Universal Journal of Mathematics and Mathematical Sciences
Volume 3, Number 2, 2013, Pages 165-174
Published Online: April 2013
Available online at http://pphmj.com/journals/ujmms.htm
Published by Pushpa Publishing House, Allahabad, INDIA

PARTICLE SWARM OPTIMIZATION ALGORITHM AND ITS APPLICATIONS IN STOCK MARKET

Guang He

Key Laboratory of Numerical Simulation of Sichuan Province
Neijiang, Sichuan 641100
P. R. China

School of Mathematics and Information Sciences
Neijiang Normal University
Neijiang, Sichuan 641100
P. R. China

Abstract

An improved optimization algorithm was designed for finding these solutions of discontinuous portfolio optimization models in stock market quickly and efficiently. By introducing crossover operations, an innovative particle swarm optimization algorithm (CPSO) based on optimal and sub-optimal locations was proposed. Then in performance test the algorithm performed better than some existed improved particle swarm optimization algorithms, and overcome the flaw of premature. Finally, CPSO was applied in stock market, and in simulation experiment optimization values of two portfolio models under different expected return rates were obtained.

2010 Mathematics Subject Classification: 90-XX.

Keywords and phrases: particle swarm optimization algorithm, crossover operation, portfolio, expected return rate, stock market.

Communicated by K. K. Azad

Received October 15, 2012

1. Introduction

Particle swarm optimization algorithm (PSO) was proposed by Kennedy and Eberhart [1] in 1995. It is a kind of evolutionary computation technique motivated by the behavior of organisms such as fish schooling or bird flocking. In the PSO system, a number of particles coexist and cooperate to get the optimal solution in complicated search space. Compared with other evolutionary algorithms, PSO has simple structure and very few parameters to adjust. Meanwhile it has shown a faster convergence rate on some high-dimension problems. However, when solving these problems with more local extreme values, PSO is easy to be trapped by poor local minima, and shows slower convergence rate in later iterations. In order to overcome these disadvantages, lots of relevant researches were derived (see [2-6] and the references therein).

During recent about six years, some studies on applying the standard PSO and its modified algorithms to the portfolio selection and relevant fields have been carried out gradually. Apart from solving the classical M-V model, Liu et al. [7] studied an optimal stocks portfolio model using PSO, Chen et al. [8] analyzed a class of portfolio models through the modified PSO, and other relevant researches on portfolio optimization based on CVaR [9] and sel-financing portfolio model [10], etc.

Motivated and inspired by the research work mentioned above, in this paper, firstly develop an improved particle swarm optimization algorithm based on crossover operation and optimal and sub-optimal locations. Then apply the improved algorithm to solve the model with real data from the Hong Kong Stock Market, and actual return rates of different portfolios are given in numerical experiments. Some main results in detail are showed later.

2. An Improved Particle Swarm Optimization Algorithm

In the standard PSO, each iteration position of every particle is obtained from two optimal positions according to its own and its neighboring-particle's experiences. Inspired by the idea of particle swarm optimization proposed in [6], consider optimal and sub-optimal positions of each particle in the process of iteration, meanwhile introduce crossover operation for overcoming premature. The improved particle swarm optimization algorithm is denoted by CPSO, and its process is as follows.

Suppose that the search space is D dimensional, and number of particle swarm is M. The past optimal and sub-optimal positions of ith particle of the swarm are represented by $P_i^1 = (P_{i1}^1, P_{i2}^1, ..., P_{iD}^1)$ and $P_i^2 = (P_{i1}^2, P_{i2}^2, ..., P_{iD}^2)$, respectively, and the global optimal and sub-optimal positions are represented by $P_g^1 = (P_{g1}^1, P_{g2}^1, ..., P_{gD}^1)$ and $P_g^2 = (P_{g1}^2, P_{g2}^2, ..., P_{gD}^2)$, respectively. Four iteration velocities are defined as follows:

