Multi-task Coalition Parallel Formation Strategy Based on Reinforcement Learning

JIANG Jian-Guo[1, 2] SU Zhao-Pin[1, 2] QI Mei-Bin[1, 2] ZHANG Guo-Fu[1, 2]

Abstract Agent coalition is an important manner of agents′ coordination and cooperation. Forming a coalition, agents can enhance their ability to solve problems and obtain more utilities. In this paper, a novel multi-task coalition parallel formation strategy is presented, and the conclusion that the process of multi-task coalition formation is a Markov decision process is testified theoretically. Moreover, reinforcement learning is used to solve agents′ behavior strategy, and the process of multi-task coalition parallel formation is described. In multi-task oriented domains, the strategy can effectively and parallel form multi-task coalitions.

Key words Multi-task coalition, parallel formation, Markov decision process, reinforcement learning

In several applications, an agent may not efficiently perform a task by itself, and thus, agents have to form a coalition in order to execute the task. Since "coalition" first was put forward in 1993, coalition formation has been a key problem and widely studied in the research of multi-agent system (MAS)[1−4].

Several works have investigated the coalition formation problem based on the n-person cooperation game theory, which mainly deals with the coalition′s utility distribution among agents according to their preference to reach a global optimal coalition[1, 3, 5−6]. But these researches relax the coalition formation algorithms, emphasize the equity of utility allocation, and ignore the difference of agents′ action during coalition formation, leading to coalition′s instability. This is because although the total utilities of the coalition increases, the utilities of its members may greatly decrease.

Luo and Chun[7] proposed a nonreducing utility allocation strategy that can encourage agents to enlarge a coalition to obtain more total utilities and personal utility. The strategy is simple and timely. But it does not distinguish the difference of agents′ contributions to a coalition by allocating utility averagely. The designed "contract rule" diminishes other agents′ interest to join the existing coalition, and affects the formation of a global optimal coalition.

Jiang et. al.[8] proposed a coalition formation strategy based on capability vector contribution-rate and auction that can partly realize the thinking of distribution according to work. But it does not give concretely the amount of capabilities that agents contribute to a coalition in course of problem-solving, and the strategy is easy to run into coalition lock. Furthermore, it allocates additional utility on auction without distinguishing the contribution of new agents that have brought certain added utility to the existing coalition, and the auction needs agents to interact with each other ceaselessly, producing a large amount of communication expense.

Shehory et. al.[9] developed a protocol that enables agents to negotiate and form coalitions, and provides them with simple heuristics for choosing coalition partners. But when two or more qualified coalitions submitted their proposals simultaneously, the proposed method cannot reach the global optimal coalition. Especially, all agents originate proposals simultaneously, and the system will run into coalition lock. Moreover, the negotiation strategy that an agent has only one turn in each round decreases the probability of forming an optimal coalition.

All these previous works deal with the payoff distribution and negotiation strategy, and do not take into account agents′ behavior strategy. In [10], the strategy can formalize the subtask selection using a Markov decision process, but in their environment, agents cannot control perfectly the consumption of their resource, and the auction approach only can form a coalition for a task serially and cannot be used in parallel multi-task oriented domains.

In this paper, we will propose a novel multi-task coalition parallel formation strategy and design agent behavior strategy based on reinforcement learning. Our example illustrates that the proposed strategy can effectively and parallel form multi-task coalitions in multi-task oriented domains.

1 Problem formulation

The problem of multi-task coalition parallel formation can be described as follows:

1) Task: Given a finite set of tasks $T = \{T^1, T^2, \cdots, T^N\}$ and that each task T^i $(i = 1, 2, \cdots, N)$ has a certain r-dimensional capability required vector $\boldsymbol{B}_{T^i} = (b^1_{T^i}, b^2_{T^i}, \cdots, b^r_{T^i})$ $(b^j_{T^i} \geq 0,\ j = 1, 2, \cdots, r)$. For executing task T^i, agents can obtain the utility $P(T^i)$.

