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:

$g^i(X,Y)<=0$, $i=1,2,3,\ldots,j$; $h^i(X,Y)=0$, $i=j+1,j+2,\ldots,k$;

$X^{lower}\le X\le X^{upper}$, $Y^{lower}\le Y\le Y^{upper}$. there in：$X\in R^p$, $Y\in N^q$, $p+q=n$. R^p is P-dimensional real number space, 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:

$$V=\varpi *V + C1* rand() * (pBest-X) + C2* rand()*(gBest-X) \quad (1)$$

$$X = X + V \quad (2)$$

V is the velocity of a particle, X is the place of the current particle, pBest and gBest are defined in the above, 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.

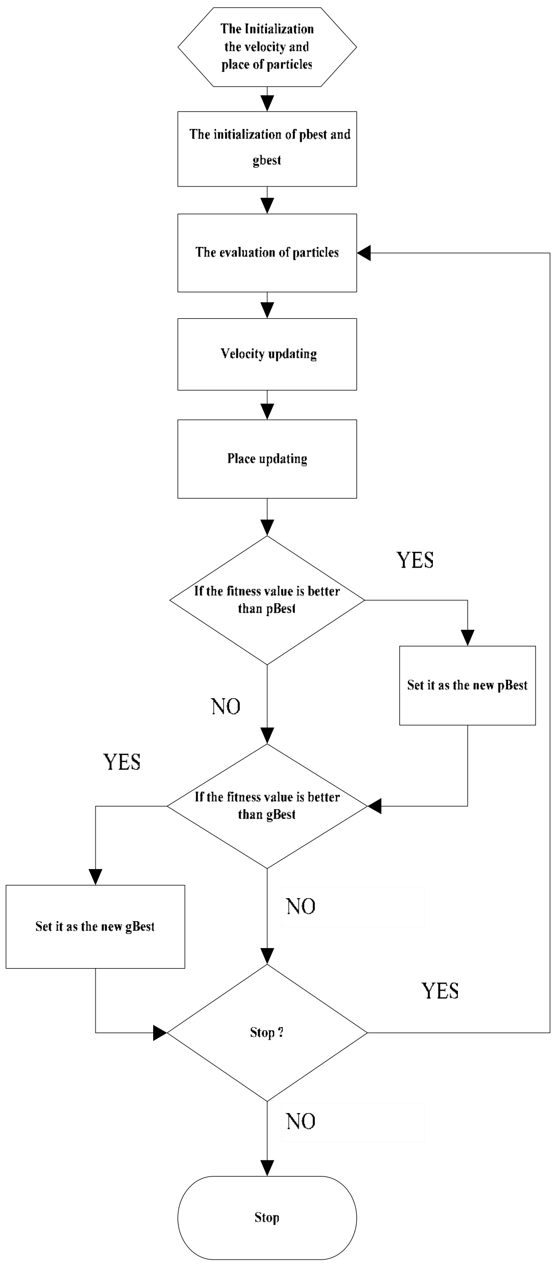

Chart 1. PSO Algorithm

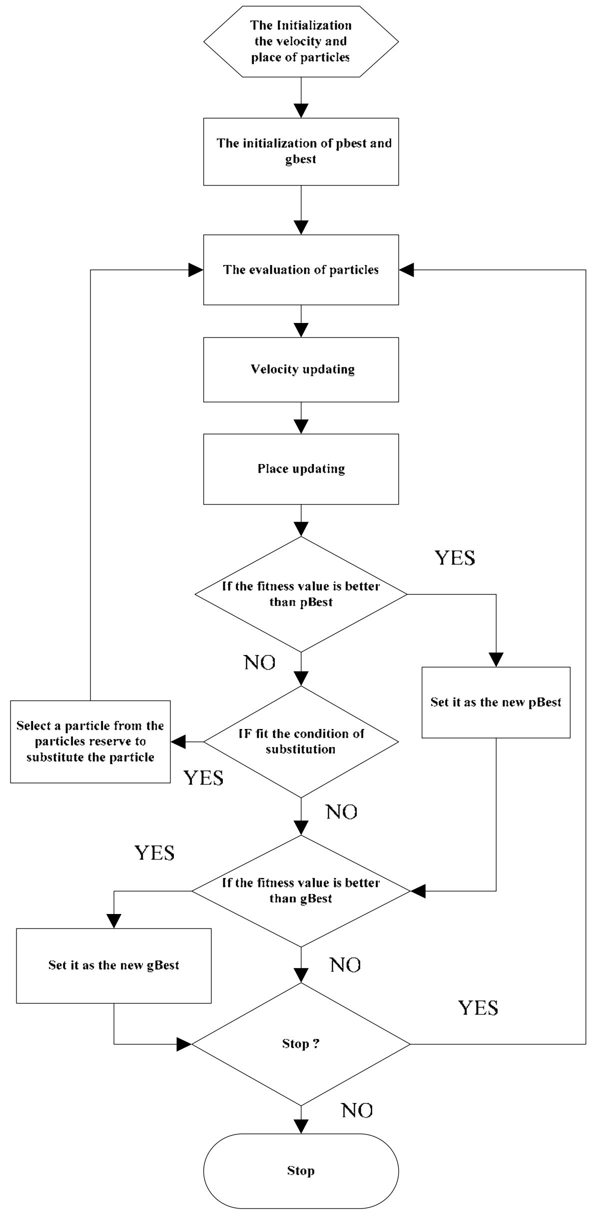

Chart 2. A Improved PSO Algorithm

3 The Improved PSO Algorithm

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.

