International Journal of Hybrid Information Technology
Vol.8, No.3 (2015), pp.1-8
http://dx.doi.org/10.14257/ijhit.2015.8.3.01

Coalition Formation in Multi-agent Systems Based on Improved Particle Swarm Optimization Algorithm

Bo Xu[1], Zhaofeng Yang[2,*], Yu Ge[3] and Zhiping Peng[1]

[1]*Department of Computer Science and Technology, Guangdong University of Petrochemical Technology, Guangdong, China*
[2]*Software Engineering School of Pingdingshan University, Henan, China*
[3]*College of Fundamental, Sichuan Normal University, Chengdu, China*
**E-mail: pdsncyiyang@163.com*

Abstract

How to generate the task-oriented optimal agent coalition is a key issue of multi-agent system, which is a typical optimization problem. In this paper, an improved particle swarm optimization (IPSO) is proposed to solve this problem. In order to overcome the premature and local optimization problem in traditional particle swarm optimization (PSO), we proposed a variation of inertia weight PSO algorithm by analyzing the feasibility of particle optimization process in PSO. Compared with several well-known algorithms such as PSO, ACO, experimental results show that the global search capability of IPSO has been significantly improved and IPSO can effectively avoid premature convergence problem. Also it can solve the multi-agent coalition formation problem effectively and efficiently.

Keywords: *multi-agent systems, coalition formation, optimization problem, particle swarm optimization*

1. Introduction

As agent technology becomes more reliable and capable, multi-agent systems have been widely utilized to model real-world applications, such as multi-robotic systems [1], cloud computing [2]. The resources, capabilities and intelligence of individual agent are limited in multi-agent systems, so individual agent mainly rely on the coalition formation to complete complex tasks which is difficult. Such as perform of tasks like parallelism, spatial distribution, the strong fault tolerance, and the distribution functions [3]. Coalition formation has been a very active area of research and is an important basic issue in multi-agent systems; the main research is how to dynamically coordinate form agent coalition in order to complete a given task [4].

How to generate the task-oriented optimal agent coalition is a key issue of multi-agent system, the coalition formation problem was first introduced around 1993, and then become an important research direction [5]. Judging from the forthcoming and recent conferences, journal special issues, research reports, it is evident that there is growing interest in coalition formation problem.

At present, typical algorithms of coalition formation can be classified into two kinds: the first one is many researchers have studied on how to reduce the complexity of exhaustive method. We identify some of the limitations of deterministic search algorithms reported in literature. These include assumption of independence of coalition values and the exponential growth in computational requirements. Such as Sandholm and Lesser V R found the optimal coalition structure is a NP complete problem [6]. Shehory introduced a constant K to limit the number of agents in agent coalition [7]. DeVany proved that with the number of agent increasing, in order to quickly find the optimal coalition structure was very difficult [8]. Ye, D.Y. proposed a self-adaptation-based dynamic coalition

ISSN: 1738-9968 IJHIT

formation mechanism. The proposed mechanism operates in a neighborhood agent network. This mechanism enables agents to dynamically adjust their degrees of involvement in multiple coalitions and to join new coalitions at any time [9].

The second is based on intelligent algorithms. Such as Yang, J.G. and Luo, Z.H proposed a GA-based algorithm for coalition structure formation which aims at achieving goals of high performance, scalability, and fast convergence rate simultaneously [10]. Na Xia proposed used ant colony optimization algorithm for this problem, but did not receive good results [11]. Guo-fu Zhang used the particle swarm optimization algorithm (PSO) for solving complex coalition problems [12]. There is convergence fast, easy to implement and robustness advantages in the basic PSO algorithm, but is easy to fall into local optimum [13]. Many improved methods were proposed to improve the quality of solutions, for example Bo Xu proposed quantum evolutionary algorithm (QEA) and quantum-behaved particle swarm optimization for this problem [14, 15]. But the convergence of the algorithm is slow, while global optimization is not strong.