2) Agent: We consider situations where a set of m rational bounded-resource agents, $Agent = \{1, 2, \cdots, m\}$, have to cooperate to execute the tasks $T = \{T^1, T^2, \cdots, T^N\}$. Each agent $k \in Agent$ has a r-dimensional capability vector $\boldsymbol{B}_k = (b^1_k, b^2_k, \cdots, b^r_k)$ $(b^1_k \geq 0,\ j = 1, 2, \cdots, r)$, where each capability is a property that quantifies the ability to perform an action. Agent k can be selected to perform the task T^i only if $\exists\, j \in \{1, 2, \cdots, r\}$, $b^j_k \geq b^j_{T^i}$. It also has a reward function g_k: $E_k(T^i) \rightarrow \mathbf{R}^+$, where $g_k(b^j_{T^i})$ represents agent k′s payoff for executing the j-th capability of T^i.

We assume that the agents are rational[4, 11] and each agent tries to maximize its utility. Among all the possible behavior strategies that an agent has, it will choose the one that will lead to its maximum utility. Besides, agents are self-interested, and an agent can reduce its own cost paid out for a task by cooperating with other agents and obtain certain additional utility. So an agent is willing to form a coalition with other agents to increase its own utility.

3) Coalition: A coalition Ξ_{T^i} for task T^i is a tuple $\langle C_{T^i}, alloc_{T^i}, \boldsymbol{u}_{T^i}\rangle$, where $C_{T^i} \subseteq Agent$ and $C_{T^i} \neq \phi$. The utility of a coalition C_{T^i} is represented by a characteristic

Received May 30, 2007; in revised form September 21, 2007
Supported by National Natural Science Foundation of China (60474035), National Research Foundation for the Doctoral Program of Higher Education of China (20060359004), Natural Science Foundation of Anhui Province (070412035)

1. Department of Computer and Information Science, Hefei University of Technology, Hefei 230009, P. R. China 2. Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, P. R. China

DOI: 10.3724/SP.J.1004.2008.00349

function $V(C_{T^i})$, and

$$V(C_{T^i}) = P(T^i) - F(C_{T^i}) \tag{1}$$

$$F(C_{T^i}) = \sum_k \sum_j b_k^j \cdot \varphi\left(T^i, k, j\right) \tag{2}$$

where $F(C_{T^i})$ represents the cost of all members$'$ capability, and $\varphi(T^i, k, j)$ is

$$\varphi(T^i, k, j) = \begin{cases} 1, & \text{agent } k \text{ executes the } j\text{th} \\ & \text{dimensional capability of } T^i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The members in C_{T^i} commonly share the utility $V(C_{T^i})$, and their payoff distribution vector is $\boldsymbol{u}_{T^i} = \{u_{T^i}^1, u_{T^i}^2, \cdots, u_{T^i}^{|C_{T^i}|}\}$, where $u_{T^i}^k$ is the payoff of agent k (See (4)), and $\sum_{A_i \in C} u_i = V(C_{T^i})$.

$$u_{T^i}^k = \sum_j g_k(b_{T^i}^j) \tag{4}$$

$$g_k(b_{T^i}^j) = \frac{b_{T^i}^j}{\sum\limits_{l=1}^{r} b_{T^i}^l} V(C_{T^i}) \cdot \varphi(T^i, k, j) \tag{5}$$

$alloc_{T^i}$ is a task allocation function that associates each dimensional capability $b_{T^i}^j$ with a member of C_{T^i} such that $alloc_{T^i}(b_{T^i}^j) = k$ only if $b_k^j \geq b_{T^i}^j$. The coalition C_{T^i} is capable of performing T^i only if for $\forall\, b_{T^i}^j$, there will be an agent $k \in C_{T^i}$, satisfying $b_k^j \geq b_{T^i}^j$.

4) Controller: Controller is used to determine the optimal coalition for each task and allocate utility for each agent in multi-task coalitions.

The problem of the multi-task coalition parallel formation is that N coalitions must be effectively formed in parallel according to N tasks in $T = \left\{T^1, T^2, \cdots, T^N\right\}$ with the purpose of maximizing the utility of each agent in system coalitions.

