Improved Particle Swarm Optimization and Application

Yue Li [a], Xiyu Liu [b]

School of Management Science and Engineering, Shandong Normal University, Jinan 250014, China

[a]liyuemoon11@163.com, [b]sdxyliu@163.com

Abstract

Particle swarm optimization algorithm is an excellent algorithm for solving optimization problems. Inertia weight and learning factor in particle swarm optimization algorithm adjust lonely, which can weaken the unity of evolution algorithm. It is difficult to adapt to complex nonlinear optimization, this paper puts forward a new kind strategy to strengthen the correlation between inertia weight and learning factor, which can control learning factors based on dynamic inertia weight. Through control the dynamic adjustment of inertia weight and learning factors to balance the global exploration and local development ability of the algorithm. In addition, this paper adopt the time factor as the weight of the linear function, in order to further improve the local development ability and faster convergence speed in the late iteration. Consider the contradiction between convergence and diversity of particle swarm algorithm, this paper puts forward the strategy that limiting the boundary position and adjusting the position to the symmetrical position when it beyond the border, to avoid particle flying away from area resulting in a decrease of species diversity, as well as making particles quickly converge to global optimal. This proposed DPSOPS can effectively improve the convergence rate and convergence speed compare with formal PSO. Applied to the optimization of base station planning issues, the higher coverage and more optimized distribution strategy can be found under the condition of a given.

Keywords

Particle swarm optimization; Base station planning; DPSO; Optimization problem.

1. Introduction

Particle swarm optimization algorithm was proposed by Kennedy and Eberhart in 1995[1], simply called PSO, originated in the research of feeding behavior of birds and fish. Because of its versatility and simple algorithm formula, and have strong ability of global optimization, the PSO quickly become a powerful tool to solve difficult problems. It has been widely used in electric power system optimization, clustering analysis, neural network training, and many other fields. Although, the particle swarm optimization has so many advantages, in the late iteration algorithm, the algorithm is easy to fall into local optimum because of the reduction of species diversity; we also call this phenomenon of premature convergence[2].

Many researchers have done a lot of work to improve the basic algorithm, there is many variations of it have been proposed. Besides inertia weight improvement strategy, many scholars introduced other algorithms into the particle swarm optimization. Inertia weight and learning factor in particle swarm optimization algorithm adjust lonely, which can weaken the unity of evolution algorithm. It is difficult to adapt to complex nonlinear optimization, this paper puts forward a new kind strategy to strengthen the correlation between inertia weight and learning factor, which can control learning factors based on dynamic inertia weight. Through control the dynamic adjustment of inertia weight and learning factors to balance the global exploration and local development ability of the algorithm. In addition, this paper adopt the time factor as the weight of the linear function, in order to further improve the local development ability and faster convergence speed in the late iteration. Consider the contradiction between convergence and diversity of particle swarm algorithm, this paper puts forward the strategy that limiting the boundary position and adjusting the position to the symmetrical position when it beyond the border, to avoid particle flying away from area resulting in a decrease of species diversity, as well as making particles quickly converge to global optimal. Applied to the optimization of base station planning issues, the higher coverage and more optimized distribution strategy can be found under the condition of a given.

2. DPSOPS Algorithm

This section will present PSO briefly. Then improved the original algorithm and design the new algorithm DPSOPS.

2.1 PSO

Particle swarm optimization (PSO) is swarm intelligence based metaheuristic algorithm proposed by Kennedy and Eberhart. High decentralization, cooperation among the particles and simple implementation make PSO efficiently applicable to optimization problems.

