Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 7 (2007) 561–568
Applied Soft Computing
www.elsevier.com/locate/asoc

Coalition formation mechanism in multi-agent systems based on genetic algorithms☆

Jingan Yang [a,b,*], Zhenghu Luo [b]

[a] School of Computer and Information Engineering, Changzhou Institute of Technology, Changzhou 213002, Jiangsu Province, PR China
[b] Institute of Artificial Intelligence and Robotics, Hefei University of Technology, Hefei 230009, Anhui Province, PR China

Received 9 May 2005; received in revised form 26 January 2006; accepted 6 April 2006
Available online 31 July 2006

Abstract

As an important coordination and cooperation mechanism in multi-agent systems, coalition of agents exhibits some excellent characteristics and draws researchers' attention increasingly. Cooperation formation has been a very active area of research in multi-agent systems. An efficient algorithm is needed for this topic since the numbers of the possible coalitions are exponential in the number of agents. Genetic algorithm (GA) has been widely reckoned as a useful tool for obtaining high quality and optimal solutions for a broad range of combinatorial optimization problems due to its intelligent advantages of self-organization, self-adaptation and inherent parallelism. This paper proposes a GA-based algorithm for coalition structure formation which aims at achieving goals of high performance, scalability, and fast convergence rate simultaneously. A novel 2D binary chromosome encoding approach and corresponding crossover and mutation operators are presented in this paper. Two valid parental chromosomes are certain to produce a valid offspring under the operation of the crossover operator. This improves the efficiency and shortens the running time greatly. The proposed algorithm is evaluated through a robust comparison with heuristic search algorithms. We have confirmed that our new algorithm is robust, self-adaptive and very efficient by experiments. The results of the proposed algorithm are found to be satisfactory.

Keywords: Multiagent system; Agent coalition formation; Genetic algorithm; Chromosome encoding; Crossover and mutation

1. Introduction

In multiagent system (MAS), coordination and cooperation between the agents are one of the most important questions [1–3]. A single agent is unable to complete some complex tasks because its capability is individually limited or although can complete, but its performance and efficiency are far lower than the performance and the efficiency with cooperation and coordination of the many agents. Multi-agents with cooperation and coordination can adjust targets, dispel conflict, share resources so that MAS system is able to complete tasks with the best disposition, the higher efficiency, and obtain the biggest benefit [4,5]. Therefore, coalition formation is necessary when agents need to perform tasks which they cannot carry out efficiently alone.

Cooperation and coordination between the agents have a variety of forms [6–8]. Of them the agent coalition formation is one of many important methods [9,10]. In the initial state, all agents are mutually independent, and not cooperative. Hereafter, as the agents acquire unceasingly more knowledge from the system and the environment, every agent may form some coalition on the basis of certain principles by consulting and comparing. Each coalition is considered as an independent entirety. All members in the coalition cooperate fully, so the coalition will be allowed to draw support from the ability and the resources which other members have for completing tasks by a higher efficiency that a single agent cannot complete or even if he can complete these tasks, his efficiency is actually lower. Finally the entire system will have a largest efficiency.

In MAS, the possibility of each other cooperation for formatting coalition among the different kinds of agents almost all exists, and total numbers of the created possible coalitions are exponential relations with agent number in MAS system [2]. In order to make the system performance optimal, the possibility of the majority of coalitions' combinations must be considered. It is a complex *combinatorial optimization*

☆ This paper is supported in part by the Natural Science Foundation of Jiangsu Province under Grant No. 04KJD520010, PR China and International Centre for Theoretical Physics, Trieste, Italy.
* Corresponding author.
E-mail addresses: yangja@czu.cn, jayang@mail.hf.ah.cn (J. Yang).

1568-4946/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2006.04.004

problem. Optimization problems could be resumed as finding the maximal or minimal value for, respectively, a fitness and an error function:

$$\text{OPT}: \quad \text{Minimize/Maximize } f(x) \text{ with respect to } x \text{ in } S$$

where x is a vector of parameters and S the solutions' space.

The fitness function is always given, but is typically to complex to be not differentiable. Therefore, researchers have proposed many correlative algorithms [3,7]. This article is to solve this problem based on *genetic algorithm* by using these achievements for Refs. [11–14].

It is well known, the genetic algorithm uses ''superior poorly selects'', ''the survival of the fittest'' the natural selection and the nature genetic mechanism of Darwin for reference. Its essence is one highly effective and parallel global search method of the question-solving [15]. The genetic algorithm embarks from any initialization community, by using genetic operations, for example, the stochastic choice (enable a best individual in community to have more opportunities to hereditary his characteristics to next generation), crossover (emboding information exchanges between individuals within community in the nature), and the mutation (introduce a new variety in community for ensuring information multiplicity in community), made the community evolve into the more and more good areas of the search space one generation by one generation, solved the optimal solution. Therefore genetic algorithm has been successfully used for solving the complex combinatorial optimization problems which are very difficult to be solved by using some traditional methods [16–19].

