TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 12, December 2014, pp. 8286 ~ 8291
DOI: 10.11591/telkomnika.v12i12.5999

Particle Swarm Optimization with Simulated Binary Crossover

Lei Yang, Caixia Yang*, Yu Liu
Department of Electrical Information Engineering, Wuhan Polytechnic University,
Wuhan Polytechnic University, Wuhan 430023, China
*Corresponding author, e-mail: s_t_p@aliyun.com

Abstract

Particle swarm optimization (PSO) is a new intelligent search technique, which is inspired by swarm intelligence. Although PSO has shown good performance in many benchmark optimization problems, it suffers from premature convergence in solving complex multimodal problems. In this paper, we propose a novel PSO algorithm, called PSO with a simulated binary crossover operator (SCPSO), to improve the performance of PSO. Experimental results on several benchmark problems show that SCPSO achieves better performance than standard PSO.

Keywords*: particle swarm optimization, evolutionary algorithms, swarm intelligence, global optimization*

1. Introduction

Many real-world problems can be formulated as optimization problems in continuous or discrete variable space. In the passed decades, some intelligent algorithms inspired by nature have been proposed to solve optimization problems, such as Genetic Algorithms (GAs) [1], Particle Swarm Optimization (PSO) [2], Differential Evolution (DE) [3], etc. PSO is one of the most popular intelligent optimization algorithms, which has shown good search abilities in find solutions.

For PSO's simple concept, easy implementation, and efficiency, it has attracted much attention. The research of PSO becomes a hot spot in optimization algorithms. Shi and Eberhart introduced a parameter w called as inertia weight into the original PSO to balance the global and local search abilities [4]. A large w is better for global search, and a mall w is fitter for local search. It has been pointed out in [4] that a linearly decreasing w over the evolutionary process is efficient. Afterwards, the version of PSO with inertia weight is called standard PSO. Suganthan proposed a neighborhood technique, which used the local best particle instead of the global best, where the best local particle indicates the best particle in the neighborhood of each particle. For each particle, a predefined number of particles were considered as its neighbors. If the predefined number is equivalent to the swarm size, the local best becomes the global best [5]. Wang et al. [6] introduced opposition-based learning and Cauchy mutation for standard PSO. The proposed approach (OPSO) estimates not only the current particle but also the corresponding opposite particle. This is helpful to provide more chances for finding better solutions. In addition, OPSO also employs a Cauchy mutation operator conducting on the global best particle. It is to hope that the long tail of the Cauchy mutation could help trapped particles escape from local minima. Liang [7] proposed a comprehensive learning particle swarm optimizer (CLPSO) for global optimization of multimodal functions. The CLPSO uses a novel learning strategy whereby all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of swarm to be preserved to discourage premature convergence. The presented simulation results show that CLPSO outperforms other eight recently proposed PSO variants. Li and He [8] proposed a novel PSO algorithm (GMPSO), which employs a Gaussian mutation and a dynamic adaptation inertia weight. The presented results showed that GMPSO outperforms LOWPSO and PSODR for all benchmark functions.

In this paper, we propose an improved PSO variant (SCPSO), called PSO with a simulated binary crossover operator. In SCPSO, we generate two trail particles based on the simulated binary crossover operator. And then, the better trail particle competes with the worst

Received March 20, 2014; Revised August 28, 2014; Accepted September 19, 2014

particle in the current population, and the fitter one is selected as the new particle. In order to verify the performance of SCPSO, we test it on several function benchmark problems. Experimental results show that SCPSO outperforms standard PSO in all test cases.

The rest of the paper is organized as follows. In section 2, we briefly introduce the standard PSO. In Section 3, the proposed approach HCPSO is proposed. In Section 4, simulated studies are conducted, including test problems, parameter settings, results and discussions. Finally, the work and future work are given in Section 5.

2. Particle Swarm Optimization

Explaining research chronological, including research design, research procedure (in the form of algorithms, Pseudocode or other), how to test and data acquisition [1, 3]. The description of the course of research should be supported references, so the explanation can be accepted scientifically [2, 4].