$$V_{id}^1(t+1)$$
$$= wV_{id}(t) + c_1 r_{1i,d}(t)(P_{id}^1(t) - X_{id}(t)) + c_2 r_{2i,d}(t)(P_{gd}^1(t) - X_{id}(t)), \quad (1)$$

$$V_{id}^2(t+1)$$
$$= wV_{id}(t) + c_1 r_{1i,d}(t)(P_{id}^2(t) - X_{id}(t)) + c_2 r_{2i,d}(t)(P_{gd}^1(t) - X_{id}(t)), \quad (2)$$

$$V_{id}^3(t+1)$$
$$= wV_{id}(t) + c_1 r_{1i,d}(t)(P_{id}^1(t) - X_{id}(t)) + c_2 r_{2i,d}(t)(P_{gd}^2(t) - X_{id}(t)), \quad (3)$$

$$V_{id}^4(t+1)$$
$$= wV_{id}(t) + c_1 r_{1i,d}(t)(P_{id}^2(t) - X_{id}(t)) + c_2 r_{2i,d}(t)(P_{gd}^2(t) - X_{id}(t)), \quad (4)$$

where w is particle inertia weight, c_1 and c_2 are cognitive and social scaling parameters, respectively. From (1)-(4), get iteration formula of particle's velocity as follows:

$$V_i(t+1) = \{V_i^j \mid f(X_i + V_i^j) = \min_{1 \le k \le 4} f(X_i + V_i^k)\}, \quad (5)$$

and iteration formula of particle's position is given by

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1). \quad (6)$$

Next, consider iteration formulas of particle's past optimal and sub-optimal positions and global optimal and sub-optimal positions, which are seen in [6]. Then, the crossover operations in genetic algorithm are introduced for strengthening ability of searching for the best solution.

Make two-point crossover operation on particle's past optimal and sub-optimal positions, its process is as follows:

$$\begin{aligned}
&d_1 = [D * r_0];\\
&d_2 = [D * r_0];\\
&\mathit{for}\ \ m = d_1\ \mathit{to}\ d_2\\
&\qquad \mathit{swap}(Z_{im}, Z_{jm});\\
&\mathit{end}
\end{aligned} \tag{7}$$

where D is the dimension of particle swarm, r_0 is a random number in the interval $(0, 1)$, $[\square]$ denotes that round toward infinity, and $\mathit{swap}(Z_{im}, Z_{jm})$ denotes that exchange mth positions of Z_i and Z_j.

Then, do algebraic crossover operation on global optimal and sub-optimal positions, and its formula is:

$$\begin{cases} Y_i^1 = aY_i + (1-a)Y_j, \\ Y_j^1 = aY_j + (1-a)Y_i, \end{cases} \tag{8}$$

where a is a random number in the interval $(0, 1)$, Y_i and Y_j are two father particles, and correspondingly Y_i^1 and Y_j^1 are two son particles. Then keep particles with better fitness values by comparing farther and son particles.

Now, demonstrate the detailed algorithm flow of CPSO as follows:

Step 1: Initialize positions and velocities of all particles, and set iteration count $t = 0$;

Step 2: Evaluate the fitness values of current particles;

Step 3: Update optimal and sub-optimal positions of each particle, and global optimal and sub-optimal positions;

Step 4: Do two-point crossover operation on particle's past optimal and sub-optimal positions obtained in Step 3 by (7), and make comparison and replacement;

Step 5: Do algebraic crossover operation on global optimal and sub-optimal positions obtained in Step 3 by (8), and make comparison and replacement;

Step 6: Update iteration velocity and position of each particle by (1)-(6), and $t = t + 1$;

Step 7: Turn to Step 2 when t is less than the maximal iteration count, unless end the process.

3. Algorithm Tests and Analyses

Here, choose the following functions to test the performance of CPSO:

(1) Rosenbrock:

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \quad x_i \in [-30, 30].$$

The function reaches global minimal value 0 when $x_i = 1, i = 1, 2, ..., n-1$.

(2) Rastrigin:

$$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10), \quad x_i \in [-5.12, 5.12].$$

The function reaches global minimal value 0 when $x_i = 0, i = 1, 2, ..., n$.

