Partner Selection of Virtual Enterprise by Improved PSO with Dynamic Inertia Weight for Iterations

Junfeng Zhao¹,⁎ and Xinyi Huang²

¹School of Mechanical and Electrical Engineering, Guangdong Polytechnic of Industry and Commerce, Guangzhou 510510, Guangdong, China
²School of Mathematics, South China University of Technology, Guangzhou 510640, Guangdong, China

*Corresponding author: 0001220039@gdgm.edu.cn

Abstract. PSO is easy to fall into local optimum, and has the defects of low convergence accuracy and difficult convergence. In order to solve this defect and improve the accuracy of the algorithm, the modified PSO algorithm by Geng is applied to solve virtual enterprise partner selection with normalization, by changing dynamically inertia weight for the number of iterations. Furthermore, for the particles with small fitness, a large inertia weight is set to accelerate the approach of particles like the optimal solution region. Finally, results of numerical example show that the search speed of improved PSO is faster than that of standard PSO, which indicates that adding dynamic inertia weight can greatly shorten the search time of the algorithm and accelerate the convergence.

Keywords: Partner Selection, Particle Swarm Optimization (PSO), Inertia Weight, Global Extremum, Iteration Time

1. Introduction

In order to quickly occupy the market, how to quickly and accurately choose partners to establish a virtual enterprise has become particularly important. Particle swarm optimization algorithm is also commonly used to solve multi-objective decision-making, but the particle swarm optimization algorithm has memory. In the process of searching the optimal solution, the search speed and direction are adjusted according to the extreme value, and the convergence speed is faster. This article uses the particle swarm optimization algorithm (PSO) to study the selection of virtual enterprise partners.

Ben [1] et al. did some research on the integrated fuzzy ANP-MOP approach for partner selection problem and order allocation optimization: the case of virtual enterprise configuration. Nyongesa [2] et al. introduced the partner selection and performance evaluation framework for a construction-related virtual enterprise: a multi-agent Systems approach. Selection algorithm based on task-resource assignment graph and partner selection integrating LINMAP and TOPSIS were presented by Jia [3] and Dong [4]. Zhang [5] et al. committed to fuzzy cognitive map approach for trust-based partner selection. Huang and Sun [6] committed to the research on modeling method and partner selection for collaborative production of virtual enterprises. Zhang [7] el al. addressed on the green partner
selection based on pareto genetic algorithms. Hsieh [8] et al. dealt with virtual enterprises partner selection based on reverse auctions. Jia [9] et al. dis some work partner selection of virtual enterprise based on rough set and genetic algorithm, the improved algorithm is applied in our paper. Some practical examples were given based on improved NSGA-II algorithm by Geng [10] et al. Deng [11] did some research on some novel portfolio selection based on the improved entropy-weighted method.

2. Basis Definitions

2.1. Virtual Enterprise

From the perspective of operation, virtual enterprise refers to the enterprise organization that multiple enterprises combine in a discrete industrial chain to play their own expertise and operate together. It is necessary to deal with the decision-making question of virtual enterprise partner selection.

2.2. Basic Principle of Standard PSO

Particle swarm optimization (PSO) is a swarm intelligent bionic optimization algorithm based on group cooperation to simulate the foraging behaviour of birds. It was proposed by Kennedy and Eberhart in 1995. In a population of scale $N$, each individual flies in the $D$-dimensional space at a certain speed. In the process of searching the optimal solution, the speed $v$ and position $x$ of the individual are adjusted by referring to individual extremum $p_{best}$ and global extremum $g_{best}$ until the maximum iteration number $gen$ is reached or the global extremum $g_{best}$ is searched. The updating of position $x_i$ and velocity $v_i$ of individual $i$ are mainly completed by the following two formulas:

$$v_i = w v_i + c_1 \times \text{rand}(\cdot) \times (p_{best} - x_i) + c_2 \times \text{rand}(\cdot) \times (g_{best} - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

In the above formula, $w$ is inertia weight, the size of inertia weight affects the range of population in search, $c_1$ and $c_2$ are learning factors, which are non-negative constants; $\text{rand}(\cdot)$ is a random number between 0 and 1; $p_{best}$ is the individual extremum; and $g_{best}$ is the global extremum.

