An Improved Particle Swarm Optimization Algorithm for MINLP Problems

Wang Zhi
Department of Computer Science and Technology, Wuhan University of Science and Technology, 430081, Hubei, China
Email: wz.zz@163.com

Ma, Jian-Jun
Department of Science, Wuhan University of Science and Technology, 430081, Hubei, China
Email: mjj1958@126.net

Liu Zhao
Department of Computer Science and Technology, Wuhan University of Science and Technology, 430081, Hubei, China
Email: liuyuxi@yeah.net

Abstract

This paper, to solve the MINLP, presents an improved algorithm of PSO. The main characteristics of the improved algorithm includes: the introduction of backup-particles and the proposal of particles substitution strategy which improves the learning ability and updating velocity of particles. It proved by the classical values experiments that the improved algorithm possesses the features of accuracy and quick convergence at the same time.

Keywords: MINLP( Mixed-Integer Nonlinear Programs) PSO( Particle Swarm Optimization) Evolutionary Computation(EC)

1 Introduction

MINLP model refers to a kind of complicated nonlinear program problems which contain both the integer variables and continuous variables.

The general form of MINLP is:

Minimize \( f(X,Y) \), Subject to:

\[ g^i(X,Y) \leq 0, \quad i = 1,2,3,\ldots,j; \quad h^i(X,Y) = 0, \quad i = j+1,j+2,\ldots,k; \]

\[ X_{\text{lower}} \leq X \leq X_{\text{upper}}, \quad Y_{\text{lower}} \leq Y \leq Y_{\text{upper}}. \]

where:

\( X \in \mathbb{R}^p, \quad Y \in \mathbb{N}^q, \quad p + q = n. \quad \mathbb{R}^p \) is P-dimensional real number space, \( \mathbb{N}^q \) is Q-dimensional integral number space. \( f(x, y) \) is the nonlinear objective function, and \( g^i(X,Y), h^i(X,Y) \) are the nonlinear constrained functions. MINLP is a NP-complete problem which has been seen as a very complicated problem until now. But the solution of MINLP is possible with the development of computer technology. In references, to solve MINLP, there are generally three methods: branch—and—bound(B&B), Generalized Benders Decomposition(GBD) and Outside Approximation(OA). Aiming to deal with the limitations of the above three algorithms, this paper provides an improvement of PSO with the addition of substitution function and the enhancement of learning ability of particles, thus making the improved algorithm deal with MINLP with higher efficiency and better results.

2 PSO( Particle Swarm Optimization)

PSO, proposed by Eberhart and Kennedy in 1995, is A Global Optimization Evolutionary Algorithm, originating from the imitation of food-looking of birds.

The brief description of PSO is: A swarm of particles is initialized at random in a certain space in which the places of particles stand for possible solutions and every particle is flying at a certain velocity. By flying many times, that is, iteration, the swarm of particles gradually approaches to the optimal place, thus finding the optimal solution .In each iteration, particles update themselves by two extremums: One is the optimal solution found by a particle itself, called pBest, the other is the current optimal solution found by the swarm, called gBest.
Particles update their velocities and places on the basis of these two extremums:
\[ V = \mathcal{O} \cdot V + C1 \cdot \text{rand()} \cdot (p\text{Best}-X) + C2 \cdot \text{rand()} \cdot (g\text{Best}-X) \]  \hspace{1cm} (1)
\[ X = X + V \]  \hspace{1cm} (2)

\( V \) is the velocity of a particle, \( X \) is the place of the current particle, \( p\text{Best} \) and \( g\text{Best} \) are defined in the above, \( \text{rand()} \) is any random value in \((0, 1)\), \( C1 \) and \( C2 \) are learning genes. Usually \( C1=C2=2 \).

Chart 1 is the flowchart of PSO.

There are several drawbacks with PSO when dealing with MINLP: The one is: easily falling into the local optimal solution, The other is: inefficiency. So, this paper proposes the particles substitution strategy which improves particles velocity updating strategy to enhance their learning abilities and the flexibility of searching in global solution space. Chart 2 is the flowchart of the improved PSO algorithm.

3 The Improved PSO Algorithm

Chart 2. A Improved PSO Algorithm
3.1 Particles Substitution Strategy

The particles falling into a local optimal solution can hardly jump out of it by general moving strategies. So they have to be substituted by new particles, that is, substituting the particles falling into the local optimal solution with particles in the legitimate solution space. However, the generation of new particles and legitimate solutions often need a high cost, especially when the constrained conditions are harsh and when there are a large number of variables of the constrained inequality. Under the consideration of this, the paper suggests we establish a dynamic backup-particle reserve in which the particles move randomly in legitimate solution space. And when needed, particles can be chosen from the backup-particles to replace the particles falling into the local optimal solution to search the global optimal solution. In this case, on one hand the cost to generate basic legitimate solution can be reduced, on the other hand, the backup-particles or their tracks in legitimate solution space are well distributed so that the global search of the algorithm is ensured.

3.2 The substitution of particles is shown in the following chart

![Chart 3. A substitution Strategy](chart3.png)

3.3 The Particles Velocity Updating Strategy

Different from the traditional particles velocity updating strategy of PSO, the improved algorithm divides velocity into two respects: direction and step; and respectively establishes the relevant alternative strategy and testing methods. Within these strategies, the direction and step, as well as whether there is need to adopt a new testing method to generate a new value are determined mainly on the basis of the particles’ experience, the times of successes and failures and the obtained results and so on.