Recently, multi-agent systems have been employed to various domains in the real-world applications, such as foraging [16], box-pushing [17], aggregation and segregation [18], formation forming, cooperative mapping soccer tournaments, site preparation, sorting, and collective construction. All of these systems consist of multiple robots or embodied simulated robots acting autonomously based on their own individual decisions [19, 20]. However, some of these approaches are either cannot scale to large numbers or fragile to dynamic environment. To develop a new algorithm is a challenging task.

As many real-world problems are dynamic, they change over time. In such cases, requiring the optimization algorithm has to track a moving optimum as closely as possible, rather than just find a single good solution. The traditional algorithms that design underlying assumptions and mainly got rid of issues that must be addressed cannot meet the requirement of real-world any more.

In this paper, we do some further researches on agent coalition formation problem in dynamic environment and how to improve the speed of finding optimal solution, an improved particle swarm optimization (IPSO) is proposed to solve this problem. In order to overcome the premature and local optimization problem in traditional particle swarm optimization (PSO), proposed a variation of inertia weight improved PSO algorithm by analyzing the feasibility of particle optimization process in particle swarm optimization.

The rest of this paper is organized as follows: Section 2 describes the background of PSO and model for agent coalition are introduced; in Section 3, an improved particle swarm optimization (IPSO) is proposed; in Section 4, experimental analyses are given. At last, the conclusions and future work are given in Section 5.

2. Background

2.1 Model for Coalition Formation

This section identifies issues that must be addressed when the algorithm is applied to the multi-agent domain. In this paper, we will assume that [8, 10]:

(1) All agents are available.

(2) The agent can execute only one operation at a time.

Current multi-agent coalition formation algorithms assume that $A=\{A_1, A_2, ...A_n\}$ is for n Agent agents and that each has r-dimension capability vector $B_i = < b_i^1, b_i^2, ...b_i^r >$, $b_i^j \geq 0$, ($1 \leq i \leq n$, $1 \leq j \leq r$), where each capability is a property that quantifies the ability to perform an action. $T=\{t_1, t_2, ...t_m\}$ is for m tasks, and that a set of corresponding capability vector $B_{t_l} = < b_{t_l}^1, b_{t_l}^2, ...b_{t_l}^r >$. The agents communicate with each other and are aware of all tasks to be performed.

Agent coalition $CS = \{C_1, C_2, ...C_m\}$, a coalition C_i is a group of agents that decide to cooperate to perform a common task and each coalition performs a single task t_i. A coalition C_i has r-dimensional capability vector B_c representing the sum of the capabilities that the coalition members contribute to this specific coalition. A coalition C_i can perform a task t_i only if its capability requirement vector satisfies $B_{t_i}^j < B_{C_i}^j$, for each C_i, there exist coalition cost $Cost_{c_i}$ and coalition value $Value_{c_i}$. The multi-agent coalition is an optimization problem and can be depicted as follows:

$$\text{Object function Max } Value_{CS} = \sum_{i=1}^{m} Value(C_i) \quad (3)$$

$$\text{Restriction } B_{t_j}^k \le B_{C_i}^k \ (1 \le i, j \le m,\ 1 \le k \le r)$$

2.2 Particle Swarm Optimization

PSO (Particle Swarm Optimization, PSO) derived from complex adaptive systems (Complex Adaptive System, CAS). CAS theory was proposed in 1994, a CAS member is called body. Such as birds research systems, each bird in this system is called the body. Body has adaptability; it can communicate with the environment and other body, and change their structure and behavior in accordance with the process of exchange "learning" or "accumulate experience" [21].