2.3. Basic Steps of Standard PSO

Step 1: Initialize the size of the particle swarm, the position and velocity of the particle.

Step 2: The fitness of particles is calculated according to the fitness function.

Step 3: The individual extremum and global extremum of particle swarm optimization are found according to fitness.

Step 4: Update the speed and position of particles.

Step 5: If the termination condition is reached, the global extremum and its corresponding particle position is output; otherwise, go to Step 2.

3. Improvement of PSO

We do some improvement work for PSO. The first is the improvement of inertia weight. According to the fitness of particles, the size of inertia weight is determined to accelerate the convergence speed and find the optimal value accurately, the formula:

$$w = \begin{cases} 
\frac{w_{\max} - (w_{\max} - w_{\min}) \times (f_{\text{avg}} - f)}{f_{\text{avg}} - f_{\min}} \times \frac{t}{\text{gen}}, & f < f_{\text{avg}} \\
(1) \text{with } \frac{f_{\text{max}} - f_{\min}}{f_{\text{avg}} - f_{\min}} \times \frac{t}{\text{gen}}, & f \geq f_{\text{avg}} 
\end{cases} \quad (3)$$
In the above formula, $w_{\text{max}}$ and $w_{\text{min}}$ are the pre-set extreme values of inertia weight. Previous studies have shown that the performance of PSO algorithm can be greatly improved when the inertia weight is between 0.4 and 0.9. $t$ is the current number of iterations, $\text{gen}$ is the total number of iterations, $f$ is the current fitness, $f_{\text{max}}$ is the maximum fitness, $f_{\text{min}}$ is the minimum fitness, and $f_{\text{avg}}$ is the average fitness. Then the similarity is introduced. The number of particles similar to other particles in the population is calculated as similarity, and then compared with the similarity threshold. When the threshold is exceeded, mutation operation is performed on the current particle, which can maintain the diversity of the population. The formula is as follows:

$$S_m = \frac{\sum_{j=1}^{n} p_{mj} \otimes p_{ij}}{N},$$

(4)

$$p_{mj} \otimes p_{ij} = \begin{cases} 1, & p_{mj} = p_{ij}, \\ 0, & \text{others}, \end{cases}$$

(5)

$$S_i = S_{\text{min}} + \frac{(n-S_{\text{min}}) \times t}{\text{gen}}.$$  
(6)

In the above formula, $S_m$ is the similarity of particle $m$, and $p_{mi}$ is the $i$-dimension of particle $m$. $N$ is the size of the population. $S_i$ and $S_{\text{min}}$ represent the similarity threshold and its minimum value respectively. At the same time, the pseudo code of PSO improvement part is as follows:

**Table 1.** Improved PSO algorithm

| Step | Description |
|------|-------------|
| Begin | For $i$ : $\text{gen}$  
| | For $j$ : $N$  
| | If $f(j) < p\text{best}(j)$ // If the individual fitness value is less than the individual extreme value,  
| | update individual extremum $p\text{best}(j)$  
| | End  
| | If $p\text{best}(j) < g\text{best}$ // If the individual extremum is less than the global extremum, update global extremum $g\text{best}$  
| | End  
| | Update inertia weight $w$ with formula (3)  
| | Update the particle velocity $v(j)$ and position $x(j)$ with formulas (1) and (2)  
| | The similarity is calculated by formulas (4)-(6)  
| | If $S_i > S_{\text{th}}$ // If the similarity of individuals exceeds the threshold, performing mutation operations on particles  
| | End  
| | Update fitness  
| | End  
| End |
4. Basis Definitions

4.1. Data Process
An enterprise wants to establish a virtual enterprise, and the initial information of the candidate enterprises is shown in the following table (see [9]):

| Candidate enterprise | Quality | Cost (million) | Risk   | Time/month | Evaluation results of last year |
|----------------------|---------|----------------|--------|------------|--------------------------------|
| 1                    | 4100    | 4.0            | general| 15         | no                             |
| 2                    | 5000    | 3.7            | good   | 11         | to be determined               |
| 3                    | 4750    | 4.1            | good   | 12         | yes                            |
| 4                    | 4550    | 3.9            | slightly good | 9      | to be determined               |
| 5                    | 4650    | 3.7            | good   | 8          | yes                            |
| 6                    | 4150    | 2.9            | general| 13         | no                             |

