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 level 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 \( x \in \mathbb{X} \subset \mathbb{R}^n \), \( y \in \mathbb{Y} \subset \mathbb{R}^m \), \( F : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{R}^l \), \( f : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{R}^l \), the linear bilevel programming problem can be written as follows:

\[
\begin{align*}
\min_{(x,y)} F(x, y) &= c_1 x + d_1 y \\
\text{s.t.} A_1 x + B_1 y &\leq b_1 \\
\min_{y} f(x, y) &= c_2 x + d_2 y \\
\text{s.t.} A_2 x + B_2 y &\leq b_2
\end{align*}
\]

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:
\[
S = \{(x, y) : x \in \mathbb{X}, y \in \mathbb{Y}, A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2 \}
\]

2. Feasible set of the lower lever for each fixed \( x \in \mathbb{X} \):
\[
S = \{ y \in \mathbb{Y} : By \leq b - Ax \}
\]

3. Projection of \( S \) onto the leader’s decision space:
\[
S(x) = \{(x, y) : x \in \mathbb{X}, \exists y \in \mathbb{Y}, A_1 x + B_1 y \leq b_1, A_2 x + B_2 y \leq b_2 \}
\]

4. Follower’s rational reaction set for \( x \in S(X) \)
\[
P(x) = \{ y \in \mathbb{Y} : y \in \arg \min \{ f(x, y^*) : y^* \in S(x) \} \}
\]

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
\[
F(x^*, y^*) \leq F(\tilde{x}, \tilde{y}), \forall (\tilde{x}, \tilde{y}) \in IR.
\]

Definition 2.3. If \((x^o, y^o)\) is the optimistic solution for problem (1), the \((x^o, y^o)\) is given by:
\[
\min_{x, y} \{F(x, y) \mid y \in P(x), G(x, y) \leq 0\}.
\]

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 \(t\)-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:
\[
\text{If}(p_{ibest} = p_{i, nbest})\text{ then}
\]
\[
x_i[t + 1] = p_{i, n} b e s t + \delta \cdot r a n d \cdot [a_j, b_j]^n;
\]
\[
v_i[t + 1] = x_i[t + 1] - x_i[t];
\]
then, go to step3.

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

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

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

Step 6. Update particle’s position and velocity
\[
v_i[t + 1] = w v_i[t] + c_1 r_1 (g b e s t - x_i[t]) + c_2 r_2 (p_{i, nbest} - x_i[t]);
\]
\[
x_i[t + 1] = x_i[t] + v_i[t + 1];
\]

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

where \(rand\) \([a_j, a_j]^n\) is a uniform random vector with its \(j\)-th component in \([a_j, a_j]\); \(\delta\) 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. $r_1, r_2 \in \text{rand}(0,1), w = 0.7298, c_1 = c_2 = 1.49618$.

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

$$
\begin{align*}
\overline{X}_u^i &= \begin{cases} 
    x_u^i + \delta & \delta \in (0, b - x_u^i) \quad k > 0.5 \\
    x_u^i - \delta & \delta \in (-x_u^i, 0) \quad k \leq 0.5
\end{cases}
\end{align*}
$$

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

Step 5. For each new $\overline{X}_u^i$, initialize a random population ($N_f$) 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, $x \in R^1, y \in R^1$

$$
\begin{align*}
\max_{x \geq 0} & \quad x + 3y \\
\text{s.t.} & \quad \max_{y \leq 0} x - 3y \\
& \quad -x - 2y \leq -10 \\
& \quad x - 2y \leq 6
\end{align*}
$$
Example 2 Consider the following linear BLP problem, $x \in R^2$, $y \in R^3$

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

subject to:

- $-x_1 - 2x_2 - y_1 - y_2 - 2y_3 \leq 0$
- $y_1 + y_2 + y_3 \leq 1$
- $2x_1 - y_2 + 2y_2 - 0.5y_3 \leq 1$
- $2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1$

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

Table 1 numerical experience result

<table>
<thead>
<tr>
<th>Example</th>
<th>Numerical result</th>
<th>Optimal solution</th>
<th>Relative error $\leq 10^{-3}$</th>
<th>Time consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>(15.9987,11.0003)</td>
<td>(16,11)</td>
<td>49.0</td>
<td>2.1578e-04</td>
</tr>
<tr>
<td>Example 2</td>
<td>(0.0.8998,0,0.5998,0.3994)</td>
<td>(0,0.9,0,0.6,0.4)</td>
<td>29.2</td>
<td>3.4258e-04</td>
</tr>
</tbody>
</table>

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.

References