This article organization is as follows. Section 2 describes formally the multi-agent system coalition question to be considered, proposes the known information and the constraint conditions of questions to be considered; Section 3 is an important part of the full paper, and proposes a novel genetic algorithm used for question-solving, discusses the chromosome coding forms, crossover and mutations operators, fitness function, initial population's production as well as the overall algorithm in detail; Section 4 presents the experimental results of this algorithm, and compares the experimental results in this paper with that of method in [2] so that we can evaluate objectively performance of our algorithm; Section 5 introduces the related work we are engaged in, and elaborates a detailed outline for the relations between this work and the article; Section 6 summarizes the method proposed in this paper, points out the merits and some drawbacks of our algorithm, and presents some of the most promising research directions in the future.

2. The question proposal

Generally speaking, the agents in MAS system may be divided into two types: the cooperative agents and the selfish agents. Among the cooperative agents, they can fully cooperate. In order to achieve the goal that the whole system is optimal, a single agent may sacrifice himself benefit; but among the selfish agents who choose their strategies as to maximize their own selfish utilities [20], they compete with each other and plunder the limited sources in the system for their maximal benefit, thus the efficiency of the entire MAS system would be very low. Therefore, the next stage of our research is to investigate ''solving optimization problems among selfish agents'' and ''how selfish agents learn to cooperate also''.

This article considers these agents belonging to the first type, namely all agents form some coalitions according to their respective resources, abilities, and likes, through repeatedly consulting and comparing in the basis of the principle for maximizing MAS system efficiency. All members in the various coalitions mutually cooperate, and share the resources that they have. They complete tasks that the system assigns to the coalition together and therefore obtain some better efficiency. In this foundation, entire MAS system efficiency should be the sum of various coalitions' efficiencies, because various coalitions all are formed by using optimized design, therefore overall performance of the MAS system also is optimal.

We may formalize and describe the above question as follows [2]: in the multi-agent system, suppose that we have n agents, are expressed as $A_1, \ldots, A_i, \ldots, A_n$, respectively. Each agent all has the certain ability, this kind of ability may be expressed as one ability vector, for example regarding A_i, its ability vector may be represented as $\mathbf{B}_i = \langle b_1^i, \ldots, b_r^i \rangle$ where the various components all are non-negative values, indicate that the ability of A_i which can complete some kind of specific behavior. The system has m tasks which need to be completed, expressed as $t_1, \ldots, t_j, \ldots, t_m$, respectively, each task of them all has the certain ability to request $\mathbf{R}_j = \langle r_1^j, \ldots, r_r^j \rangle$. A group of agents in the system may form one coalition C for completing some task through the consultation together, ability vector of the coalition C may be considered as ability vector sum of the coalition's all members, namely:

$$\mathbf{B}_C = \sum_{A_i \in C} \mathbf{B}_i \tag{1}$$

With regard to the coalition C, for completing task t_j, only when its ability $\mathbf{B}_C$ is on the premise of fulfilling ability requirements $\mathbf{R}_j$ of t_j, namely:

$$\forall\, 0 \le i \le r, \quad r_i^j \le b_i^C \tag{2}$$

This goal then can be achieved. In order to evaluate the efficiency of MAS system which is produced for the coalition C to complete the task t_j, we introduce function V_j which is a function of the coalition ability $\mathbf{B}_C$ and the task ability requirement $\mathbf{R}_j$. V_j is positive when the equation above holds, otherwise V_j is 0 (namely if coalition C cannot complete task, and also cannot bring efficiency to system).

With regard to some agent, this agent can join into some coalition, only the system total efficiency after this agent can join into the coalition larger than the system efficiency when this agent works independently, and then only one agent can join into one coalition. After once agent joins into some coalition, it only can work cooperatively with other agents in this coalition, but cannot join into other coalitions again. This requests that an agent cannot have one's wish when he choose which coalition that he want to join into, but must follow the principle which enable the system efficiency maximization, namely after joining into some coalition, own contribution to total efficiency of the system is largest. Because each coalition is one group of agents that is established dynamically for completing some concrete task, therefore suppose one coalition only can be engaged in one task.

In the MAS system containing n agents, the number of the possible coalition is 2^n. If given n agents, k tasks, then there are $k(2^n - 1)$ different possible coalitions. The number of configurations is $O(N^{(N/2)})$. So search space is extremely huge. It is very difficult to search entire space by using the conventional method. Therefore it is also difficult to solve the optimal solution. This article focuses on seeking a kind of effective search algorithm which enables total efficiency $\sum_i V_i$ of the MAS system to maximize or to have approximate maximization, and to satisfy the above constraints and the time scope which may be accepted.

