I.J. Education and Management Engineering **2012, 2, 21-28**
Published Online February 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.02.04

Available online at http://www.mecs-press.net/ijeme

Improved Particle Swarm Optimization for Constrained Optimization

Zhicheng Qu[a], Qin Yang*[b]

Sichuan Agricultural University, Dujiangyan 611830, China

Abstract

In this paper, we present an improved particle swarm optimization (PSO) algorithm to solve constrained optimization problems. The proposed approach, called MPSO, employs a novel mutation operator to enhance the global search ability of PSO. In order to deal with constrains, MPSO uses mean violations mechanism and boundaries search. Simulation results on five famous benchmark problems show that MPSO achieves better results than standard PSO and another variant of PSO.

Index Terms: particle swarm optimization (PSO); evolutionary computation; constranined optimization

1. Introduction

In real world, many problems can be formulated as optimization problems. With the development of economic and society, these optimization problems become much complex. Some ones may have constrained conditions. To solve this kind of problems (constrained optimization problems), traditional pure mathematical methods suffer from some difficulties, such as constraints handling and boundaries search.

Generally, a constrained optimization problem can be mathematically described as follows.

$$\text{Minimize} \quad f(x) \tag{1}$$

subject to

$$g_i(x) \le 0, \quad i = 1, 2..., q \tag{2}$$

$$h_j(x) \le 0, \quad j = q+1, q+2..., m \tag{3}$$

*Corresponding author
E-mail address: [a]zhichengqu@yahoo.cn; [b]yangqin_sc@yahoo.com.cn

where $x \in R^D$, $g(x)$ is inequality constraint, $h(x)$ is the equality constraint, m is the number of constraints, q is the number of inequality constraints, m-q is the number of equality constraints.

During the last decades, some intelligent approaches have been proposed to solve constrained optimization problems. Compared to the traditional methods, these new kinds of approaches do not need to consider the properties of optimized problems. Michalewicz and Nazhiyath [1] proposed a co-evolutionary algorithm for numerical optimization with nonlinear constraints (Gencop III), in which the constraints are only used to judge whether the current search point is feasible or not. However, it is very difficult to generate initial feasible candidate solutions when the feasible search region is very small. In [2], the concept of constraint violation is introduced. It means the sum of the violation of all constraint functions. By simultaneously optimizing the constraint violation and the objective function, some excellent researches [2-4] achieved good results. One of the famous methods is the penalty function. For this kind of methods, the selection of the penalty coefficients highly depends on the features of problems. Different strength of the penalty may lead different performance. To tackle this problem, some methods based on dynamical controlled penalty coefficient were proposed, in which the constraint violation and the objective function are optimized separately. These approaches mainly focus on ranking mechanisms. Deb [2] proposed a method using an extended objective function to realize the ranking of individuals. Runarsson and Yao [3] introduced a stochastic ranking method based on evolutionary strategy. Besides of the above approaches, some researchers used multiobjective optimization methods to deal with the constraints and the object function at the same time [5]. Although this method is ideal, it is a more difficult and expensive task than solving single objective optimization problems.

In this paper, we present an improved PSO algorithm to solve constrained optimization problems. The proposed approach called MPSO employs a novel mutation operator to help trapped particles escape from local minima. Moreover, we use mean violations and boundaries search to deal with constrains. In order to verify the performance of MPSO, we test it on six benchmark functions. The simulation results show that MPSO outperforms PSO.

The rest of the paper is organized as follows. Section 2 briefly introduces the standard PSO algorithm. In Section 3, our improved PSO algorithm is proposed. The experimental results and discussions are presented in Section 4. Finally, the work and future work are summarized in Section 5.

2. Particle Swarm Optimization

Particle swarm optimization (PSO) is an intelligent computational technique firstly developed by Kennedy and Eberhart in 1995 [6]. It is a stochastic global optimization method which is based on the simulation of social behavior, such as bird flocking and fish schooling. As in genetic algorithm (GA) and evolutionary strategy (ES), PSO exploits a population of potential solutions to search the problem domain space. Compared to the aforementioned methods in PSO, no operators inspired by natural evolution are applied to extract a new generation of candidate solutions (called particle). In stead of mutation, PSO depends on the interaction of information between particles in the population. Each particle adjusts its flying direction according to its own previous best flying experience and the best experience attained by any particle of the population.