The discrete rules are set to normalize them. The discrete rules are as follows.

| Results after normalization | Quality | Cost (million) | Risk   | Time/month |
|-----------------------------|---------|----------------|--------|------------|
| 1                            | < 4000  | < 3.2          | general| < 10       |
| 2                            | 4000-4500 | 3.2-3.7   | slightly good | 10-12     |
| 3                            | > 4500  | > 3.7          | good   | > 12       |

The normalized information decision table is as follows:

| Candidate enterprise | Quality | Cost (million) | Risk | Time/month |
|----------------------|---------|----------------|------|------------|
| 1                    | 2       | 3              | 1    | 3          |
| 2                    | 3       | 2              | 3    | 2          |
| 3                    | 3       | 3              | 3    | 2          |
| 4                    | 3       | 3              | 2    | 1          |
| 5                    | 3       | 2              | 3    | 1          |
| 6                    | 1       | 1              | 1    | 3          |

4.2. Results of IPSO
According to the data of Table 3, Now, the improved particle swarm optimization algorithm is simulated. We set population size $N = 40$ and Number of iterations $gen = 500$. The inertia weight is set as follows $w_{max} = 0.9$, $w_{min} = 0.4$. Both learning factors $c_1 = c_2 = 2$. The particle's velocity interval is set to $[-1, 1]$, the particle position range is set to $[1, 3]$ and the minimum value of similarity threshold is set to 2. The fitness function is set as follows:

$$f = \frac{W(2)x(2) + W(3)x(3) + W(4)x(4)}{W(1)x(1)}$$

In the above formula, $W = (0.3152, 0.2753, 0.2185, 0.1909)^T$ is the weight of each index in the data. The simulation results are as follows:
Figure 1. The fitness evolution curve of improved PSO

As can be seen from Figure 1, the optimal individual has been found in the 10th generation of the simulation operation. The corresponding particle position is \((2.9877,1.0284,1.0088,1.0013)\) and the corresponding fitness value is 0.7377. The Euclidean distance between candidate enterprises and optimal individual:

Table 5. Euclidean distance between optimal individuals and candidate enterprise in IPSO

| Candidate enterprise | E1     | E2     | E3     | E4     | E5     | E6     |
|----------------------|--------|--------|--------|--------|--------|--------|
| Euclidean distance   | 1.2376 | 0.8971 | 1.1724 | 0.7635 | 0.7067 | 1.0178 |

It can be seen from Table 5 that the distance between enterprise E5 and the optimal individual is the smallest, that is, the closest to the optimal individual. Therefore, candidate enterprise E5 should be selected, followed by enterprise E4, enterprise E2, enterprise E6, enterprise E3 and enterprise E1.

4.3. Results of SPSO

The following is the result of simulation test with standard PSO:

Figure 2. The fitness evolution curve of standard PSO
Table 6. Euclidean distance between optimal individuals and candidate enterprise in SPSO

| Candidate enterprise | E1   | E2   | E3   | E4   | E5   | E6   |
|----------------------|------|------|------|------|------|------|
| Euclidean distance   | 1.2028 | 0.8591 | 1.1344 | 0.7683 | 0.7115 | 1.0130 |

4.4. Results of SPSO

We try to compare the results of Tables 5 and 6.

Table 7. Comparison of the priority of selecting candidate enterprises between standard PSO and improved PSO

| Priority | 1     | 2     | 3     | 4     | 5     | 6     |
|----------|-------|-------|-------|-------|-------|-------|
| Standard PSO | E5   | E4   | E2   | E6   | E3   | E1   |
|           | (0.7115) | (0.7683) | (0.8591) | (1.0130) | (1.1344) | (1.2028) |
| Improved PSO | E5 | E4 | E2 | E6 | E3 | E1 |
|           | (0.7076) | (0.7635) | (0.8971) | (1.0178) | (1.1724) | (1.2376) |

We can obtain Figure 3 from the data of Table 7.