3. Algorithm designing and realization

Genetic algorithms (GAs) are heuristic search schemes based on a model of Darwinian evolution. Although not guaranteed to find optimal solution, genetic algorithms have been shown to be effective at finding approximate optimal solution, in some cases, optimal solutions to combinatorially explosive problems. Genetic algorithms are inspired from biological processes (i.e. cells' division, DNA, crossover, and mutation, etc.). The underlining idea is to generate successive sets of solutions (generations), making each new generation inheriting properties from the best solutions of the precedent. In order to perform such a step, we have to select the best solutions and mix them together (crossover). A GA typically looks like that:

(1) Generate a first generation with random parameters.
(2) Evaluate all individuals of the generation.
(3) Crossover the best individuals together to get the new generation (children).
(4) Make random mutation across the new generation.
(5) Go back to (2).

Key techniques to solve the problems by using genetic algorithm is to seek the coding expression of gene space (solution space) suitable to the question itself, crossover and the mutations operator correspond to chromosome encoding method as well as fitness function form with the chromosome encoding method. Therefore this can introduce the problem-related domain knowledge and the constraint conditions into the genetic algorithm, so that we enable the algorithm to be more stable, more effective, and may speed up the convergence rate of the algorithm and improve the quality of the solutions.

3.1. Two-dimensional binary chromosome encoding

First step of using genetic algorithm is to seek one reasonable chromosome coding way, and the most direct encoding way that is suitable to the problems discussed in this article is the *unidimensional* natural number coding. In the coding method, the chromosome length is n, the chromosome gene order corresponds, respectively, to the different agents, but the gene values, namely *isoelectronic* genes correspond to the task number which agent is engaged in. Agents engaged in the same tasks constitute a coalition. Fig. 1 showed this chromosome expression method.However we have discovered that it is actually very difficult to discover the corresponding crossover and mutations operators if we use this chromosome expression method. Therefore, we have designed a novel two-dimensional binary chromosome coding method, as shown in Fig. 2. Each chromosome is a two-value matrix of $m \times n$, each row of this matrix all corresponds to one agent, but each column of the matrix corresponds to one task. $a_{ij} = 1$ expresses agent j is engaged in task i, namely joins into coalition C_i that is engaged in task i; $a_{ij} = 0$ expresses agent j is not engaged in task i, namely this agent is removed from coalition C_i that is engaged in task i. Because one agent only can join into one coalition, but one coalition only can be engaged in one task, therefore each row of the matrix only can have one element, namely 1, other elements are 0.

Gene place :	0	1	2	...	n-1
Allele :	2	5	0	...	5

Fig. 1. The encoding type of one-dimensional natural number chromosome.

The encoding method of two-dimensional binary chromosome has the following merits: this encoding method not only retained the binary coding, but also had many excellent characteristics since the method can satisfy the smallest character collection encoding rules [8]. Also it is fundamentally fit for two-dimensional essence for solving the multi-agent coalition questions. Therefore, this established the good foundation for question-solving, and provided the extremely broad space for designing the crossover and mutations operators which have the excellent performances.

3.2. Two-dimensional "or" crossover operator

In view of this two-dimensional binary chromosome encoding method, this paper proposed a two-dimensional "or" crossover operator corresponding to this approach (namely: two-dimensional "or" crossover-TDOC). Biggest differences between TDOC and the ordinary *unidimensional* crossover operator: The TDOC operator not only can carry on

Task \ Agent	0	1	⋯	i	⋯	n-1
0	0	0	⋯	0	⋯	1
1	1	0	⋯	1	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
m-1	0	1	⋯	0	⋯	0

Fig. 2. Encoding method of two-dimensional binary system chromosomes.

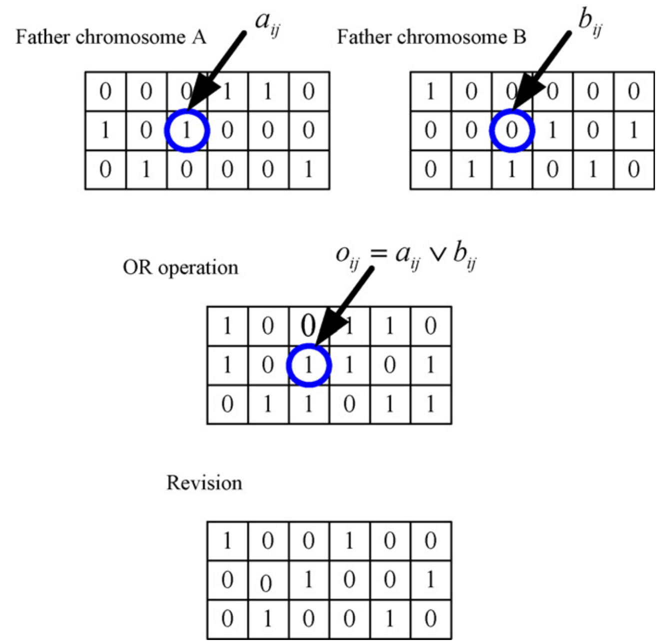


Fig. 3. Two-dimensional ‘‘or’’ crossover operator.