Since the introduction of the original version of PSO, many different PSO variants have been proposed. In order to control the global exploration and the local exploitation validly, Shi and Eberhart [7] introduced a concept of inertia weight to the original PSO and developed a modified PSO. Compared with the original version, the modified PSO gets a notable improvement of the performance in some problems, so it is often referred to as the standard PSO version and adopted by many following researches. The standard PSO version can be described by the following update equation [7]:

$$V(t+1) = wV(t) + c_1 r_1 (pbest_i(t) - X_i) + c_2 r_2 (gbest(t) - X_i(t)) \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

where $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ and $V_i = \{v_{i1}, v_{i2}, \ldots, v_{iD}\}$ represent the position vector and velocity vector of the *i*th particle, $pbest_i = \{pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD}\}$ is the best previous position yielding the best fitness value for the *i*th particle, and $gbest = \{gbest_1, gbest_2, \ldots, gbest_D\}$ is the global best particle found by all particles so far. The inertia weight w is a scaling factor controlling the influence of the old velocity on the new one. c_1 and c_2 are two constants known as cognitive and social coefficients, which determine the weight of $pbest_i$ and $gbest$, respectively. r_1 and r_2 are two random numbers generated by uniformly distribution in the range of [0, 1].

3. Improved PSO Algorithm

In PSO, each particle is attracted by two special particles, its own previous best particle $pbest_i$ and the global best particle *gbest*. Particles' movements highly depend on the quality of the above two kinds of particles. If these best particles are trapped in local minima, other particles will quickly converge to the minima by the attraction. To avoid this case, some researchers introduced mutation operators to the standard PSO. It is hope that the mutation could help trapped particles to escape from local minima.

In this paper, we employ a novel mutation operator which was proposed in our previous work. It is described as follows.

$$X_i^* = gbest + a_1 \cdot (gbest - X_i) + (1 - a_1) \cdot (X_{i1} - X_{i2}) \quad (6)$$

where X_i is the position vector of the *i*th particle, *gbest* is the global best particle found so far, a_1 is a random number within [0,1], $i1$ is a random integer within [1, *ps*], *ps* is the population size, and $i \neq i1 \neq i2$.

The main steps of the proposed MPSO are described as follows.

Step 1. Randomly initialize the population, and calculate the fitness value of each particle.

Step 2. Update the previous best particle pbesti and the global best particle gbest.

Step 3. Calculate the velocity of the each particle according to the equation (4).

Step 4. Calculate the position of the each particle according to equation (5).

Step 5. Calculate the fitness value of each particle in the population.

Step 6. For each particle Xi, if rand(0, 1)<pm, then generate a new particle Xi* according to equation (6) and calculate its fitness value, where rand(0,1) is a random number within [0,1] and pm is the probability of mutation; otherwise, go to Step 9.

Step 7. Select a fitter one between Xi and Xi* as new Xi.

Step 8. Update pbesti and gbest.

Step 9. If the terminate condition is satisfied, then stop the algorithm; otherwise, go to Step 3.

In order to deal with the constraints, we employ mean violations $\bar{v}$ by the suggestions of [8]. It is defined:

$$\bar{v}=\frac{\sum_{i=1}^{q}g_i(x)+\sum_{j=q+1}^{m}h_j(x)}{m} \tag{7}$$

where

$$g_i(x)=\begin{cases} g_i(x), & \text{if } g_i(x)>0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$h_j(x)=\begin{cases} \left|h_j(x)\right|, & \text{if } \left|h_j(x)\right|-\varepsilon>0 \\ 0, & \text{if } \left|h_j(x)\right|-\varepsilon\le 0 \end{cases} \tag{9}$$

The sum of all constraints violations is zero for feasible solutions and positive when at least one constraint is violated. An obvious application of the constraint violation is to use it to guide the search towards feasible areas of the search space. In this paper, ε is set to 0.0001.