2 Multi-task coalition parallel formation strategy

2.1 Markovity of multi-task coalition formation process

Definition 1. A Markov decision process (MDP)[12] model contains: a set of possible states $\boldsymbol{S}$, a set of possible actions $\boldsymbol{A}$, a real valued reward function $R: \boldsymbol{S} \times \boldsymbol{A} \rightarrow \mathbf{R}$, and a state transition function $T: \boldsymbol{S} \times \boldsymbol{A} \rightarrow P(\boldsymbol{S})$. Let $R(s, a, s')$ denote the immediate reward after transition from state s to s' executing action a, and $P(s, a, s')$ denote the state transition probability from state s to s' executing action a.

The essence of MDP is that the effects of an action taken in a state depend only on that state and not on the prior history.

In order to maximize their individual benefits with their available resources and capabilities, the self-interested agents seek each to form coalitions rationally. In other words, the process of multi-task coalition parallel formation is a decision process that each agent selects tasks to execute.

Definition 2. Agents state $\boldsymbol{S}$ is defined as a state vector of all agents in the system and is presented by $(s_1, s_2, \cdots, s_m)$, where s_k $(k = 1, 2, \cdots, m)$ is the state of agent k presented by $s_k = \langle RA_k, TA_k \rangle$, RA_k is the available capabilities of the agent k currently, denoted by $RA_k = < ra_k^1, ra_k^2, \cdots, ra_k^r >$, where ra_k^j $(j = 1, 2, \cdots, r)$ is the jth dimensional residual capability; TA_k is the set of the tasks selected to perform. We let $SS = \{\boldsymbol{S}_i = (s_{i,1}, s_{i,2}, \cdots, s_{i,m}) | i = 1, 2, \cdots\}$ denote the set of all possible agents states.

For task T^i, agent k views the selection of T^i as an action to take that as result on its available capabilities and its gain, and needs to decide what is useful for it: 1) if $ra_k^j \geq b_{T^i}^j$, agent k joins a coalition to execute T^i, consume certain capabilities, and obtain benefits, denoted by J; 2) if $ra_k^j < b_{T^i}^j$, agent k saves its capabilities in order to perform another task, denoted by K.

Definition 3. The set of agent actions is denoted as $Action = \{J, K\}$.

In order to obtain the maximum utility, many agents may be inclined to execute the same tasks, so there may be a collision among agents. To reduce the collision, agent k needs to exchange states to know about other agents$'$ residual capability and decides its state transition probability $P(s_{i,k}, a, s_{i+1,k})$, and thus, agent k will be picked in a bigger probability by the controller to join the task final coalition and obtain its maximum utility. But due to its self-interest, each agent does not want other agents to know its residual capability orienting task T^i and only exchanges whether its residual capabilities can perform task T^i with other agents. So agents$'$ residual capabilities must be first binarized and the binarization results is defined as duration vector.

Definition 4. Duration vector of agent k is defined as $\boldsymbol{Dur}_k = \left(d_k^1, d_k^2, \cdots, d_k^r\right)$, where d_k^j $(j = 1, 2, \cdots, r)$ is as follows.

$$d_k^j = \begin{cases} 1, & \text{if } ra_k^j \geq b_{T^i}^j \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Agent k exchanges its duration vector with other agents by blackboard method and decides its state transition probability $P(s_{i,k}, a, s_{i+1,k})$ as

$$P(s_{i,k}, J, s_{i+1,k}) = \begin{cases} \sum\limits_j \frac{d_k^j}{\sum\limits_{l=1}^{m} d_l^j} \cdot \frac{1}{r}, & \begin{array}{c} \exists\, j \in \{1, 2, \cdots, r\}, \\ ra_k^j \geq b_{T^{i+1}}^j \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Therefore, the system$'$s state transition probability $P(\boldsymbol{S}_i, a, \boldsymbol{S}_{i+1})$ is as follows.

$$P(\boldsymbol{S}_i, J, \boldsymbol{S}_{i+1}) = \prod_k P(s_{i,k}, J, s_{i+1,k}) \tag{8}$$

Orienting task T^i, the selection of action a (J or K) drives the agent into a new state $\boldsymbol{S}_{i+1}$ from state $\boldsymbol{S}_i$, which is shown as (9)∼(12).