crossover operation to the two-dimensional structure chromosome, moreover each time crossover of two parents individuals only produce one descendant individual, certainly is not the traditional two descendants. Working mechanism of two-dimensional ‘‘or’’ crossover operator is as shown in Fig. 3. Suppose the chromosomes of two parent individuals participating in crossover are A and B separately, the elements of its two-dimensional matrix are a_{ij} and b_{ij} separately. Suppose the filial generation individual chromosome produced after crossover is O, its matrix element is o_{ij}. For $\forall 0 \leq i < m$, $0 \leq j < n$, let $o_{ij} = a_{ij} \vee b_{ij}$, namely carries on ‘‘or’’ operation bit by bit. Thus all two parent chromosomes information is fused into the filial generation, but this time filial generation chromosome possibly did not satisfy constraint that each row only has one element 1. In fact, if two parents' agents are engaged in same tasks, then row j in the O only has one element which is 1. But if two parents' agents are engaged in the different tasks, then row j in the O will have two elements which are 1. For this reason, we should revise O according to the following steps:

(1) Stochastically choose one row j from O which is not processed. If this row only has one element which is 1, then do not make any adjustment, directly transfer to step 3.
(2) Suppose two 1 elements in row j are, respectively, o_{ij} and o_{hj} ($i < h$), do the following judgement to o_{ij}. If we have $\sum_{o_{ik}=1,k\neq j}\mathbf{B}_k \geq \mathbf{R}_i$, then judge $\sum_{o_{hk}=1,k\neq j}\mathbf{B}_k \geq \mathbf{R}_h$ hold or not hold. If it hold, then stochastically set o_{ij} to 0 or 1, and will set o_{hj} to $(1 - o_{ij})$. Otherwise, keep o_{hj} invariable, sets o_{ij} to 0.
(3) Repeat step 1, until all rows are processed.

After two parental chromosomes pass through ‘‘or’’ crossover, two messages are fused into one, henceforth in descendant individual ‘‘O’’, coalition C_i that was engaged in task i contains all agents in two parental generation individuals which all are engaged in task i. Because two parental generation individuals are the legitimate individuals, it is obvious that coalition C_i certainly may complete the task i, namely C_i is full, but simultaneously C_i also is redundant. In order to eliminate this redundant, judgement condition $\sum_{o_{ik}=1,k\neq j}\mathbf{B}_k \geq \mathbf{R}_i$ in step 2 judges when agent j does not participate (to be the members of both C_i and C_k at the same time) under situation, C_i still could complete this task. If the condition holds, this indicates that the agent is elective in C_i. Then we must judge affect of this agent in C_k at the moment. If this agent is also elective in C_k, then we may stochastically decide whether this agent does join into coalition C_i, otherwise this agent only can join into coalition C_k and withdraw from coalition C_i. If $\sum_{o_{ij}=1,k\neq 1}\mathbf{B}_k \geq \mathbf{R}_i$ does not hold, this indicates the agent j plays an important and not-lacking role in coalition C_i, then this agent must join coalition C_i. This judgement process indicates that an agent joins or does not join some coalition by no means of the non-goal stochastic behavior, but under the instruction of parental generation information, and on the premise of completing the corresponding task, thus all tasks can normally be completed, therefore O also should be a legitimate individual. In brief, two legitimate parental generation individuals affected by two-dimensional ‘‘or’’ crossover operator produces one legitimate descendant individual.

3.3. Mutation operator

Consider the structure of the two-dimensional chromosome as well as the constraint condition that is mentioned above, the mutation operator designed in this paper mutates according to the following steps:

(1) Take any two columns from chromosome, represented by using i,j separately.
(2) Suppose ‘‘1’’ elements in the i and j columns are separately a_{mi} and a_{nj}, if $m = n$, transfer to (1).
(3) If $\sum_{a_{mp}=1,p\neq i}\mathbf{B}_p + \mathbf{B}_j \geq \mathbf{R}_m$ also $\sum_{a_{nq}=1,q\neq j}\mathbf{B}_q + \mathbf{B}_i \geq \mathbf{R}_n$, then let $a_{mi} = a_{nj} = 0$, $a_{mj} = a_{ni} = 1$, i.e. exchange i and j columns, the mutation process is completed; or else transfer to (1).

Core thought of the mutations operator is: two coalitions exchanged one respective member on the premise of satisfying the constraint conditions. Fig. 4 shows this process.