In some cases, the boundaries may contain feasible solutions. To search the boundaries space, we use the following method.

$$x_{i,j}=\begin{cases} \overline{x_j}+k\left(x_{\min}-\overline{x_j}\right), & \text{if } x_{i,j}<x_{\min} \\ \overline{x_j}+k\left(x_{\max}-\overline{x_j}\right), & \text{if } x_{i,j}>x_{\max} \end{cases} \tag{10}$$

$$\overline{x_j}=\frac{\sum_{i=1}^{ps}x_{i,j}}{ps} \tag{11}$$

where k is a random number within [0, 1], ps is the population size, and $\left[x_{\min}, x_{\max}\right]$ is the definition domain.

4. Experimental Results

A. Test Problems

In order to verify the performance of MPSO, we select five famous benchmark functions in the following experiments. These problems were considered in an early study. Table I describes the features of the test problems, where |S| indicates the number of solutions randomly generated in the whole search space, |F| is the number of feasible solutions, |F|/|S| is the property of feasible solution space and the whole search space, LI is the number of linear inequality constraints, NI is the number of nonlinear inequality constraints, LE is the number of linear equality constraints, and NE is the number of nonlinear equality constraints.

Table 1. Main characteristics of the test problems

Problems	g01	g02	g03	g04	g05
D	13	20	10	5	4
Function	quadratic	nonlinear	polynomial	quadratic	cubic
|F|/|S| (%)	0.0003	99.9973	0.0026	27.0079	0.0000
LI	9	1	0	0	2
NI	0	1	0	8	0
LE	0	0	0	0	0
NE	0	0	1	0	3

The descriptions of the test suite are listed as follows.

g01: Minimize $f(x)=5\sum_{i=1}^{4}x_i-5\sum_{i=1}^{4}x_i^2-\sum_{i=1}^{13}x_i$

$$g_1(x)=2x_1+2x_2+x_{10}+x_{11}-10\le 0$$
$$g_2(x)=2x_1+2x_3+x_{10}+x_{12}-10\le 0$$
$$g_3(x)=2x_2+2x_3+x_{11}+x_{12}-10\le 0$$
$$g_4(x)=-8x_1+x_{10}\le 0$$
$$g_5(x)=-8x_2+x_{11}\le 0$$
$$g_6(x)=-8x_3+x_{12}\le 0$$
$$g_7(x)=-2x_4-x_5+x_{10}\le 0$$
$$g_8(x)=-2x_6-x_7+x_{11}\le 0$$
$$g_9(x)=-2x_8-x_9+x_{12}\le 0$$

where $0\le x_i\le 1(i=1,2,...,9), 0\le x_i\le 100(i=10,11,12), 0\le x_{13}\le 1$ and the global optimum is -15.

g02: Maximize $f(x)=\left|\frac{\sum_{i=1}^{D}\cos^4(x_i)-2\prod_{i=1}^{D}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{D}ix_i^2}}\right|$

$$g_1(x) = 0.75 - \prod_{i=1}^{D} x_i \le 0$$

$$g_2(x) = \sum_{i=1}^{D} x_i - 0.75D \le 0$$

where D=20, $0 \le x_i \le 10 (i = 1, 2, ..., D)$ and the global optimum is 0.803619.

$$\text{g03: Maximize } f(x) = \left(\sqrt{D}\right)^D \prod_{i=1}^{D} x_i$$

$$h_1(x) = \sum_{i=1}^{D} x_i^2 - 1 = 0$$

where D=10, $0 \le x_i \le 1 (i = 1, 2, ..., D)$ and the global optimum is 1.

$$\text{g04: } \begin{aligned} \text{Minimize } f(x) &= 5.3578547x_3^2 + 0.8356891x_1x_5 \\ &+ 37.293239x_1 - 40792.141 \end{aligned}$$

$$\begin{aligned} g_1(x) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\ &- 0.0022053x_3x_5 - 92 \le 0 \\ g_2(x) &= -85.334407 - 0.0056858x_2x_5 \\ &- 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0 \\ g_3(x) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\ &+ 0.0021813x_3^2 - 110 \le 0 \\ g_4(x) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 \\ &- 0.0021813x_3^2 + 90 \le 0 \\ g_5(x) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 \\ &+ 0.0019085x_3x_4 - 25 \le 0 \\ g_6(x) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 \\ &- 0.0019085x_3x_4 + 20 \le 0 \end{aligned}$$

where $78 \le x_1 \le 102, 33 \le x_2 \le 45, 27 \le x_i \le 45 \, (i = 3, 4, 5)$ and the global optimum is -30665.539.