3.1 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 3

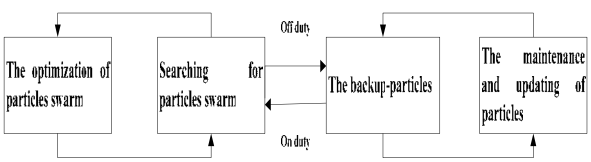

Chart 3. A substitution Strategy

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 alterative 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

```
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.

Question1.

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 $\leq$ 0 ;

g2 (X,Y) =0.00954*x1 -0.0625*y2 $\leq$ 0 ;

g3 (X,Y) =750*1728-π* (x1)^2 *x2 -4/ 3*π*(x1)^3 $\leq$ 0 ;

$$g4(X,Y) = x2 - 2\ 40 \le 0 .$$

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

Question2.

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

Constrained conditions :

$$y1+y2+y3+x1+x2+x3 \le 5;$$
$$(y3)^2 + (x1)^2 + (x2)^2 + (x3)^2 \le 5.5;$$
$$y1 + x1 \le 1.2;$$
$$y2 + x2 \le 1.8;$$
$$y3 + x3 \le 2.5;\ y4 + x1 \le 1.2;$$
$$(y2)^2 + (x2)^2 \le 1.64;$$
$$(y3)^2 + (x3)^2 \le 4.25;$$
$$(y2)^2 + (x3)^2 \le 4.64;$$
$$x1, x2, x3 \ge 0;$$
$$y1, y2, y3, y4 \in \{0, 1\};$$

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

Question3.

$$\text{Max } f(xi) = \frac{\left| \sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i) \right|}{\sqrt{\sum_{i=1}^{n} i x_i^2}},$$

Constrained conditions:

$$0<xi<10;$$

Therein ,i=1,2,…,n; $0.75 \le \prod_{i=1}^{n} x_i \le 0.75n$;

The above problem is called BUMP which is fist 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 1 Experiment Result of Question1

	PSO	NPSO	
	The optimal solution	The optimal solution	The worst solution
(y1, y2)	(12, 6)	(12, 6)	(12, 6)
x1	38.8601	38.8601,0362,6943,00	38.8601,0349,8189,06
x2	221.365	221.3654,7135,6008,24	221.3654,7333,7910,37
f(X,Y)	5850.38	5850.3830,6032,9161,70	5850.3830,7839,6416,70
time (s)	5.877	6.254	5.542
average time (s)	5.877	5.775	
average value	5850.38	5850.3830,7065,9293,50	

Table 2 Experiment Result of Question2

	PSO	IPSO	
	the optimal solution	the optimal solution	the worst solution
(y1, y2,y3,y4)	(1,0,0,1)	(1,0,0,1)	(1,0,0,1)
x1	0.2	0.1999 9999 9974 57	0.19999998111443
x2	1.28062	1.2806 2484 7486 57	1.28061919638455
X3	1.95448	1.9544 8202 8539 09	1.95448573323154
f(X,Y)	3.557473	3.5574 6125 8182 20	3.5574 6167 2283 77
time(s)	10.443	10.024	10.372
average time(s)	10.443	10.263	
average value	3.557473	3.5574 6157 4456 53	

Table 3 Experiment Result of Question3

	PSO	IPSO	
	the optimal solution	the optimal solution	the worst solution
x1	1.2869 5694 9968 87	1.6008 6041 8773 99	1.6008 4269 9735 93
x2	0.0647 5151 2956 92	0.4684 9805 9671 18	0.4685 0324 5274 33
f(X,Y)	0. 3649 7974 5870 66	0.3649 7974 5870 66	0.3649 7974 5130 04
time (s)	5.377	6.875	5.023
average time	5.377	5.544	
average value	0. 3649 7974 5870 66	0.3649 7974 5688 42	