3.4. Fitness function and initial population's generation

The fitness function is used to evaluate quality (optimal or inferior) of the solutions. The population is precisely carried on the evolution under this function instruction, obviously choosing one appropriate fitness function is very important. Obviously, on the problems which this article makes research on, the total efficiency of MAS system is considered as a very good evaluation criterion, thus this article directly selects the total efficiency of the system as a fitness function. The

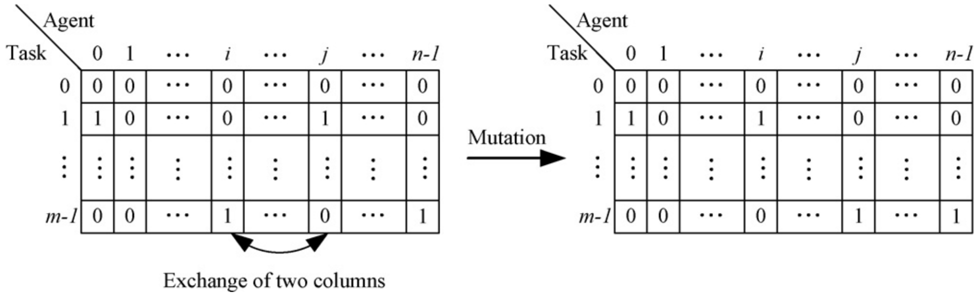


Fig. 4. Mutation operator.

chromosome corresponding to MAS system that cannot complete all tasks is the illegal individual, and its fitness is defined as 0; but the fitness of the legitimate individual that can normally complete all tasks should be the sum of total efficiency of a variety of coalitions. Thus survival probability of the illegal individuals corresponding to MAS system that cannot complete all tasks is 0. These illegal individuals must been eliminated along with population's evolution, but higher the efficiency of MAS system which corresponds to the efficient individuals is, the fitness values of these individuals are also higher. The population's survival probability of them are also bigger, this conforms to ''the survival of the fittest'' of Darwin.

(1) If the individual is legitimate, then needs not to be revised, directly returns.
(2) With regard to the illegal individual, stochastically choose a task i which cannot be completed and find the corresponding coalition C_i.
(3) Stochastically choose one task j which may be completed and find corresponding coalition C_j.
(4) Elect one member m from coalition C_j.
(5) If satisfying the condition $\sum_{k\in C_j,k\neq m}\mathbf{B}_k \geq \mathbf{R}_j$, then transfer m from C_j to C_i, otherwise return to (4), until all members of C_j are even judged.
(6) If the task i still could not be completed, return to (3), otherwise return to (2), until the all tasks are completed.

The core thought of the above steps is: extracting the members from the redundancy coalition is used for completing these tasks which cannot be completed, thus can achieve the task assignment balance, and transform the illegal individuals into the legitimate individuals.

3.5. Flowchart for implementing algorithm

According to the sections above, we give the genetic algorithm realization flowchart used for deducing the multi-agent system coalition mechanism as shown in Fig. 5.

4. Experimental results

In order to evaluate the performance of our algorithm, we complete three groups of the experiments with the different numbers of agents. Each group of them operates, respectively, 10 times. Finally we indicate the results of the average values, and objectively compare them with the results produced by using method in [2] simultaneously. The parameter settings of three groups of tests are shown in Table 1. We discover from our experimental results when the crossover probability p_c is 0.95,

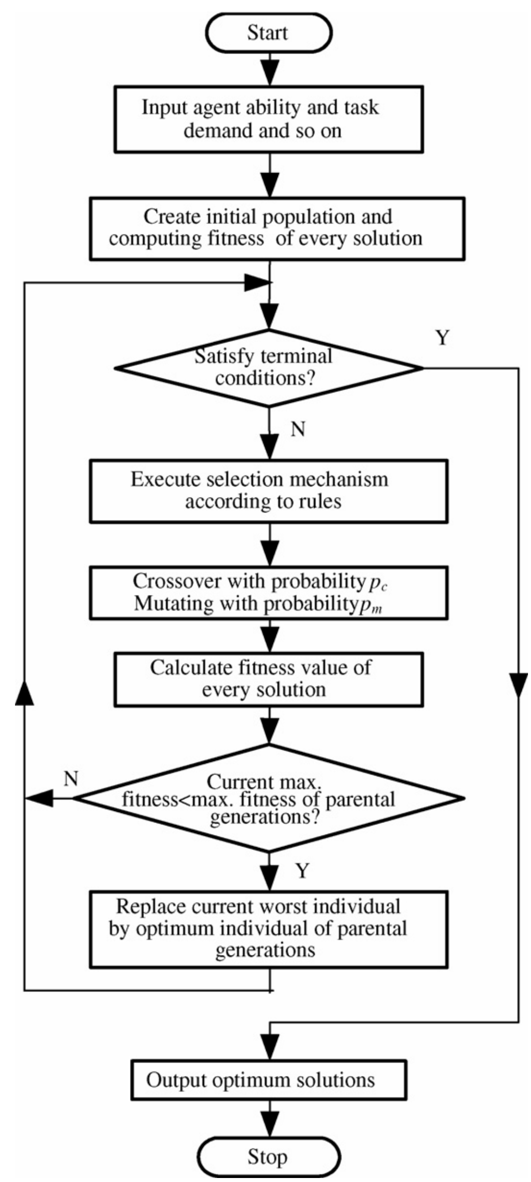


Fig. 5. Global flowchart for implementing our algorithm based on genetic algorithm.