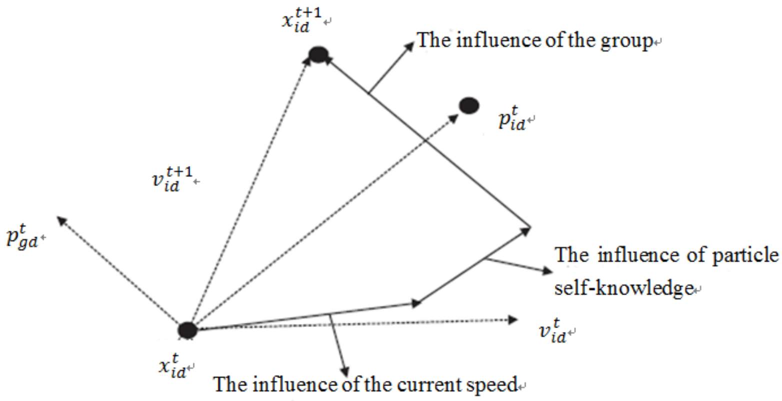

Fig. 1 The updating of particle position

In PSO, each individual in the swarm, called a particle. It has three main components, particles, social and cognitive components and the velocity of the particles. Cognitive learning is represented by personal best and social learning is represented by the global best value. The p_i solution is the best solution the particle has ever achieved in its history, the p_g value is the best position the swarm has ever achieved. The swarm guides the particle using parameter p_g. Together cognitive and social learning are used to calculate the velocity of particles to their next position (Fig. 1 is the updating of particle position). Each particle updates its speed and position, according to the following iterative formula, and produces a new particle of the next generation.

$$\begin{cases} \Delta x_{id}^{t+1} = w\Delta x_{id}^{t} + c_1 rand1()(p_{id}^{t} - x_{id}^{t}) + c_2 rand2()(p_{gd}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + \Delta x_{id}^{t+1} \end{cases} \quad (1)$$

The ith particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, C_1 and C_2 are positive constants, rand1( ) and ran*d* 2( ) are random numbers between 0 and 1,D is the dimension of the search space. The pBest solution is $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ .The pBest solution is $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ and the position change is represented as $\Delta X_i = (\Delta x_{i1}, \Delta x_{i2}, \ldots, \Delta x_{iD})$.

2.2 Improved focus

2.2.1 The setting of inertia weight

Inertia weight of PSO algorithm is an extremely important parameter, the setting of it directly related to the algorithm's ability to develop and explore. Greater weight value will help improve the global search ability of the algorithm, which can avoid premature caused by the local minimal. Smaller weight value helps to achieve accurate search of search area, improve the convergence precision [3].Although linear decreasing weight can balance the global exploring ability and local search ability to some extent, its value shows the tendency of a linear gradient with the increase of the number of iterations, which will cause the local search ability becomes poor in the later. So set it as nonlinear changes. Paper [4] using the Sugeno function in fuzzy theory to produce a set of decline curve that changes from large to small, and using them as alternative choice of inertia weight curve in particle swarm optimization. Experiments point out that the function curve is conducive to expand the search space through the change of weight at the beginning of the iteration, as well as avoid premature convergence. However the algorithm is not stable, as a poor convergence results at the end especially. After many experiments, this paper uses the following function to control the inertia weight, which can make each particle group has its unique inertia weight value:

$$w = w_{max} - (w_{max} - w_{min})(t^n)/T_{max}^n \tag{2}$$

The new algorithm uses multiple inertia weight value at the same time, instead of using a single inertia weight value. In the process of iterative evolution, it can gradually eliminate poor weight value and leave the weight value with good effect, so as to adapt to solve a variety of functions without the need for a preliminary analysis and selecting weights.

A contrast experiment research was made among the new algorithm, PSO algorithm with the strategy of the linear decreasing deciding inertia weight, the nonlinear decreasing PSO algorithm with Sugeno function curve and the improved PSO algorithm with new dynamic weight curve [5]. The key parameters in the algorithm with new inertia weight is n, this paper values it as {0.16,0.3,0.6,1,2，3,5,9}. Iterative 400 times, the curve is shown in Fig. 2.