Particle swarm optimization (PSO) was firstly developed by Kennedy and Eberhart, which is motivated from the social behavior of bird flocks or fish schooling [2]. It is a stochastic optimization algorithm which maintains a swarm of candidate solutions, referred to as particles. In PSO, each particle has two vectors, position (X) and velocity (V). Particles fly through the search space by flowing the previous best particles and the global best particles. Based on this model, the position and velocity of particles are updated from generation to generation. There are several main versions of the PSO algorithms, and the following version modified by Shi and Eberhart [4] is used in this paper.

$$V_i^{(t+1)} = w*V_i^{(t)} + c_1*rand1()*(pbest_i - X_i^{(t)}) + c_2*rand2()*(gbest - X_i^{(t)}) \quad (1)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)} \quad (2)$$

Where Xi and Vi are the position and velocity of the ith particle, pbesti and gbest are previous best particle of the ith particle and the global best particle found by all particles so far respectively, and w is an inertia factor [4], and rand1() and rand2() are two random numbers independently generated within the range of [0,1], and c1 and c2 are two learning factors which control the influence of the social and cognitive components.

In Equation (1), the first part is momentum vector, which represents the inertial motion of current velocity. The second part is the cognitive vector, which indicates the previous experiences and help particle find better positions. The third part is the social vector, which shares the information of the global best particle among particles.

3. PSO with a Simulated Binary Crossover Operator

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that make the reader understand easily [2, 5]. The discussion can be made in several sub-chapters.

Like other evolutionary algorithms (EAs), PSO is also based population random search algorithm. Although they have some commons in search solutions, they have differences for evolutionary operators. For a general EA, it has three evolutionary operators: selection, crossover and mutation. While in PSO, there is not any operator except for the velocity and position updating models. In this paper, we propose a novel PSO algorithm, which employs a simulated binary crossover operator to achieve good performance.

The simulated binary crossover operator is proposed by Deb and Beyer, which has been successfully applied to GAs [9]. In the simulated binary crossover operator, two offspring, $Y_1 = \left(y_{1,1}, y_{1,2}, ..., y_{1,D}\right)$ and $Y_2 = \left(y_{2,1}, y_{2,2}, ..., y_{2,D}\right)$, are generated based on two parent vectors $X_1 = \left(x_{1,1}, x_{1,2}, ..., x_{1,D},\right)$ and $X_2 = \left(x_{2,1}, x_{2,2}, ..., x_{2,D},\right)$ as follows:

$$y_{1,j} = \frac{1}{2}\left[\left(1-\beta_1\right)x_{1,j} + \left(1+\beta_1\right)x_{2,j}\right] \quad (3)$$

$$y_{2,j} = \frac{1}{2}\left[\left(1+\beta_2\right)x_{1,j} + \left(1-\beta_2\right)x_{2,j}\right] \quad (4)$$

Where $\beta_k (k = 1, 2)$ is a random number and generated as follows [9].

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{if } u(0,1) \le \frac{1}{2} \\ \left[2(1-u)\right]^{-\frac{1}{\eta+1}}, & \text{if } u(0,1) > \frac{1}{2} \end{cases} \quad (5)$$

Where $u(0,1)$ is a normally distributed random number within (0,1), and η is a parameter, which is set 5.0 in this paper.

In our proposed SCPSO, we use the above crossover operator to generate two trial particles Y_1 and Y_2. And then, we select the fittest one among Y_1, Y_2 and the worst particle P_w as the new P_w. For each particle X_i, we use the modified simulated binary crossover as follows:

$$y_{1,j} = \frac{1}{2}\left[\left(1-\beta_1\right)x_{i,j} + \left(1+\beta_1\right)Best_j\right] \quad (6)$$

$$y_{2,j} = \frac{1}{2}\left[\left(1+\beta_2\right)x_{1,j} + \left(1-\beta_2\right)Best_j\right] \quad (7)$$

In order to search better solutions, we introduce the information of the global best particle Best for the simulated binary crossover operator. The main steps of SCPSO are described in Algorithm 1, where ps is the population size, where rand(0,1) is a normally distributed random number within [0,1], p_c is the crossover rate, FEs is the number of function evaluations, and MAX_FEs is the maximum number of function evaluations.