PSO was first formally proposed in 1995 by Eberhart and Kennedy, and was originally derived from studies on the foraging behavior of birds [22]. PSO algorithm aims to find the global optimum value, the basic PSO algorithm speed formula is as follows [23]:

$$v_{id}(t+1) = V_{id}(t) + r_1 * c_1 * (P_{id}(t) - X_{id}(t)) + r_2 * c_2 * (P_{gd}(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = V_{id}(t) + X_{id}(t) \quad (2)$$

Where $V_{id}(t+1)$ denotes the speed value of i particle d dimension in the $t+1$ generation, r_1, r_2 is random number in [0,1], c_1, c_2 is the velocity coefficient and is a constant, P_{id} is the individual current best location, P_{gd} is the current global best position, X_{id} is position of i particle d dimension in t generation. PSO has fast convergence, simple operation, less the required parameters, but the algorithm global search capability is weak, easy to fall into local optimum [24].

3. Improved Particle Swarm Optimization (IPSO)

3.1. Improving Ideological

Based on the formula of velocity and position, we increased inertia weight w. The value of w is monotonically decreasing, when w is large will help broaden the scope of the search, when w is small can conducive to convergence. Calculated velocity and position based on improved formula of particle velocity and position, and then calculate the fitness of the particle.

However, only the inertia weight w is not enough, because in the actual search, although this approach converges faster, but it is easy to fall into local optimal medium. Therefore, we then do further improved, when the historical individual optimal value equal to the current optimum, or historical global optimal value is equal to the current best, believed that this particle may be trapped into local optimum, the position of the particle randomly selected. It can break the restrictions of local optimal value, effectively reducing premature convergence problem, greatly increasing the probability to obtain the global optimum.

Specific methods: due to the basic PSO algorithm is easy to fall into local optimum drawback, this article has been modified their algorithm formula. Speed formula changed as follows:

$$v_{id}(t+1) = w_{t+1} * V_{id}(t) + r_1 * c_1 * (P_{id}(t) - X_{id}(t)) + r_2 * c_2 * (P_{gd}(t) - X_{id}(t)) \quad (4)$$

$$X_{id}(t+1) = V_{id}(t) + X_{id}(t) \quad (5)$$

$$w_{t+1} = w_t - w_t * (\frac{t+1}{T_{\max}}) \quad (6)$$

Where w_{t+1} is the inertia weight, expressed the inertia weight in $t+1$. The formula (6) can guarantee the inertia weight w is a monotonic decreasing.

3.2. The Calculation of the Fitness of the Particle

We also improved the calculation of the fitness of the particle, if the fitness value of current particle is equal to the optimal value of history individual particles, or the fitness value of current particle is equal to the global adaptation best fitness, and then randomly selected particle position. If the current fitness of particles is better than the best historical individual particle or the best history, replace the current particle fitness historical individual or global optimum. $pop_j = rand(\Omega)$, if $fitness(pop_j) = fitnesspbest(j)$ or $fitness(pop_j) = fitnessgbest$

Where pop_j indicates the position of the first particle, Ω is the particle location selection, $fitness(pop_j)$ means that current fitness of the j-th particle, $fitnesspbest(j)$ means that the history optimal of the j particle, $fitnessgbest$ means that the optimal solution of the global history, $rand(\Omega)$ indicating that the particles position were randomly selected within the Ω range.

3.3. The Procedure of PSO

The Procedure of IPSO is as follows:

1. The initial setup particle size, lower and upper limits of inertia weight w, the upper and lower limits of particle position, acceleration factor, the number of maximum allowable iterations.

2. For each particle, evaluate the fitness of the particles according to the evaluation function.

3. According to the new w formula to calculate the value of w_t.

4. Calculate the new location of the particle according to the original formula, limiting the particle and position.

5. Re-evaluate the fitness of each particle according to the evaluation function.

6. For each particle, comparing the current value of the optimal value and history optimal value, if it is better to replace individual optimal and save individual best position, if the current particle is equal to the fitness of the best fitness history, then the particle positions were selected randomly.

7. Comparing the current fitness value of all particles and the history of its global best fitness values are better than the current history, if the current global best value, saving the value of the global best position, or if they are equal then selected the particle position randomly.