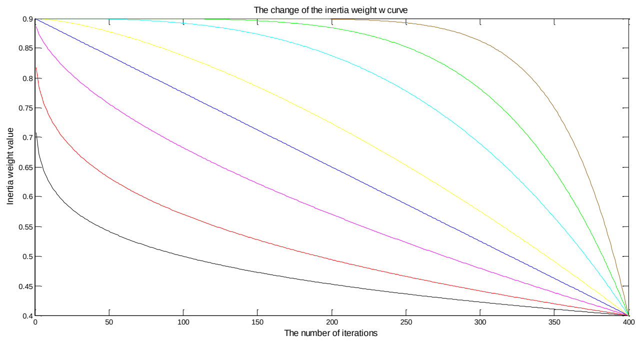

Fig. 2 The curve of inertia weight

This paper selects the function $f(x,y) = \frac{\sin\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}} + e^{\frac{\cos 2\pi x+\cos 2\pi y}{2}} - 2.7289$. The maximum weight is 0.9. The minimum weight value is 0.4. The experiment iterates 400 times respectively. Results is shown in Fig. 3 the new algorithm, PSO algorithm with the strategy of the linear decreasing deciding inertia weight, the nonlinear decreasing PSO algorithm with Sugeno function curve and the improved PSO algorithm with new dynamic weight curve [5] are colored in black, pink, blue and green. The contrast experiment results shows that it has stronger global searching ability compared with the PSO algorithm with traditional nonlinear decreasing strategy; and converges to a better result compared with the other two types PSO algorithm.

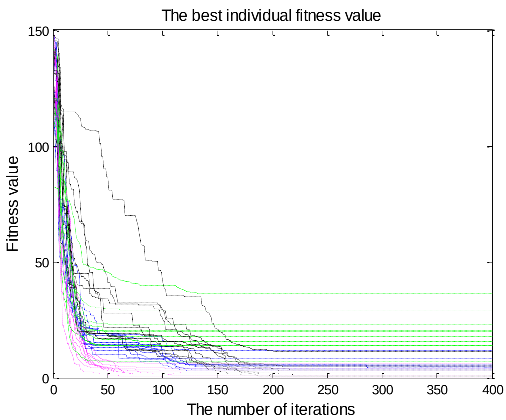

Fig. 3 The optimal curve of three algorithms

2.2.2 Asynchronous learning factor

In the process of optimization iteration, two learning factors c_1 and c_2 changes with time called asynchronous. The values of two learning factors determine the influence of the particles self-knowledge and social cognition on particle trajectories, which can reflect the degree of information exchange between particles in groups. Larger c_1 value would make the particles hovering at local scope too much, and larger c_2 value would lead to particle premature convergence to local minimum [6].This paper sets the learning factors change according to the following expression:

$$c_1 = c_{1_s} - (c_{1_s} - c_{1_e}) \times w$$
$$c_2 = c_{2_s} + (c_{2_e} - c_{2_s}) \times w \qquad (3)$$

Explain that c_{1_s} and c_{2_s} are the initial value of learning factors c_1 and c_2 respectively, c_{1_e} and c_{2_e} are the termination value of learning factors c_1 and c_2 respectively.

This set is conducive to realize the quick search and strengthen the global search ability at the beginning of the optimization iterations, as the particle's social learning ability weaker and self-learning ability stronger. In the later stages of the optimization iterations, the particle's self-learning ability is weaker and the ability of social learning is stronger, which is advantageous to the local fine search and converging to global optimal solution with higher accuracy.

According to the last section, the corresponding changing curves of the learning factors can be shown as the Fig.4.