$$\text{g05: } \begin{aligned} \text{Minimize } f(x) &= 3x_1 + 0.000001x_1^3 + 2x_2 \\ &+ (0.000002/3)x_2^3 \end{aligned}$$

$$g_1(x) = -x_4 + x_3 - 0.55 \le 0$$
$$g_2(x) = -x_3 + x_4 - 0.55 \le 0$$
$$h_3(x) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$$
$$h_4(x) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$$
$$h_5(x) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$$

where $0 \le x_1 \le 1200, 0 \le x_2 \le 1200, -0.55 \le x_3 \le 0.55, -0.55 \le x_4 \le 0.55$ and the global optimum is -5126.4981.

B. Simualtion Results

In this section, we compare the performance of PSO, RVPSO [9] and MPSO on the test suite. For PSO and MPSO, we use the following parameter settings. The population size *ps* is set 50, w = 0.72984, $c_1 = c_2$ = 1.49618. When the number of function evaluations reaches to 100, 000, the algorithm is terminated. For the sake of fair comparison, we use the same comparison strategy.

Table 2 presents the mean results of PSO, RVPSO and MPSO on the five test functions. It can be seen that MPSO achieves better results than PSO and RVPSO in all test cases except for g04. For this function, all the three algorithms could find the global optimum.

Table 2. The mean results achieved by PSO, RVPSO and MPSO

Problems	PSO	RVPSO [9]	MPSO
	Mean	Mean	Mean
g01	-14.376	-14.4187	-15
g02	-0.3328	-0.413257	-0.78304
g03	-1.0034	-1.0025	-1
g04	-30665.539	-30665.539	-30665.539
g05	53662.4026	5241.0549	5126.5832

5. Conclusion

In this paper, we propose an improved PSO algorithm called MPSO to solve constrained optimization problems. In MPSO, we employ three strategies. The first one is the mutation operator, which is beneficial for global search and help trapped particle jump out for local optima. The second one is the mean violations which could measure the constraints and judge whether the current candidate solutions is feasible or not. The last one is the boundaries search mechanism which helps the algorithm search the feasible solutions in the boundary region.

Experimental studies on five well-known benchmark problems show that the proposed approach MPSO outperforms standard PSO and RVPSO.

References

[1] Z. Michalewicz and G. Nazhiyath, "Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinar constraints," Proceedings of the Second IEEE International Conference on Evolutionary Compuation, IEEE Press, 1995, pp. 647-651.
[2] K. Deb, "An efficient constraint handling method for genetic algorhtms," Computer Methods in Applied Mechanics and Enginnering, vol. 186, no. 2/4, pp. 311-338, 2000.
[3] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," IEEE Transactions on Evolutionary Compuation, vol. 4, no. 3, pp. 284-294, 2000.
[4] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," IEEE Transactions on Evolutionary Compuation, vol . 9, no. 1, pp.. 1-17, 2005.
[5] A. H. Aguirre, S. B. Rionda, C. A. C. Coello, G. L. Lizarrga and E. M. Montes, "Handling constraints using multiobjective optimization concepts," International Journal for Numerical Methods in Engineering, vol. 59, no. 15, pp. 1989-2017, 2004.
[6] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, 1998, pp. 1942–1948.
[7] Y. Shi, R.C. Eberhart, "A modified particle swarm optimizer", Proceedings of the Conference on Evolutionary Computation, IEEE Press, Piscataway, 1998, pp. 69–73.
[8] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, N. Suganthan, C. A. C. Coello, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization," Technical Report, No. 2006005, Nanyang Technological University, Singapore and et al., Dec, 2005.
[9] H. Lu and W. Chen, "Dynamic-objective particle swarm optimization for constrained optimization problems," Journal of Combinatorial Optimization, vol. 12, pp. 409-419, 2006.