanloo G, Sepehri M: Linear bilevel programming solution by genetic algorithm. Computers & Operations Research. 29,1913-1925(2001)

[26]. Wang Y, Jiao Y, Li H: An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handing scheme. IEEE Transactions on Systems Man and Cybernetics: Part C. 35(2), 221-232 (2005)

[27]. Wang G, Wang X, Wan Z: An adaptive genetic algorithm for solving bilevel linear programming problem. Applied Mathematics and Mechanics. 28(12), 1605-1612 (2007)

[28]. Calvete H, Gate C, Mateo P: A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research. 18,814-28(2008).

[29]. Kennedy J, Eberhart R, Shi Y: Swarm intelligence. Morgan Kaufmann Publisher, San Francisco (2001)

[30]. Li X, Tian P, Min X: Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems. Lecture Notes in Computer Science. 1169-1178(2006)

[31]. Kuo R, Huang C: Application of particle swarm optimization algorithm for solving bilevel linear programming problem. Computer and Mathematics with Application.58 678-685(2009)

[32]. Jiang Y, Li X: Application of particle swarm optimization based on CHKS smoothing function for solving nonlinear bilevel programming problem. Applied Mathematics and Computation. 129, 4332-4339(2013).

[33]. Gao Y, Zhang G, Lu J, Wee H: Particle swarm optimization for bi-level pricing problems in supply chains. Journal of Global Optimization. 51, 245-254(2011)

[34]. Zhang G, Zhang G, Gao Y, Lu J: Competitive strategic bidding optimization in electricity markets using bi-level programming and swarm technique. IEEE Transactions on Industrial Electronics. 58, 2138-2146(2011)

[35]. Yaakob S, Watada J:A hybrid intelligent algorithm for solving the bilevel programming models. Knowledge-Based and Intelligent Information and Engineering Systems. 6277, 485–494(2010)

[36]. Kuo R, Han Y: 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, 3905–3917(2011)

[37]. Wan Z, Wang G, Sun B: A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems. Swarm and Evolutionary Computation. 8, 26-32(2013)

[38]. Bergh F. V. D.: An analysis of particle swarm optimizers. Ph.D. Thesis, Department of Computer Science，University of Pretoria, Pretoria, South Africa, 2001.

[39]. Wan Z.P., Fei P. S.: The Theory and Algorithm of Optimization, Wuhan University Press, 2004.

[40]. Wei Q. L., Yan H.: The Theory and Model of Generalized Optimization, Science Press, 2003.

[41]. Zhong W. J., Xu N.R., Boltzmann machine method of two-level decision making, Journal of Systems Engineering. 10 (1), 7-13 (1995).