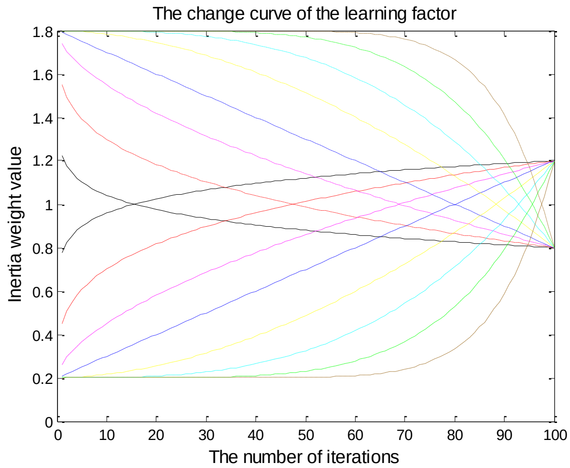

Fig.4 The change curve of the learning factor

2.2.3 Asynchronous time factor

After a careful study of the standard PSO algorithm, and found that the location of particle is updated in the original position plus the velocity of the current time as the location of the next time. In physics concept,, only the two physical quantities with same dimension can be direct process, that only displacement and displacement can be operated as well as velocity and velocity can be operated. However, the position and velocity are operated directly in the new the position and velocity update formula, so there is implicit time factor in the position and velocity update formula. The traditional particle swarm algorithm update each particle position where time factor is a fixed constant of 1, which is an important factor leading the particles oscillate back and forth in the vicinity of the optimal solution [7].

$$\begin{cases} \Delta x_{id}^{t+1} = w\Delta x_{id}^{t} + c_1 rand1()(p_{id}^{t} - x_{id}^{t}) + c_2 rand2()(p_{gd}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + T\Delta x_{id}^{t+1} \end{cases} \qquad (4)$$

Experiments show that choosing an appropriate T time factor plays a positive role of convergence of the algorithm. This paper construct a linear time factor as$T = 0.3 + w$, when T = 1, the formula (4) is the original formula (1); When the T Is not equal to 1, the time factor value is bigger in the early iterations of the algorithm that is conducive to achieve rapid global search. Along with the iteration, time factor is more and more small. Smaller time factor is advantageous to the local fine search that is easy to find the global optimal point and high precision.

2.2.4 Speed and boundary limit policy

When there is large distance between the particles current position from individual optimal position and the global optimal position, the velocity update step length must be larger. In this case, the velocity of the particles updated by formula (1) must be very large, which will cause particle flying away from solving area and decrease of species diversity. Moreover, excessive velocity also can make particles can't quickly converge to global optimal. But such situation can be avoided by the velocity limit [8]. Maximum velocity limit determines the resolution of the area from the current position to the best position, which equivalent to the accuracy of the solving interval.

$v_{id}^{max} = \rho(x_{max} - x_{min}), \rho \in (0,1)$ is the maximum velocity of the particle in dimension d. To further improve the performance of PSO algorithm and prevent particles fly away from search region which can result in reducing of population diversity, this paper increases position symmetric recall strategy when beyond the border. When particles fly off border, the position of particle can be adjusted to the location of the symmetrical about border. The adjust rule can be expressed as:

$$x_{id}^{t+1} = \begin{cases} x_{id}^{t} + T\Delta x_{id}^{t+1}, x_{id}^{t} + T\Delta x_{id}^{t+1} \in [x_{min}, x_{max}] \\ 2x_{min} - (x_{id}^{t} + T\Delta x_{id}^{t+1}), x_{id}^{t} + T\Delta x_{id}^{t+1} < x_{min} \\ 2x_{max} - (x_{id}^{t} + T\Delta x_{id}^{t+1}), x_{id}^{t} + T\Delta x_{id}^{t+1} > x_{max} \end{cases} \quad (5)$$

Explain that x_{max} and x_{min} are the upper and lower bounds of search space respectively.

2.2.5 Reverse learning

This improve inspired by NPSO algorithm[9], which is a modified form of the basic PSO and was introduced in 2005, each particle adjusts its position according to its own previous worst solution and its group's previous worst to find the optimal value. The strategy here is to avoid a particle's previous worst solution and its group's previous worst based on similar formula of the basic PSO[10].