3.4 The pseudocode of the improved PSO algorithm is as follows

```plaintext
Initialize backup-particles;
Initialize on-duty-particles;
While (not terminated)
    Do {
        For each on-duty-particle {
            Calculate fitness value
            If the fitness value is better than the best
            fitness value (p-best) in history
                Set current value as the new p-best
            Else If (without hope)
                for each backup-particle       // the maintenance and
                updating of backup-particles
                    if( valid(present[]+v[]))
                        present[]=present+v[];
                    else while (not valid(present[]+v[]))
                        do{ v[]=rand()% L[];}    // generate a motional
                        vector of short step in random
                        if(used times > N)    // N is an adjustable constant
                            while (not valid(present[]))
                                do {present[]=rand();}
        }
    }
Choose the particle with the best fitness value of all the
particles as g-best
    For each particle {
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    }
}
```

4 The experimental results and comparative analysis

We select three classical testing problems to do values experiments on the sake of testing the efficiency, velocity and accuracy of the new PSO. The condition for experiments is: PII-366 CPU, 256M memory and Windows XP operating systems.

Question 1.
Minimize f (X,Y) =0.6224*(0.0625*y1)*x1*x2 +1.7781*(0.0625*y2)*
(x1)^2+ 3.1661* (0.0625*y1)^2*x2 +19.84* (0.0625*y1)^2*
x1;
Constrained conditions:
g1 (X,Y) =0.0193* x1- 0.0625*y1 ≤ 0 ;
g2 (X,Y) =0.0954* x1 -0.0625*y2 ≤ 0 ;
g3 (X,Y) =750*1728-π* (x1)^2 *x2 -4/3*π*(x1)^3 ≤ 0 ;
```
g4(X,Y) = x^2 - 2x - 40 \leq 0.

This testing problem is proposed by Reference [2] and has been dealt with by Reference [3, 4, 5, 6].

**Question 2.**

Min \( f(x1, x2, x3, y1, y2, y3, y4) = (y1-1)^2 + (y2-2)^2 + (y3-1)^2 - \ln(y4+1) + (x1-1)^2 + (x2-2)^2 + (x3-3)^2 \).

**Constrained conditions:**

\[
\begin{align*}
y1 + y2 + y3 + x1 + x2 + x3 & \leq 5; \\
(y3)^2 + (x1)^2 + (x2)^2 + (x3)^2 & \leq 5.5; \\
y1 + x1 & \leq 1.2; \\
y2 + x2 & \leq 1.8; \\
y3 + x3 & \leq 2.5; \\
y4 + x1 & \leq 1.2; \\
(y2)^2 + (x2)^2 & \leq 1.64; \\
(y3)^2 + (x3)^2 & \leq 4.25; \\
(y2)^2 + (x3)^2 & \leq 4.64; \\
x1, x2, x3 & \geq 0; \\
y1, y2, y3, y4 & \in \{0, 1\};
\end{align*}
\]

This problem has been dealt with by Reference [7, 8, 9, 10, 11].

**Question 3.**

\[
\sum_{i=1}^{n} \cos^4(x_i) - 2 \prod_{i=1}^{n} \cos^2(x_i)
\]

Max \( f(x_i) = \sqrt{\sum_{i=1}^{n} x_i^2} \),

**Constrained conditions:**

\[
0 < x_i < 10; \quad 0.75 \leq \prod_{i=1}^{n} x_i \leq 0.75n;
\]

Therein, \( i = 1, 2, \ldots, n; 0.75 \leq \prod_{i=1}^{n} x_i \leq 0.75n; \)

The above problem is called BUMP which is first proposed by Keane [12] in optimal structure design in 1994. Because BUMP possesses three superior features (super nonlinearity, super multi-peak, super high-dimensional), it has become an internationally universal testing problem for measuring algorithm optimization.

The paper uncovers the results of ten experiments by the improved PSO for the above problems. The numbers of on-duty particles and backup particles are all set as 30, and the results are as follows:

<table>
<thead>
<tr>
<th>Question 1 Experiment Result</th>
<th>PSO</th>
<th>NSPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1.600860046, 0.468498044) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860439, 0.468498055) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860460, 0.468498056) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860471, 0.468498057) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860482, 0.468498058) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860493, 0.468498059) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860504, 0.468498060) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860515, 0.468498061) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860526, 0.468498062) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
</tbody>
</table>

**Table 2 Experiment Result of Question 2**

<table>
<thead>
<tr>
<th>Question 2 Experiment Result</th>
<th>PSO</th>
<th>NSPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1.600860046, 0.468498044) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860439, 0.468498055) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860460, 0.468498066) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860471, 0.468498067) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860482, 0.468498068) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860493, 0.468498069) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860504, 0.468498070) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860515, 0.468498071) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860526, 0.468498072) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
</tbody>
</table>

**Table 3 Experiment Result of Question 3**

<table>
<thead>
<tr>
<th>Question 3 Experiment Result</th>
<th>PSO</th>
<th>NSPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1.600860046, 0.468498044) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860439, 0.468498055) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860460, 0.468498066) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860471, 0.468498067) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860482, 0.468498068) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860493, 0.468498069) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860504, 0.468498070) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860515, 0.468498071) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
<tr>
<td>( (1.600860526, 0.468498072) )</td>
<td>0.365</td>
<td>0.365</td>
</tr>
</tbody>
</table>

**Conclusion**

The paper provides an improved PSO basing on the analysis of PSO’s basic principle of work and presents the application of the improved PSO to MINLP. Experiments show that the improved PSO algorithm is both faster in convergence and more accurate in solution. The use of the improved PSO is convenient because only the fitness function, the expressions of constrained conditions and the limits of its variables are asked to input for different problems. In all, this algorithm is a very effective one to deal with MINLP and other optimization problems.
References