Table 1
Parameter settings

No. of groups	Number n of agents	Number m of tasks	Population sizes	Number of running generations designated
1	10	4	20	300
2	20	4	30	2000
3	20	5	30	3000

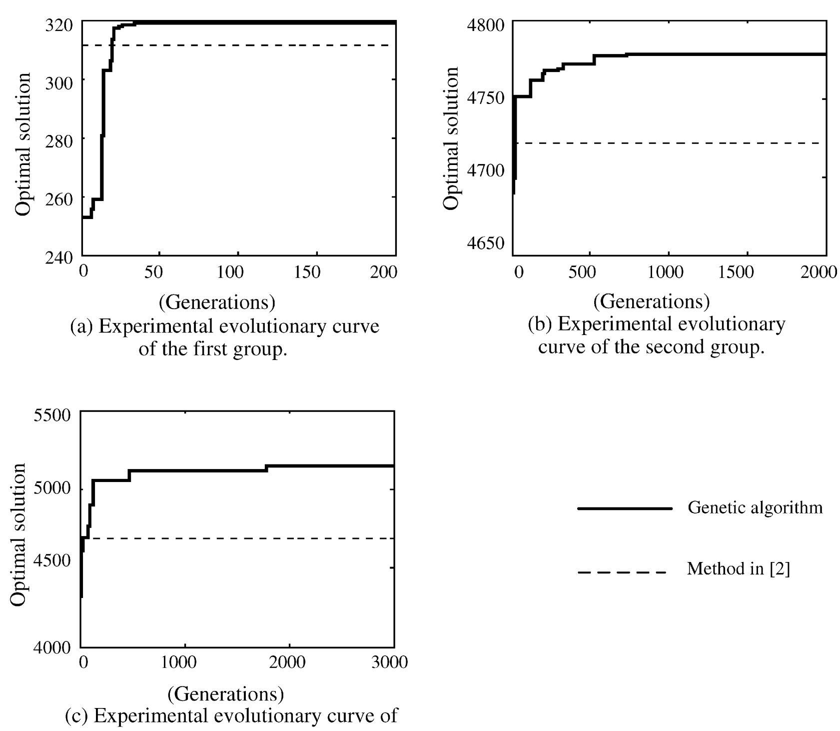
Table 2
Experimental results of the three groups

No. of groups	Best solutions (10 runs)	Average best solutions	Average worst solutions	Average values	Average running time (s)	Comparison to results in [2]	
						Running results	Running time (s)
1	318.74	315.38	299.63	309.18	0.06	311.23	0.04
2	4785.43	4779.33	4677.77	4733.67	1.57	4720.95	1.66
3	5334.50	5322.89	5200.97	5265.03	2.65	4502.31	2.20

mutation probability p_m is 0.02, the convergence rate is fast, and the solution quality is optimal.

Table 2 shows the experimental results of three groups. Because the method in [2] is one kind of definite method, thus each group of experiments only operates once. Fig. 6 gives the most representative optimal solution evolution curve of each experimental group, and presents the evolutionary process of the solutions of our algorithm.

From Table 2 and Fig. 6, in contrast with other method, for example, method proposed in [2], we may discover that the convergence rate of the algorithm in this paper is fast, the robustness is strong. In the experiments with different scales, our algorithm is very self-adaptive, its running rate was slightly lower than method in [2], but the solution quality is obviously superior to the results in [2]. We should point out more specifically: in a smaller scale, 10 experiments of first group, the solutions are converged to the same optimal solution (318.74) for 9 runs, although we could not determine whether 318.74 were theoretically the optimal solution, but we have demonstrated that the solution quality of our algorithm is very high as compared with the experimental result 311.23 in [2], and from the experimental scale and results of this group, this solution most approaches to a theoretical optimal solution or this solution itself *may be just* a theoretical optimal solution. This has also been demonstrated by the tested results of another two groups. Therefore the experimental results in this paper have proved that our algorithm not only retains the simple, of common use, and strong controllable performances of the



(a) Experimental evolutionary curve of the first group.

(b) Experimental evolutionary curve of the second group.

(c) Experimental evolutionary curve of the third group.