Algorithm 1: SCPSO
Begin
While FEs < MAX_FEs **do**
For i=1 to ps **do**
Calculate the velocity of X_i according to (1);
Calculate the position of X_i according to (2);
Calculate the fitness value of X_i;
FE++;
If rand(0,1) < pc **then**
Generate Y_1 and Y_2 according to (6) and (7);
Calculate the fitness values of Y_1 and Y_2;
FEs=FEs+2;
Find the worst particle P_w in current population;
Select the fittest one among Y_1, Y_2 and P_w as the new P_w;
End If
End For
End While
End

4. Experimental Studies

4.1. Benchmark Problems

In this paper, ten well-known benchmark functions are selected in the experiments. According to the properties of functions, we divide then into two classes: unimodal functions (f_1-f_4) and multimodal functions (f_5-f_{10}). These functions were chosen from an early study in [10]. All

the functions used in this paper are to be minimized. The description of the benchmark functions and their global optima are listed as follows:

$$f_1 = \sum_{i=1}^{D} x_i^2$$

Where $x_i \in [-100, 100]$, D=30, and the global optimum is 0.

$$f_2 = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} x_i$$

Where $x_i \in [-10, 10]$, D=30, and the global optimum is 0.

$$f_3 = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$$

Where $x_i \in [-100, 100]$, D=30, and the global optimum is 0.

$$f_4 = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$$

Where $x_i \in [-1.28, 1.28]$, D=30, and the global optimum is 0.

$$f_5 = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{|x_i|}\right)$$

Where $x_i \in [-500, 500]$, D=30, and the global optimum is -12569.5.

$$f_6 = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$$

Where $x_i \in [-5.12, 5.12]$, D=30, and the global optimum is 0.

$$f_7 = -20 * \exp\left(-0.2 * \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos\left(2\pi x_i\right)\right) + 20 + e$$

Where $x_i \in [-32, 32]$, D=30, and the global optimum is 0.

$$f_8 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Where $x_i \in [-600, 600]$, D=30, and the global optimum is 0.

$$f_9 = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$$

Where $x_i \in [-50, 50]$, D=30, and the global optimum is 0.

$$f_{10} = \frac{\pi}{D}\{10\sin^2(3\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin^2(3\pi y_{i+1})] + (y_D - 1)^2 [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$$

Where $x_i \in [-50, 50]$, D=30, and the global optimum is 0.

4.2. Results

In the experiment, we compare SCPSO with standard PSO on the ten test problems. For the sake of fair competition, we use the same parameter settings for both PSO and SCPSO. For the common parameters, population size (*ps*), w, c_1, and c_2 are set as 10, 0.72984, 1.49618 and 1.49618, respectively. For SCPSO, the probability of crossover p_c is set 0.05. For all test problems, the maximum number of function evaluations MAX_FEs is set 150,000. Both PSO and SCPSO are run 25 times, and the mean fitness value and the standard deviation are recorded.

Table 1 presents the comparison results of standard PSO and SCPSO on the test suite, where 'Mean' indicates the mean best function values. As seen, SCPSO achieve better results than standard PSO on all test functions. Especially for f_2, f_3, f_4, f_8 and f_{10}, SCPSO significantly improves the results. It demonstrates that the proposed crossover operator is effective. Figure 1 shows the evolutionary processes of PSO and SCPSO shows the convergence curves of HCDE on f_1 and f_4. As seen, HCDE converges very fast over the evolution.

Table 1. The results achieved by standard PSO and SCPSO

Functions	Standard PSO		SCPSO	
	Mean	Std Dev	Mean	Std Dev
f_1	2.64e-10	3.17e-09	8.89e-13	5.58e-13
f_2	0.133	0.74	2.43e-06	6.82e-06
f_3	8.00	5.93	0	0
f_4	0.329	0.69	4.92e-03	3.96e-03
f_5	-6646.3	725.2	-7149.7	531.6
f_6	58.7	39.3	46.6	29.7
f_7	8.19	3.89	4.17	2.32
f_8	0.58	0.42	1.40e-09	3.62e-09
f_9	0.62	0.59	0.31	0.17
f_{10}	0.02	3.11e-03	8.90e-09	6.35e-09