Following Eberhart and Kennedy's naming conventions; D is the dimension of the search space.

The ith particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, its previous worst is $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$. The index of the worst in the whole group is g and the position change is represented as $\Delta X_i = (\Delta x_{i1}, \Delta x_{i2}, \dots, \Delta x_{iD})$.

Each particle updates its position with the following two equations:

$$\begin{cases} \Delta x_{id}^{t+1} = \Delta x_{id}^{t} + c_1 rand1()(x_{id}^{t} - p_{id}^{t}) + c_2 rand2()(x_{id}^{t} - p_{gd}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + \Delta x_{id}^{t+1} \end{cases} \quad (6)$$

Where c_1 and c_2 are positive constants, rand1 ( ) and ran*d* 2( ) are random numbers between 0 and 1.

When the optimal position don't updated within a certain iteration steps, argues that algorithm has ground to a halt. This paper creates reverse learning to help the particles jump out of local optimum. Firstly selecting N' particles in current population ($N' = N/4$,φ is the number of parameter value that contained by inertia weight control function), the N' particles run T' times reverse learning; the other $(N - N')$ particles remain positive learning as usual. The ith particle that process reverse learning learn from the worst position of the particle, which can be recorded as $q_i = (q_{i1}, q_{i2}, \dots, q_{iD})$. The velocity of ith particle that process reverse learning updates follows the formula:

$$\Delta x_{id}^{t+1} = w\Delta x_{id}^{t} + c_3 rand1()(x_{id}^{t} - q_{id}^{t}) + c_4 rand2()(x_{id}^{t} - q_{gd}^{t}) \quad (7)$$

Learning factors $c_3 = 0.7$、$c_4 = 0.3$, and the number of reverse learning iteration is $T' = T/10$. q_{id}^{t} refers to the d dimension values of the worst position that the ith particle has got iterate to t generation. q_{gd}^{t} refers to the d dimension values of the global worst position that all particle has got iterate to t generation.

2.3 The new algorithm: DPSOPS

2.3.1 Design of DPSOPS

According to the section 2.2 comprehensive, this paper put forward an improved particle swarm algorithm named DPSOPS, namely the particle swarm algorithm with learning factors and time factor adjusted according the inertia weight and velocity limit and position symmetric recall strategy. The inertia weight value is dynamic optimal evolution according to the given initial parameter $\{n_1, n_2, \dots n_\varphi\}$, the velocity and position update formula as follows:

$$\begin{cases} \Delta x_{id}^{t+1} = w\Delta x_{id}^{t} + c_1 rand1()(p_{id}^{t} - x_{id}^{t}) + c_2 rand2()(p_{gd}^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = \begin{cases} x_{id}^{t} + T\Delta x_{id}^{t+1}, (x_{id}^{t} + T\Delta x_{id}^{t+1}) \in [x_{min}, x_{max}] \\ 2x_{min} - (x_{id}^{t} + T\Delta x_{id}^{t+1}), (x_{id}^{t} + T\Delta x_{id}^{t+1}) < x_{min} \\ 2x_{max} - (x_{id}^{t} + T\Delta x_{id}^{t+1}), (x_{id}^{t} + T\Delta x_{id}^{t+1}) > x_{max} \end{cases} \end{cases} \quad (8)$$

Among them:

$w = w_{max} - (w_{max} - w_{min})(t^n)/T_{max}^n, n \in \{n_1, n_2, \dots n_\varphi\}$

$c_1 = c_{1s} - (c_{1s} - c_{1e}) \times w$

$c_2 = c_{2s} + (c_{2e} - c_{2s}) \times w$

$v_{id}^{max} = \rho(x_{max} - x_{min}), \rho \in (0,1)$

$T = 0.3 + w$

If $p_{gd}^{t} = p_{gd}^{t+1} = \dots = p_{gd}^{t+m} (m > T/10)$, then select N' particle run T'times as

$$\Delta x_{id}^{t+1} = w\Delta x_{id}^{t} + c_3 rand1()(x_{id}^{t} - q_{id}^{t}) + c_4 rand2()(x_{id}^{t} - q_{gd}^{t}) \quad (9)$$

$N' = N/4, c_3 = 0.7, c_4 = 0.3, T' = T/\varphi$.