In Question1, this algorithm calculated the current optimal solution with a shorter time and more accuracy.

In Question2, the solution is more accurate than current solutions of other algorithms.

In Question3, this algorithm calculated the optimal solution in a shorter time, in the mean while, it makes clear some other points also take the same optimal solution (0.36497974587066). The solutions of these questions are: (1.60086004652328,0.46849805155024);(1.60086040960895;0.46849806235336); (1.60086043325990;0.46849805543182);(1.60086044189444;0.46849805290488); (1.60086046865024;0.46849804507470);(1.60086046892781;0.46849804499346); and so on.

5 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

[1] I.E. Grossmann, N.V. Sahinidis. Special Issue on Mixed Integer Programming and It’s Application to Engineering, Optim. Eng, 3 (4), Kluwer Academic Publishers, Netherlands. 2002.
[2] E. Sandgren. Nonlinear Integer and Discrete Programming in Mechanical Design. ASME Journal Mechanical Design, 1990 , 112 (2):223～229
[3] B.K. Kannan, S.N. Kramer. An Augmented Lag Range Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design. Journal of Mechanical Design. Transactions on the ASME,1994,116(2):318～320
[4] Y.J. Cao, Q.H. Wu. Mechanical Design Optimization by Mixed-variable Evolutionary Programming. Proc of the 1997 Int'l Conf on Evolutionary Computation. Indianapolis: IEEE Press,1997.443～446
[5] Yung Chienlin. A Hybrid Method of Evolutionary Algorithms for Mixed-integer Nonlinear Optimization Problem. Proc of Congress on Evolutionary Computation. Piscataway, NJ:IEEE Press,1999.2159～2166
[6] A. Carlos, Cello Cello. Self-adaptive Penalties for GA-based Optimization. Proc of the Congress on Evolutionary Computation. Washington:IEEEPress,1999.537～580
[7] C. A. Floudas, A. Aggarwal, and A. R. Ciric, Global Optimum Search for Nonconvex NLP and MINLP problems. Computers & Chemical Engineering, 13 (10), 1117-1132, 1989.
[8] H. S. Ryoo, B. P. Sahinidis. Global Optimization of Non-convex NLPs and MINLPs with Application in Process Design. Computers & Chemical Engineering, 19, 551, 1995.
[9] L. Costa, P. Olivera. Evolutionary Algorithms Approach to The Solution of Mixed Integer Nonlinear Programming Problems. Computers & Chemical Engineering, 25, 257-266, 2001.
[10] M. F. Cardoso, R. L. Salcedo, S. Feyo de Azevedo. A Simulated Annealing Approach to The Solution of MINLP Problems. Computers & Chemical Engineering, 21, 1349-1364, 1997.
[11] R. L. Salcedo. Solving Non-convex Nonlinear Programming Problems with Adaptive Random Search. Industrial & Engineering Chemistry Research, 31, 262, 1992.
[12] A. J. Keane. Experience with Optimizers in Structural Design. The Conf on Adaptive Computing in Engineering Design and Control, PEDC, Plymouth, 1994.