(3) Griewank:

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad x_i \in [-600, 600].$$

The function reaches global minimal value 0 when $x_i = 0, i = 1, 2, ..., n$.

In the system of CPSO, take $c_1 = c_2 = 2,$ and get w as follows:

$$w(t) = w_{\min} + \frac{T - t}{T - 1} \cdot (w_{\max} - w_{\min}),$$

where T and t are maximal and present iteration counts, respectively, and $w_{\max}$ and $w_{\min}$ are maximal and minimal values of inertia weight, respectively, then set $w_{\max} = 0.9,\ w_{\min} = 0.4.$

Then, choose PSO, IPSO developed in [5], OSL-PSO proposed in [6] and CPSO as test algorithms, and compare their optimization results. Take $D = 10,\ \ T = 1000$ and $M = 20, 40, 80, 160,$ respectively, and calculate each selection for 50 times independently, then average them. The related results are showed in Table 1.

Table 1. Optimization results of three test functions

Function	M	PSO	IPSO	OSL-PSO	INPSO
Rosenbrock	20	21.3351	10.5172	6.0433	1.3117
	40	15.6461	1.2446	0.1500	0.1030
	80	1.6683	0.1922	0.0230	0.0066
	160	1.1445	0.0598	1.1348E-12	0.0001
Rastrigin	20	6.6214	3.2928	0.9505	0.5970
	40	2.7401	2.6162	0.1281	0.0995
	80	2.3373	1.7054	0.0249	0.0000
	160	1.4410	0.8001	2.2204E-15	0.0000
Griewank	20	0.0905	0.0784	0.0332	0.0301
	40	0.0760	0.0648	0.0097	0.0295
	80	0.0649	0.0594	0.0006	0.0167
	160	0.0566	0.0507	0.0000	0.0102

From data in Table 1, it is seen that CPSO outperforms significantly PSO and IPSO in tests of three functions. Compared with OSL-PSO, except

$n = 160$ optimization values of CPSO are better than those of OSL-PSO in test of Rosenbrock function; in test of Griewank function, CPSO obtains better value when $n = 20$; and in test of Rastrigin function, CPSO performs much better than OSL-PSO. To sum up, when dimension of search space $D = 10$, CPSO displays effective optimization ability.

Furthermore, in order to test the operation speed of CPSO, compare it with that of the standard PSO. Take $M = 40$, and calculate each value for 50 times independently, and average them. These results are showed in Table 2.

Table 2. Time complexities of two algorithms (second)

Function	Algorithm	Best value	Worst value	Mean value
Rosenbrock	PSO	6.2340	6.3590	6.2929
	INPSO	6.7707	7.6040	7.2365
Rastrigin	PSO	6.2030	6.5000	6.3422
	INPSO	6.9377	7.4863	7.1478
Griewank	PSO	6.6880	6.9850	6.8500
	INPSO	7.7810	8.1880	8.0219

It follows from Table 2 that PSO runs faster than CPSO in all tests, but the differences between them are small. Moreover, from results showed in Table 1, it is worth that CPSO improves optimization ability significantly by adding a little running time.

4. Applications in Stock Market

In this section, CPSO will be applied in stock market by numerical simulation experiments.

Assume that an investor owns a wealth M_0, which is going to be invested in *m* stocks S_i, $i = 1, 2, ..., m.$ Let R_i be the random return rate of S_i, and r_i be the mathematical expectation of R_i. Define the expected return of a portfolio $x = (x_1, x_2, ..., x_m)$ as follows:

$$E\left[\sum_{i=1}^{m} R_i x_i\right] = \sum_{i=1}^{m} r_i x_i.$$

The experiments use the real data from the Hong Kong Stock Market. The closing prices of 10 blue-chip stocks dated from 6/18/2008 to 6/18/2010 are collected. Set the number of the trading days in the investment period to be 60, and the number of trading days prior to the investment day to be 80. Two classical optimization models: Konno's model and Cai's model are chosen, and the feasible region of $x = (x_1, x_2, ..., x_m)$ is

$$S = \left\{ x \,\middle|\, \sum_{i=1}^{m} r_i x_i \geq \rho M_0,\ \sum_{i=1}^{m} x_i = M_0,\ 0 \leq x_i \leq \mu_i,\ i = 1, 2, ..., m \right\},$$

where ρ denotes the expected return rate.