2.3.2 The procedure of DPSOPS algorithm

Algorithm is the main computational steps are as follows:

Step1: Setting right weight values of minimum and maximum Inertia weight w_{max} and w_{min}, the initial value and the end value of learning factor c1 and c2, and values of c3 and c4, the maximum evolution times N. The initial position and velocity of every particle are setting according to the uniform distribution within the domain. Individual optimal initial position p_i is stetted as the initial position; the optimal location for the whole group p_g is stetted as the best fitness value of particles of the position.

Step2: Calculate each particle's fitness, comparing the particle's fitness value and the individual optimum value p_i. If the value of current particle is better than p_i, then set the individual optimal value as the current particle.

Step3: Compare the particle fitness populations and the global optimal valuep_g. If the value of current particle is better than p_g, then set it as the global optimal value of population.

Step4: Check whether there is only one strategy of inertia weight, if not, turn to the next step. If so, turn to Step7.

Step5: Optimize the strategy of inertia weight that group with worst fitness value, change the it as the strategy that the best fitness value groups using.

Step6: Update the particle's speed and position according to the formula (8).

Step7: Check whether group optimal value changes. If there were no changes in successive $m(m > T/10)$generations, then go to the next step; otherwise, turn to Step9.

Step8: Update the particle's speed and position according to the formula (9).

Step9: Check the end conditions. If achieve the optimal maximum number of iterations then stop the algorithm, or transferred to Step2.

3. Applied to the optimization of base station planning

Base station planning is aimed to make the wireless network meet the needs of user that under the coverage area; for this purpose choosing the location quantity and configuration parameters of base station reasonably. As we know, a reasonable planning of base station can't do without a reasonable programming model. Before planning should choose appropriate mathematical methods describe the distribution of the user or business coverage area distribution, clear quality requirements of users. In this paper, mathematical model is established, and then using dynamic particle swarm algorithm (DPSO) and improved algorithm to solve.

3.1 Mathematical model

This paper established mathematical model is:

$$\min C = |\Omega| \tag{10}$$

$$s.t.$$

$$\frac{P_{ij} L_{ij} x_i}{\beta_{ij} P_{ij} L_{ij} x_i + \left(\sum_{i \neq j, i \in \Omega} P_{max} L_{ij}\right) + N_{x\_th}} \geq \Upsilon_{\overline{F}}, i \in \Omega, j \in J, k = \{0,1,2\} \tag{11}$$

$$\frac{P_0 x_i}{V_{AF} * (n_i - 1)(1 + F) P_0 + N_{x\_th}} \geq \Upsilon_{\underline{F}}, i \in \Omega \tag{12}$$

$$\sum_{i \in \Omega} x_i \left(\frac{s_i^k}{s^k}\right) \geq \Upsilon^k, \tag{13}$$

$$n_i \leq n_{max} * \eta^k, i \in \Omega, k = \{0,1,2\} \tag{14}$$

$$\mathrm{L}_{ij} = \begin{cases} f(\mathrm{d_{ij}}) & x_i = 1 \\ 0 & x_i = 0 \end{cases}, i \in \Omega \tag{15}$$

$$P_{ij} = \frac{\left(\frac{E_b}{N_0}\right)_{\overline{F}}}{PG} \left[P_{max}\left(\beta_k + \sum_{i \neq j, i \in \Omega} \frac{L_{ij}}{L_{ij}}\right) + N_{x\_th} L_{ij}\right], i \in \Omega, k = \{0,1,2\} \tag{16}$$