Fig. 6. Optimal solutions of the evolutionary curves.

genetic algorithm and so on, but also greatly promotes the solution quality and the convergence rate by introducing question-related domain knowledge and some constraint conditions so that this algorithm is used successfully for deducing the mechanism of the multi-agent system coalition construction, and for a multi-agent system in the robot soccer domain to investigate both simulated and embodied intelligence. This task is significant important for applications-oriented research, and our research has focused on investigating other cooperation and coordination behaviors of multi-agent system coalition towards *real world applications*, for example, *multirobot systems and evolutionary multirobot group*.

5. Related work

In [2], the coalition planning is disposed by using a certain inference method. In each turn of circulation, an independent agent that does not join any coalition makes an attempt to combine all kinds of the coalitions containing own (because search space is huge, this method stipulates coalition largest member k), and calculates the efficiency of these coalitions which carry out a different task so that the coalition combination with the highest average efficiency and the corresponding task can be discovered from all possible situations, quotes the average efficiency to the system, then system evaluates all quoted prices. The agent that quoted price is highest is authorized to form an coalition and carry out the corresponding task. Finally, system will delete all members which this coalition involves and the task engaged in from tabulation and, continues the next turn of circulation, until all tasks assignment is finished. Merits of this method are: (1) this algorithm is simple; (2) the algorithm is easy to be realized and its performance is stable; (3) the solution of the algorithm can approach to the optimal solution by the logarithm way [20–22], but this is only suitable for the small scale MAS system, and is unable to be expanded to larger scale system containing many agents, moreover the largest member of the coalition is not easy to be determined.

The coalition method proposed in [3] is: all agents mutually propose the cooperation efficiency *quoted price* according to oneself grasping the knowledge about the MAS system and other agents in the system. If two agents are mutually interested in the *quoted prices* proposed by opposite side, then the coalition is composed of these two agents. Two members again enter a bid, respectively, to an objective appraisal of the cooperation benefit and the benefit assignment scheme, and according to entering a bid above from the two agents selects one as this coalition superintendent. After this coalition formed, it is considered as one whole, its function may be equivalent to one independent agent, in the next turn of circulation, the new coalition quoted price is carried out again by the coalition superintendent with other coalition or independent agent, if succeeds, a bigger scale coalition may consist of this coalition and the opposite sides which mutually are interested in, and select the new coalition superintendent. So circulates, until the system creates no longer the new coalition.

This method is simple, effective, and robust. After an agent joins some coalition, it could not be separated from the coalition again by receiving other factor seducement outside, thus the system running is stable. But each agent needs the related knowledge about the system and other agents, this aggravates the system workload. Moreover, in the process to enter a bid, the agent with higher bid always is elected as a superintendent of this coalition. This possibly causes an individual agent to promote a bid price intentionally in order to work as a coalition superintendent.

In [14], genetic algorithm was introduced for research on formation mechanism question of virtual clustering for a holon. A task assignment strategy is proposed based on Contract Net Protocol [15], optimized by using genetic algorithm, and obtained the very good effect. But main differences between his method and our method: this method assigns one task to one holon, but is not one group of holons, among all holons the effective cooperation also is not carried out.

6. Conclusions and future work

Research on the cooperation of multiple agents has to address three main problems: (1) how to appropriately divide the functionality of the system into multiple agents; (2) how to manage the dynamic configuration of the system in order to realize cooperative behaviors; and (3) how to achieve coordination and learning for a group of multiple agents.

This paper proposed and evaluated a coalition formation mechanism that takes trust relationships between the agents into consideration. The mechanism extended the existing concept of temporary customer coalitions to long-term coalitions of both customers and vendors. The genetic algorithm, due to the characteristics of its intrinsic parallelism, self-organization, as well as self-adaptive, and self-learning and so on [21–23], has been applied successfully to solve the complex combinatorial optimal questions which are difficult to be solved by the traditional methods. The experiments have proven that genetic algorithm can solve the optimal solution or the approximate optimal solution of many combinatorial optimal questions, and already became one of the effective tools to study this kind of questions. This article introduces genetic algorithm into research on the multi-agent system coalition mechanism, and proposes a novel two-dimensional binary chromosome encoding method as well as two-dimensional or crossover operator and mutations operator adaptive to this algorithm. Individual of the legitimately parental generations can create the effective descendant individual under act of two-dimensional or the crossover operator function. This greatly promotes efficiency, reduces the computational cost and computational complexity. In contrast with the experimental results of the other methods, our algorithm has the strong robustness, high self-adaptive and fast convergence rate. This is particularly encouraging because we can now attempt to solve the optimal coalition structure recognition problem for realistic problem size. So the results of the proposed algorithm are found to be satisfactory. But the method proposed in this paper is also required to be improved as follows:

(1) How to promote the operating speed of this algorithm, and make this algorithm superior to the other methods in both solution quality and the computational cost and computational complexity?
(2) Whether or not can agent alleviate the question constraint conditions, for example, whether or not can one agent simultaneously join several coalitions, the MAS systems do whether have to complete all tasks?
(3) How does the method proposed in this paper continue working stably and effectively under this frame and can have the optimal coalition disposition?