8. Meet the conditions, output search results, otherwise returns three search.

9. The g_{best} is the global optimum.

4. Experimental Results and Analysis

In this paper, we designed experiments of ACO, PSO[7] and IPSO. The parameters of algorithms are set as follows: agent has established the different ability $A_i = < b_i^1, b_i^2, b_i^3, b_i^4 >$, A_i complete the task $Cost_{A_i} = 1 \times b_i^1 + 2 \times b_i^2 + 3 \times b_i^3 + 4 \times b_i^4$; tasks require the ability $B_l = < b_l^1, b_l^2, b_l^3, b_l^4 >$, the interests of the task $\mathrm{Pr}\,ofit_l = 1 \times b_l^1 + 2 \times b_l^2 + 3 \times b_l^3 + 4 \times b_l^4$. In experiments set the number of Agent = 200.

Environment 1 $B_{all} = \sum_{i=1}^{n} B_{A_i} < B_l$, Three algorithms for the task can not be generated agent coalition.

Environment 2 $B_{all} = \sum_{i=1}^{n} B_{A_i} \geq B_l$, Three methods for the task can be generated agent coalition, but the quality of the results are different (see Table 1).

Environment 3 $B_{all} = \sum_{i=1}^{n} B_{A_i} >> B_l$, ACO and PSO algorithms have a great waste of resources, the coalition value is very small. And the advantages of our method are more obvious.

The statistical comparisons are showed in Table 1, Figure 1-2.

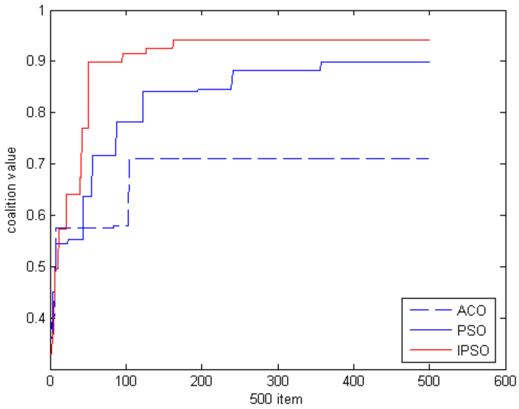

Figure 1. The Optimal Solution Evolving Curve (Environment 2, 500 Items)

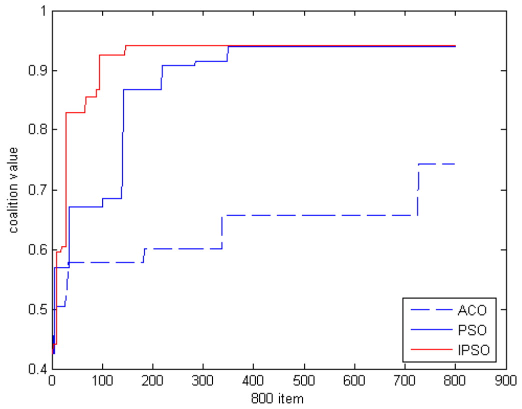

Figure 2. The Optimal Solution Evolving Curve (Environment 2,800 Items)

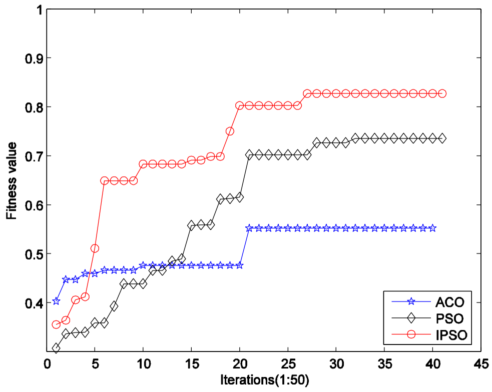

Figure 3. The Optimal Solution Evolving Curve (Environment 3, 500 Items)