$$x_i \in \{0,1\} \tag{17}$$

In which the objective function of formula (10) and (11) express respectively the least number of base station and the maximum coverage. Formula (12) express down link SNR constraint, which can ensure the quality of communication, as the signal that launch to the user base station can be successfully demodulation. Formula (13) express the minimum requirements set of each density areas, the sum of effective regional coverage ratio ruling out the repeat points covering user , if simple addition probably won't be covering the requirements. Formula (14) express the uplink station district general user ni must not exceed the allowed maximum number of users. Formula (15) and (16) express the link loss and down-link power calculation respectively. Formula (17) said the decision variables.

Fig. 5 Sample terrain

3.2 Contrast experiment and analyses

This paper has carried on the contrast experiment between the improved algorithm in this paper and dynamic particle swarm optimization algorithm. Select a 10 km*7 km developed mountainous area of a city as sample terrain, which is shown in Fig. 5:

We coded the algorithm with MATLAB R2014a in windows 10 system. Each part of the experiments was repeated 50 times, and then takes the average of coverage ratio. The results are shown in Table 1.

Table 1 The result of contrast experiment

Algorithm	Optimal coverage ratio	Average coverage ratio	Average running time
DPSO	88.74%	83.32%	162.3s
DPSOPS	93.62%	89.21%	118.4s

Using DPSO algorithm run several times, calculate the average coverage ratio is 83.32%, the example that can represent the average level is shown in Fig. 6 and Fig. 7.

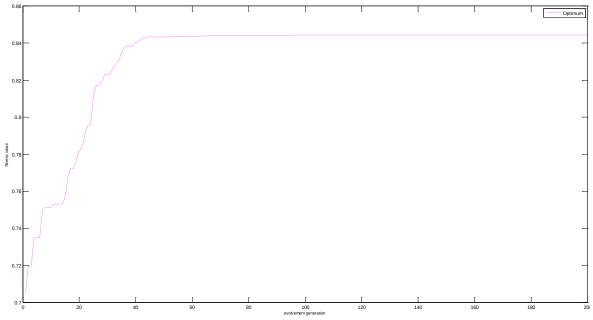

Fig. 6 The optimal curve of DPSO

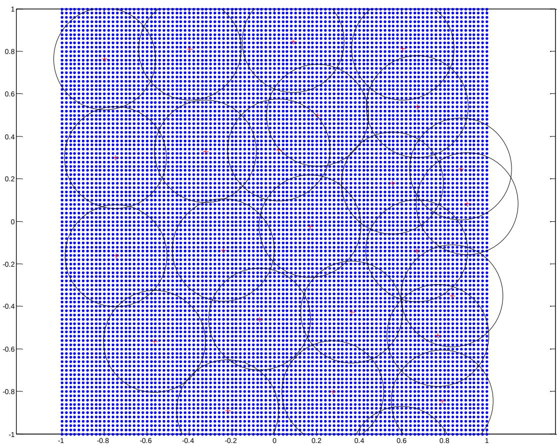

Fig. 7 The optimal layout diagram

The DPSO example achieved the final coverage ratio is 84.43%.The optimal curve in Fig. 6 can show that the algorithm run into the local optimum lead to early maturity. Fig. 7 shows out the early maturity as repeat coverage on the left side of the region. The dynamic particle swarm algorithm can't achieve high coverage ratio.

Using DPSOPS algorithm run several times, calculate the average coverage ratio is 89.21%, the example that can represent the average level is shown in Fig. 8 and Fig. 9.

The DPSOPS example achieved the final coverage ratio is 89.33%.The optimal curve in Fig. 9 can show that the algorithm avoid early maturity in certain extent. Fig. 10 shows out the optimal layout diagram which is more reasonable than another, but it can be improved to avoid repeat coverage and higher coverage.

On the whole, the new algorithm DPSOPS has higher coverage and more optimized distribution strategy.