$$\boldsymbol{S}_{i+1} = (s_{i+1,1}, s_{i+1,2}, \cdots, s_{i+1,m}) \tag{9}$$

$$s_{i+1,k} = < RA_{i+1,k}, TA_{i+1,k} > \tag{10}$$

$$RA_{i+1,k} = \begin{cases} RA_{i,k} - \sum\limits_j \left(0, \cdots, 0, b_{T^i}^j, 0, \cdots, 0\right), \text{ if } a = J \\ RA_{i,k}, \text{ if } a = K \end{cases} \tag{11}$$

$$TA_{i+1,k} = \begin{cases} TA_{i,k} \cup \sum\limits_j \left\{b_{T^i}^j\right\}, \text{ if } a = J \\ TA_{i,k}, \text{ if } a = K \end{cases} \tag{12}$$

where $j = 1, 2, \cdots, r$. Furthermore, orienting a new task, agent k updates its duration vector $\boldsymbol{Dur}_{i+1,k}$ on the basis of its residual capabilities according to (6).

Claim 1. The process of multi-task coalition parallel formation is a MDP.

Proof. According to the properties of Markov decision process, here we only need to prove the process constructed by $\boldsymbol{S}_i = (s_{i,1}, s_{i,2}, \cdots, s_{i,m})(i \geq 0)$ of agent task selection is a Markov decision process.

According to Definition 1, we only need to prove state $\boldsymbol{S}_{i+1}$ only depends on $\boldsymbol{S}_i$. Note that at time $i+1$, the system state transition probability $P(\boldsymbol{S}_i, a, \boldsymbol{S}_{i+1})$ is determined only by agent′s available capabilities at time i as (7) and (8). Furthermore, the state $\boldsymbol{S}_{i+1}$ only depends on $\boldsymbol{S}_i$ as (9)∼(12). So, this process is a Markov decision process. □

2.2 Agent behavior strategy based on reinforcement learning

Reinforcement learning[12−14] is a method which maps environment states to actions and obtains optimal policy through trial-and-error and interaction with dynamic environment. In the problem of the multi-task coalition parallel formation, the long-term influences of agent behavior should be considered. So Definition 5 will define state value as objective function to decide the optimal actions.

Definition 5. State value $\nu(s_k)$ of agent k is defined as the sum of last payoff in the tasks that have been allocated to agent k before it reaches state s_k.

Definition 6. Immediate reward $r(s'_k)$ of agent k is defined as the reward obtained after action a has been taken and driven agent k to state s'_k from state s_k. For example, if agent k takes action J to execute the jth dimensional capability of task T^{i+1}, it will obtain the reward $g_k(b^j_{T^{i+1}})$, that is $r(s'_k) = g_k(b^j_{T^{i+1}})$.

Being in state $\boldsymbol{S}$, each agent has to select an action to maximize its long-term reward. A behavior strategy π_k is a mapping from state to actions, and the state value s_k is as follows, where γ is a discount factor.

$$v^{\pi^k}(s_k) = r(s_k) + \gamma \sum_{\boldsymbol{S}' \in SS} P(s_k, a, s'_k) v^{\pi^k}(s'_k) \tag{13}$$

The theory of dynamic programming can guarantee at least an optimal strategy π^{k*} for an agent to obtain maximum utility as follows.

$$v^{\pi^{k*}}(s_k) = \max_{a^k \in Action} \left\{ r(s_k) + \gamma \sum_{\boldsymbol{S}' \in SS} P(s_k, a, s'_k) v^{\pi^{k*}}(s'_k) \right\} \tag{14}$$

The basic learning steps are as follows:

1) Initialization: $i = 0$, for each $\boldsymbol{S} \in SS$, $a_k \in Action$, $k = 1, 2, \cdots, m$, make $v(s_{i,k}) = 0$ and initialize state $\boldsymbol{S}_0$.

2) Loop: For $k = 1, 2, \cdots, m$, agent k chooses action a^k_i according to its state transition probability $P(s_{i,k}, a, s_{i+1,k})$, observes $r(s_{i+1,k})$ and $s_{i+1,k}$, and updates $v^{\pi^k}(s_{i,k})$ as (13) and (14), $i = i + 1$.