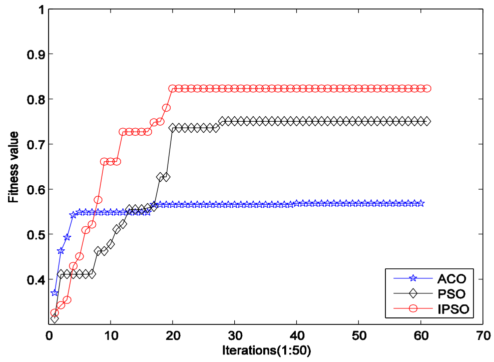

Figure 4. The Optimal Solution Evolving Curve (Environment 3, 800 Items)

Table 1. Comparison of Three Algorithms for Coalition Value (Statistical 200)

Environment	number of tasks	coalition value		
		IPSO	PSO	ACO
1	4	0.0	0.0	0.0
	8	0.0	0.0	0.0
	50	0.0	0.0	0.0
2	4	0.9414	0.9412	0.8332
	8	0.9349	0.9200	0.8011
	50	0.9300	0.9108	0.7832
3	4	0.8400	0.8412	0.8332
	8	0.7323	0.7200	0.7011
	50	0.5576	0.5108	0.5832

Table 2. Comparison of Three Algorithms for the Times (Statistical 100)

Environment	The number of tasks	Time (Seconds)		
		ACO	PSO	IPSO
1	4	-	-	-
	8	-	-	-
	50	-	-	-
2	4	7.45	6.11	5.01
	8	13.11	12.12	11.56
	50	42.09	34.11	31.01
3	4	7.76	4.13	3.11
	8	16.12	11.12	10.89
	50	44.90	38.32	34.99

Table 1 shows the process of the mean of best coalition value of population found by IPSO、PSO and ACO in there environments. It can be seen that IPSO with only improved strategy can get better results. Table 2 shows the times cost by IPSO、PSO and ACO when find the best coalition in there environments. It can be seen that IPSO with only improved strategy can get better results and its running time is a little faster than the PSO and ACO. From Figure 1-4, we can find that IPSO is hardly falls in the local

minimum, also its running time to reach the optimal solution is a little faster than the PSO and ACO. And it can avoid the algorithm getting in the local optima area easily.
From Table 1, 2, Figure 1-4, we can see that in the environment 1 task can not be solved by the three algorithms. In the environment 2 the three methods can generated agent coalition, but the quality of the results are different, The sum of coalition value of our algorithm is the largest, and our algorithm also has the fastest convergence, the highest utilization rate of resources. And results are the best. In the environment 3 ACO and PSO algorithms have a great waste of resources; the coalition value is very small. And the advantages of our method are more obvious.

4. Conclusion

Agent coalition is a key issue of multi-agent system, there are still many areas to be studied, especially for dynamic environments. We face not only the complexity of the agent coalition formation, but also the actual application process including not only the combination of resources, task allocation, but also co-verification, co-simulation and other follow-up steps. Of course this is the next step in this research. This is future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China ((No.61272382)