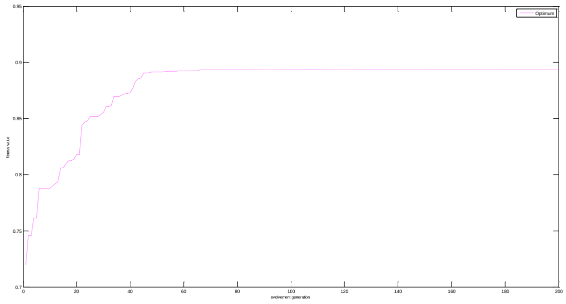

Fig. 8 The optimal curve of DPSOPS

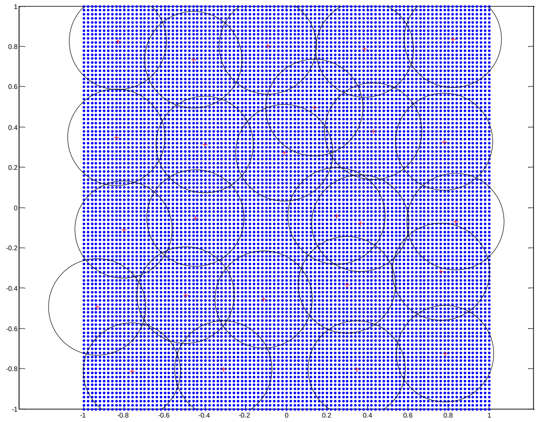

Fig. 9 The optimal layout diagram

4. Conclusion

This paper proposes a novel evolutionary algorithm, DPSOPS, based on the concepts and principles of particle swarm optimization and NPSO. This algorithm is characterized by employing a variety of value strategy and the evolutionary principles of PSO and NPSO at the same time. Its effectiveness and validity are verified by experiment. Applied to the optimization of base station planning issues, the higher coverage and more optimized distribution strategy can be found under the condition of a given.

Also notes that the new PSO algorithm can be more perfect. It can be combining with the membrane computing to further improve, the algorithm this also is later research.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61170038, 61472231 and 61502283), Jinan City independent innovation plan project in Colleges and Universities, China (201401202), Ministry of education of Humanities and social science research project, China (12YJA630152), Social Science Fund Project of Shandong Province, China (11CGLJ22).

References

[1] Eberhart, R. C., and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43. Piscataway, NJ: IEEE Service Center1995.

[2] Duan XD, Gao HX, Zhang XD Relations between population structure and population diversity of particle swarm optimization algorithm. Comput Sci 2007, 34(11):164–167

[3] Zhou Xihu, Gao XB. Particle swarm optimization algorithm with the time factor [J]. Journal of textile university journal of basic science, 2011, 24 (2): 303-308.

[4] Lei Kaiyou. Particle swarm optimization algorithm and its application research [D]. Southwest university,2006.
[5] Xu W, Liu X. One Level Membrane Structure P System Based Particle Swarm Optimization[C]. International Conference on Human Centered Computing. Springer International Publishing, 2016: 936-941.
[6] Sun Yongguang. Sheet forming structure crashworthiness optimization design and key technology research [D]. Changsha: Hunan University, 2011
[7] Kong Xianren, and Qin Yuling. Particle swarm algorithm with flying factor correct aluminum honeycomb sandwich panel model [J]. Journal of computer applications, 2010, 30 (3): 786-788.
[8] Wu Zhengke, Yang Qingzhen. Particle swarm algorithm with limited velocity and release strategy [J]. Computer application research, 2013, 30 (3): 682-683687
[9] Yang C. and Simon, D., "A new particle swarm optimization technique", in Proc. IEEE Int. Conf. on Systems Engineering, pp. 164-169, 2005. DOI: 10.1109/ICSENG.2005.9.
[10] Du P, Xiang L, Liu X. P system based particle swarm optimization algorithm [M].Frontier and Future