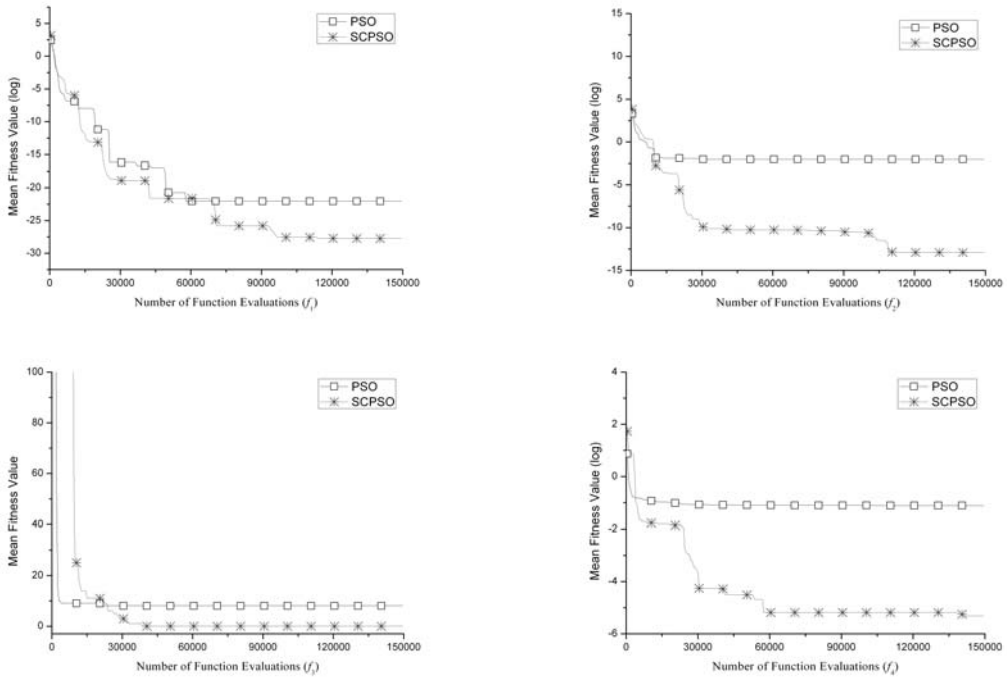

Figure 1. The evolutionary processes of standard PSO and SCPSO on four functions

4. Conclusion

This paper presents an improved PSO algorithm, called PSO with a simulated binary crossover operator. The proposed approach SCPSO employs a modified simulated binary crossover operator. First, we generate two trail particles based on current particle and the global best particle. Then, the better one between the two trail particles is compared the worst particle in current swarm. The fitter one is replaced with new worst particle. Simulation results show that SCPSO achieves better results than standard PSO in all test cases. It demonstrates that the proposed strategy is effective. In our future work, we will apply SCPSO to solve some real-world problems

References

[1] DE Goldberg, E David. *Editors*. Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley. NewYork City: Longman Publishing Co. 1989.
[2] J Kennedy, RC Eberhart. *Particle swarm optimization*. IEEE International Conference on Neural Networks. Perth. Australia. 1995: 1942–1948.
[3] R Storn, K Price. Differential evolution--A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997; 11: 341–359.
[4] Y Shi, RC Eberhart. *A modified particle swarm optimizer*. IEEE Proceedings of the Conference on Evolutionary Computation. Piscataway. 1998: 69–73.
[5] PN Suganthan. *Particle swarm optimiser with neighborhood operator*. Proceedings of the IEEE Congress on Evolutionary Computation. Washington DC. 1999: 1958–1962.
[6] H Wang, Y Liu, SY Zeng, H Li, CH Li. *Opposition-based particle swarm algorithm with Cauchy mutation*. Proceedings of the IEEE Congress on Evolutionary Computation. Brisbane. 2007: 4750–4756.
[7] JJ Liang, AK Qin, PN Suganthan. *Comprehensive learning particle swarm optimizer for global optimization of multimodal functions*. IEEE Transactions on Evolutionary Computation. Parsopoulos. 2006; 10(3):281–295. Parsopoulos
[8] L Li, X He. *Gaussion mutation Particle Swarm Optimization with dynamic adaptation inertia weight*. World Congress on Software Engineering. Xiamen. 2009; 4: 454–459.
[9] K Deb, H Beyer. Self-adaptive genetic algorithms with simulated binary crossover. *Evolutionary Computation Journal*. 2001; 9(2): 195-219.
[10] J Brest, S Greiner, B Boskovic, M Mernik, V Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.*. 2007; 10(6): 646-657.