Konno's model:

$$\min E\left| \sum_{i=1}^{m} R_i x_i - E\left(\sum_{i=1}^{m} R_i x_i\right) \right|$$

$$\text{s.t.} \quad x \in S,$$

Cai's model:

$$\min \max_{i=1, 2, ..., m} E| R_i x_i - r_i x_i |$$

$$\text{s.t.} \quad x \in S.$$

It is easy to see that the risk function in Cai's model is discontinuous, so consider applying CPSO to solve it. In simulation experiments, set $M_0 = 1,$ the dimension of search space $D = 10,$ the number of maximal iteration $T = 100,$ and the number of swarm $n = 40.$ These optimization results of two models under short-term expected return rate (5%) and long-term expected return rate (20%) are listed in Table 3 and Table 4, respectively.

Table 3. Optimization values under short-term expected return rate

Algorithm	Konno's model	Cai's model
PSO	0.0916	0.0499
CPSO	0.0459	0.0228

Table 4. Optimization values under long-term expected return rate

Algorithm	Konno's model	Cai's model
PSO	0.2664	0.1748
CPSO	0.1003	0.0429

From data in Table 3 and Table 4, when expected return rate is given, the values at risk calculated by CPSO are all smaller than those calculated by PSO, and CPSO performs better in risk control. Moreover, the values at risk of Cai's model are less than those of Konno's model, which indicates that the former could reduce more losses while gain same return rate.

5. Conclusions

In order to avoid premature, a modified particle swarm optimization algorithm (CPSO) is developed based on crossover operations and the particle's optimal and sub-optimal locations. By algorithm tests and analyses, CPSO performs better than some existed algorithm, and avoid being trapped by poor local minima. In simulation experiments, CPSO is applied to solve optimization values under different expected return rates of two portfolio models. Furthermore, how to combine other intelligent algorithms to improve ability is the next work.

References

[1] J. Kennedy and R. C. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks 4 (1995), 1942-1948.

[2] J. F. Chang and S. W. Hsu, The construction of stocks portfolios by using particle swarm optimization, Proceedings of the Second International Conference on Innovative Computing, Information and Control, 2007, 390.

[3] B. Niu, Y. L. Zhu and X. X. He, Construction of fuzzy models for dynamical systems using multiple cooperative particle swarm optimizer, Lecture Notes in Artificial Intelligence 3613 (2005), 987-1000.

[4] D. Z. Pan, An improved particle swarm optimization algorithm, Communications in Computer and Information Science 1237 (2011), 550-556.

[5] Y. Jiang, T. Hu, C. Huang and X. Wu, An improved particle swarm optimization algorithm, Appl. Math. Comput. 193(1) (2007), 231-239.

[6] D. Z. Pan, Research on particle swarm optimization algorithm and its application in oil field development planning, Doctoral Dissertation of Southwest Petroleum University, 2011.

[7] X. F. Liu, T. Chen and L. Y. Zhang, Study on the portfolio problem based on particle swarm optimization, J. Systems & Management 17 (2008), 221-224.

[8] W. Chen, R.-T. Zhang and L. Yang, Portfolio selection model based on the improved particle swarm optimization, Computer Science 36(1) (2009), 146-147.

[9] S. H. Zhou and Y. S. Ni, Research on improved particle swarm optimization algorithm for portfolio optimization based on CVaR, J. Wuhan Univ. Tech. 32 (2010), 179-182.

[10] Y. M. Liu, Q. Z. Zhao and B. Niu, Constrain particle swarm optimizer for solving sel-financing portfolio model, Math. Practice and Theory 41 (2011), 78-84.