Therefore, we will further introduce the domain-related knowledge, and combine genetic algorithm with the heuristic methods for solving the combinatorial optimization problems. These are some of our most promising research directions in the future.

References

[1] B.P. Gerkey, M.J. Mataric, Sold! Auction methods for multi-robot coordination, IEEE Trans. Robot. Automat. 18 (5) (2002) 758–768.
[2] O. Shehory, S. Kraus, Task allocation via coalition formation among autonomous agents, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, August, Morgan Kaufman, 1995, pp. 655–661.
[3] S. Ketchpel, Forming coalitions in the face of uncertain rewards, in: Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, July, (1994), pp. 414–419.
[4] Shehory, S.K. Sycara, S. Jha, Multi-agent coordination through coalition formation, Lect. Notes Artif. Intell. 1365 (1997) 143–154.
[5] T.W. Sandholm, K.S. Larson, M.R. Andersson, et al., Anytime coalition structure generation with worst case guarantees, in: Proceedings of the National Conference on Artificial Intelligence (AAAI), Madison, WI, July, (1998), pp. 46–53.
[6] K. Tuyls, T. Maes, et al., Towards a relation between learning agents and evolutionary dynamics, in: Proceedings of BNAIC 2002, KU Leuven, Belgium, 2002.
[7] T. Scully, M.G. Madden, G. Lyons, Coalition calculation in a dynamic agent environment, in: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.
[8] Y. Murata, N. Shibata, M. Ito, Agent oriented self-adaptive genetic algorithm, IPSJ Trans. Math. Model. Appl. 44 (SIG07) (2003).
[9] S. Nolfi, D. Floreano, Evolutionary: The Biology, Intelligence, and Technology of Self-organizing Machines, MIT Press/Bradford Books, Cambridge, MA, 2000.
[10] Q. Zhong, T. Xie, H.W. Chen, Duty assignment and dispatch based on genetic algorithm, J. Comput. Res. Dev. 37 (10) (2000) 1197–1203.
[11] J.A. Yang, H.Y. Wang, Research on behavior switch method for evolutionary robotics based on neural network, J. Shanghai Jiaotong Univ. 36 (2002) 89–95.
[12] G. Chalkiadakis, C. Boutilier, Coordination in multiagent reinforcement learning: a Bayesian approach, in: Proceedings of the Second International Joint Conference on Autonomous Agent and Multiagent Systems (AAMAS-03), 2003, pp. 709–716.
[13] G. Chen, Z. Yang, H. He, K.M. Goh, Coordinating multiple agents via reinforcement learning, Auton. Agents Multi-agent Syst. 10 (3) (2005) 273–328.
[14] Y. Xu, R.W. Brennan, X. Zhang, et al., A genetic algorithm-based approach to Holon virtual clustering, in: Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000), vol. III, Orlando, FL, USA, July 23–26, (2000), pp. 380–385.
[15] T.W. Sandholm, An implementation of the contract net protocol based on marginal cost calculations, in: Proceedings of the National Conference on Artificial Intelligence, Washington, DC, July, (1993), pp. 256–262.
[16] L.-K. Soh, C. Tsatsoulis, Real-time satisfying multiagent coalition formation, in: Proceedings of the International AAAI Workshop Coalition Formation in Dynamic Multi-agent Environments, AAAI Press, Menlo Park, CA, 2002.
[17] M. Klusch, A. Gerber, Dynamic coalition formation among rational agents, IEEE Intell. Syst. 17 (3) (2002) 42–47.
[18] T. Sandholm, K. Larson, M. Andersson, O. Shehory, F. Tohme, Coalition structure generation with worst case guarantees, Artif. Intell. J. 111 (1/2) (1999) 209–238.
[19] P. Merz, B Freisleben, Genetic local search for the TSP: new results, in: Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, IEEE Press, (1997), pp. 159–164.
[20] N. Nisan, Algorithms for selfish agents, in: Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March, (1999), pp. 1–15.
[21] H. Liu, M. Tang, Evolutionary design in a multi-agent design environment, Appl. Soft Comput. 6 (2006) 207–220.
[22] E. Fabricio, B. Jean-Paul, ELA—a new approach for learning agents, Auton. Agents Multi-agent Syst. 10 (3) (2005) 215–248.
[23] S.-R. Yang, S.-B. Cho, Co-evolutionary learning with strategic coalition for multiagents, Appl. Comput. 5 (2) (2005) 193–203.