International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 02 - Issue 10 || October 2017 || PP. 38-44

A hybrid DE-PSO algorithm for bilevel linear programming

Tao Zhang

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 linear BLP, in which the hybrid strategy can efficiently prevent the premature convergence of the swarm. Finally, we use the test problems from the reference to measure and evaluate the proposed algorithm.

Keywords: Linear 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. The BLP problem and the updating strategy of particle swarm

For $x \in X \subset R^n, y \in Y \subset R^m, F: X \times Y \to R^1$, $f: X \times Y \to R^1$, the linear bilevel programming problem can be written as follows:

$$\min_{(x,y)} F(x,y) = c_1 x + d_1 y$$

$$\text{s.t. } A_1 x + B_1 y \le b_1 \tag{1}$$

$$\min_{y} f(x,y) = c_2 x + d_2 y$$

$$\text{s.t. } A_2 x + B_2 y \le b_2$$

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 linear bilevel programming

The DE-PSO algorithm—Algorithm 1

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

$$v_i[t+1] = wv_i[t] + c_1r_1(gbest - x_i[t]) + c_2r_2(p_{i,nbest} - x_i[t]);$$

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

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. $r_1, r_2 \in rand(0,1)$, $w = 0.7298$, $c_1 = c_2 = 1.49618$.

The DE-PSO algorithm for linear bilevel programming—Algorithm 2

Step 1. Initialization scheme. Initialize a random population (N_u) of the upper level variables. For each upper level member, initialize a random population (N_l) of the lower level variables and perform a lower level optimization procedure to determine the corresponding optimal lower level variables using **algorithm 1.**

Step 2. Combine the upper level variables and the corresponding optimal lower level variables to generate the complete upper level solution (Z^u). Evaluate the fitness value of the complete upper level solutions based on the upper level function and constraints.

Step 3. Stopping criterion. If $t = T$, stop. Otherwise, go to step 4.

Step 4. Update the upper level variables x_u^i as following:

$$\bar{x}_u^i = \begin{cases} x_u^i + \delta & \delta \in (0, b - x_u^i) & k > 0.5 \\ x_u^i + \delta & \delta \in (-(x_u^i - a), 0) & k \le 0.5 \end{cases},$$

where a and b are the upper and lower bounds of upper decision variables respectively , $\delta \in rand(a,b)$, $k \in rand(0,1)$ and $k \sim U(0,1)$ 。

Step 5. For each new $\bar{x}_u^i$, initialize a random population (N_l) of the lower level variables and perform a lower level optimization procedure to determine the corresponding optimal lower level variables using **algorithm 1.** Then, go to step 2.

4. Numerical experience

In this section, we will present some linear bilevel programming problems to illustrate the validity of the DE-PSO for the linear bilevel programming.

Example 1[39] Consider the following linear BLP problem, $x \in R^1, y \in R^1$

$$\max_{x \ge 0} \; x + 3y$$

$$s.t. \; \max_{y \ge 0} \; x - 3y$$

$$\text{s.t.} \; -x - 2y \le -10$$

$$x - 2y \le 6$$

$$2x - y \le 21$$

$$x + 2y \le 38$$

$$-x + 2y \le -18$$

Example 2[6] Consider the following linear BLP problem , $x \in R^2, y \in R^3$

$$\max_{x \ge 0} \; 8x_1 + 4x_2 - 4y_1 + 40y_2 + 4y_3$$

$$s.t \quad \max_{x \ge 0} \; -x_1 - 2x_2 - y_1 - y_2 - 2y_3$$

$$s.t \quad -y_1 + y_2 + y_3 \le 1$$

$$2x_1 - y_2 + 2y_2 - 0.5y_3 \le 1$$

$$2x_2 + 2y_1 - y_2 - 0.5y_3 \le 1$$

All results presented in this paper have been obtained on a personal computer (CPU:AMD Phenon (tm) ⅡX6 1055T 2.80GHz; RAM:3.25GB) using a C# implementation of the proposed algorithm. For the example 1and example 2, the $N_u = N_l = 50$.

Table1 numerical experience result

Example	Numerical result		Optimal solution		Relative error	Time consumed
	(x,y)	$F(x,y)$	(x,y)	$F(x,y)$	$\le 10^{-3}$	
Example 1	(15.9987,11.0003)	48.9998	(16,11)	49.0	2.1578 e-04	64.9874s
Example 2	(0,0.8998,0,0.5998,0.3994)	29.1902	(0,0.9,0,0.6,0.4)	29.2	3.4258e-04	35.7082s

5. Conclusion

In this paper, a hybrid differential evolution and particle swarm (DE-PSO) is proposed for solving the linear BLP, 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]. Anandalingam G. and White D. J.. A solution method for the linear static stackelberg problem using penalty functions[J] . IEEE Transaction on Automatic control, 35(10), 1170-1173 (1990)