References

[1] L. Vig and J. A. Adams, “Multi-robot coalition formation”, IEEE Transactions on Robotics, vol. 22, no. 4, **(2006)**, pp. 637-649.
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz and A. Konwinski, "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, **(2010)**, pp. 50-58.
[3] T. C. Service and J. A. Adams, “Coalition formation for task allocation: theory and algorithms”, Autonomous Agents and Multi- Agent Systems, vol. 22, no. 2, **(2011)**, pp. 225-248.
[4] L. Vig and J. A. Adams, “Coalition formation: From software agents to robots”, Journal of Intelligent & Robotic Systems, vol. 50, no. 1, **(2007)**, pp. 85-118.
[5] T. W. Sandholm and V. R. Lesser, “Coalition among Computationally Bounded Agent”, Artificial Intelligence, vol. 94, no. 1, **(1997)**, pp. 99-137.
[6] O. Shehory and S. Kraus, “Methods for task allocation via agent coalition formation”, Artificial Intelligence, vol. 101, no. 1, **(1998)**, pp. 165-200.
[7] O. Shehory and K. S. Feasible, “Formation of Coalition among Autonomous Agents in Non-Super-Additive Environments”, Computational Intelligence, vol. 15, no. 3, **(1999)**, pp. 218- 251.
[8] A. De Vany, “Information, bounded rationality, and the complexity of economic organization”, Taiwan. Journal of Political Economy, vol. 2, no. 1, **(1996)**, pp. 21–44.
[9] D. Y. Ye, M. J. Zhang and D. Sutanto, “Self-Adaptation-Based Dynamic Coalition Formation in a Distributed Agent Network: A Mechanism and a Brief Survey”, IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 5, **(2013)**, pp. 1042 -1051.
[10] J. G. Yang and Z. H. Luo, “Coalition formation mechanism in multi-agent systems based on genetic algorithms”, Applied Soft Computing, vol. 7, no. 2, **(2007)**, pp. 561-568.
[11] J.-G. Jiang and N. Xia, “Searching for Agent Coalition for Single Task Using Improved Ant Colony Algorithm”, Journal of Computer Research and Development, vol. 42, no. 5, **(2005)**, pp. 734-739.
[12] G.-F. Zhang, J.-G. Jiang and N. Xia, “Solutions of Complicated Coalition Generation Based on Discrete Particle Swarm Optimization”, Acta Electronica Sinica, vol. 35, no. 2, **(2007)**, pp. 323-327.
[13] I. C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection”, Information Processing Letters, vol. 35, no. 6, **(2003)**, pp. 317-325.
[14] Z. Li, B. Xu, L. Yang, J. Chen and K. Li, “Quantum Evolutionary Algorithm for Multi-Robot Coalition Formation”, ACM/SIGEVO Summit on Genetic and evolutionary Computation, **(2009)**, pp. 295-302.
[15] B. Xu, Q. Guan and K. Chen, “Multi-Agent Coalition Formation Based on Quantum-behaved Particle Swarm Optimization”, Journal of Information and Computational Science, vol. 7, no. 5, **(2010),** pp. 1059-1064.
[16] J. Guerrero and G. Oliver, “Multi-robot coalition formation in real-time scenarios”, Robotics and Autonomous Systems, vol. 60, no. 10, **(2012)**, pp. 1295-1307.
[17] P. Chand and D. A. Carnegie, “Development of a reduced human user input task allocation method for multiple robots”, Robotics and Autonomous Systems, vol. 60, no. 10, **(2012)**, pp. 1231-1244.

[18] Y. Charif and N. Sabouret, "Dynamic service composition enabled by introspective agent coordination," Autonomous Agents and Multi-Agent Systems, vol. 26, no. 1, **(2013)**, pp. 54-85.
[19] T. Rahwan, T. Michalak and M. Wooldridge, "Anytime coalition structure generation in multi-agent systems with positive or negative externalities", Artificial Intelligence, vol. 186, no. 6, **(2012)**, pp. 95-122.
[20] P. Tabuada, G. Pappas and P. Lima, "Motion feasibility of multi-agent formations", IEEE Transactions on Robotics, vol. 21, no. 3, **(2005)**, pp. 387–392.
[21] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, **(1995)**, pp. 39-43.
[22] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", Proceedings of IEEE International Conference on Computational Cybernetics and Simulation, **(1997)**, pp. 4104-4108.
[23] S. Kiranyaz, T. Ince and M. Gabbouj, "Particle Swarm Optimization", Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition. Springer Berlin Heidelberg, **(2014)**, pp. 45-82.
[24] T. Blackwell, "A study of collapse in bare bones particle swarm optimization", Evolutionary Computation, IEEE Transactions on, vol. 16, no. 3, **(2012)**, pp. 354-372.