**Figure 3.** The comparison of Euclidean distance between candidate enterprise and optimal individual in standard PSO and improved PSO

It can be seen from Figure 2 and Table 6 that the standard PSO finds the optimal individual around the 200th generation, and the selection order of candidate enterprises is E5, E4, E2, E6, E3, E1. It can be seen from Figure 3 and Table 6 that the results of standard PSO are basically consistent with those of improved PSO. Comparison the results of Figures 1-2, it can be seen that the search speed of improved PSO is faster than that of standard PSO, which indicates that adding dynamic inertia weight can greatly shorten the search time of the algorithm and accelerate the convergence.

5. Conclusion

This paper mainly introduces particle swarm optimization algorithm and its improvement. The improved idea is to add a dynamic inertia weight into the particle update formula, and change the inertia weight with the iteration times, so as to change the size of the population search range, prevent premature falling into local optimum, and quickly locate the global extreme value in the later stage. Finally, the improved particle swarm optimization algorithm is used for simulation operation. Finally, according to the relative distance between the candidate enterprise and the optimal individual obtained by the algorithm, the more suitable candidate enterprise is selected.
Acknowledgments

This research was supported by the “Natural Science Foundation of Guangdong Province, No. 2019A1515011038”, “Soft Science of Guangdong Province, and No. 2018A070712006, 2019A101002118”, “Guangdong Province Characteristic Innovation Project of Colleges and Universities, No. 2019GKTSCX023”. The authors are highly grateful to the referees and editor in-chief for their very helpful comments.

References

[1] S.S. Ben, Y.W. Ben, O. Ayadi, F. Masmoudi, An integrated fuzzy ANP-MOP approach for partner selection problem and order allocation optimization: the case of virtual enterprise configuration, RAIRO-Operations Research. 53(1): 223-241 (2019).
[2] H. Nyongesa, G.W. Musumba, N. Chileshe, Partner selection and performance evaluation framework for a construction-related virtual enterprise: a multi-agent systems approach, Architectural Engineering and Design Management. 13(5):344-364 (2017).
[3] X. Jia, Selection algorithm of virtual enterprise partner based on task-resource assignment graph, International Journal of Grid and Distributed Computing. 9(9):185-192 (2016).
[4] J.Y. Dong, S.P. Wan, Virtual enterprise partner selection integrating LINMAP and TOPSIS, Journal of the Operational Research Society. 67(10):1288-1308(2016).
[5] W. Zhang, Y.C. Zhu, Y. Zhao, Fuzzy cognitive map approach for trust-based partner selection in virtual enterprise, Journal of Computational and Theoretical Nanoscience, 13(1):349-360(2016).
[6] M. Huang, B. Sun, Research on modeling method and partner selection for collaborative production of virtual enterprises, Journal of Applied Sciences. 13(12):2189-2195(2013).
[7] Y. Zhang, F. Tao, Y.L. Li, B. Hou, L. Lv, L. Zhang, Green partner selection in virtual enterprise based on pareto genetic algorithms, The International Journal of Advanced Manufacturing Technology. 67(9):2109-2125 (2013).
[8] F.S. Hsieh, J.B. Lin, Virtual enterprises partner selection based on reverse auctions. Procedia Computer Science, International Journal of Advanced Manufacturing Technology. 62(5-8):847-859 (2012).
[9] R.Y. Jia, W.W. Pan, F. Liu, Partner selection of virtual enterprise based on rough set and genetic algorithm, Journal of Harbin Engineering University. 33(6):730-734(2012). (Chinese)
[10] Z.Q. Geng, S. Bi, Z. Wang, Y.X. Zhu, Y.M. Han, Operation optimization of ethylene cracking furnace based on improved NSGA-II algorithm, The 30th China Process Control Conference 71(3):1088-1094 (2020). (Chinese)
[11] X. Deng, C.J. Chen, A novel portfolio selection with prospect value constraint and distance measure of IFSs based on the improved entropy-weighted method, Journal of Intelligent & Fuzzy Systems. 39(3):3519-3543(2020).