2.3 Process of multi-task coalition parallel formation

The process of multi-task coalition parallel formation can be described as follows:

1) Each agent decides the set of tasks to execute and the contributive capabilities according to each task through reinforcement learning.

2) Each agent submits its final state $s_{opt} =< RA_{opt}, TA_{opt} >$ to the system, and the controller decides the final coalitions for tasks according to (15) satisfying that the residual capabilities of all agents in coalitions is the least, and the utilization factor of agent capability is maximal.

$$\min\{\sum_{k \in C} RA^k_{opt}\}, C = \bigcup_i C_{T^i} \tag{15}$$

3) The payoff of each agent is allocated according to (4) and (5).

3 Example analysis

Suppose there are 3 tasks $T = \{T^1, T^2, T^3\}$ and 10 agents $Agent = \{1, 2, \cdots, 10\}$. The capability required vector and the utility of each task are the following: $\boldsymbol{B}_{T^1} = (3, 4, 6)$, $\boldsymbol{B}_{T^2} = (2, 5, 3)$, $\boldsymbol{B}_{T^3} = (5, 6, 7)$, $P\left(T^1\right) = 26$, $P\left(T^2\right) = 20$ and $P\left(T^3\right) = 36$. And the capability vector of agents are: $\boldsymbol{B}_1 = (1, 1, 3)$, $\boldsymbol{B}_2 = (2, 2, 4)$, $\boldsymbol{B}_3 = (3, 5, 8)$, $\boldsymbol{B}_4 = (7, 4, 2)$, $\boldsymbol{B}_5 = (5, 7, 7)$, $\boldsymbol{B}_6 = (4, 6, 5)$, $\boldsymbol{B}_7 = (5, 5, 6)$, $\boldsymbol{B}_8 = (3, 2, 1)$, $\boldsymbol{B}_9 = (8, 7, 6)$ and $\boldsymbol{B}_{10} = (10, 10, 10)$.

We will take agent 3 for an example to illustrate agent behavior strategy and the process of multi-task coalition parallel formation.

1) Initialization: $i = 0$, agent 3′s initial state $s_{i,3} = \langle (3, 5, 8), \emptyset \rangle$ and state value $v(s_{i,3}) = 0$.

2) $i = i + 1$, when agent 3 orients the first dimensional capability of each task:

a) When agent 3 orients the first dimensional capability of task T^1, duration vector $\boldsymbol{Dur}_3 = (1, 1, 1)$ can be obtained by (6) through blackboard method, and the selection of action J drives the agent into a new state $s_{1,31} = \left\langle (0, 5, 8), \left\{ b^1_{T^1} \right\} \right\rangle$ defined by (10)∼(12) from $s_{0,3}$, and the state transition probability is $P(s_{0,3}, J, s_{1,31}) = (1/8 + 1/7 + 1/5) \cdot 1/3 = 0.17$ defined by (7), and the state value of $s_{1,31}$ is $\nu(s_{1,31}) = 3$ defined by Definition 5.

b) When agent 3 orients the first dimensional capability of task T^2, $\boldsymbol{Dur}_3 = (1, 1, 1)$, $P(s_{0,3}, J, s_{1,32}) = (1/9 + 1/6 + 1/8) \cdot 1/3 = 0.14$, $s_{1,32} = \left\langle (1, 5, 8), \left\{ b^1_{T^2} \right\} \right\rangle$ and $\nu(s_{1,32}) = 2$ can be obtained by the same methods.

c) When agent 3 orients the first dimensional capability of task T^3, $\boldsymbol{Dur}_3 = (0, 0, 1)$, $P(s_{0,3}, J, s_{1,33}) = (0/5 + 0/4 + 1/3) \cdot 1/3 = 0.11$, $s_{1,33} = \langle (3, 5, 8), \emptyset \rangle$ and $\nu(s_{1,33}) = 0$ can be obtained by the same methods.

As $v(s_{1,31}) > v(s_{1,32}) > v(s_{1,33})$, agent 3 transits to state $s_{1,31}$ in the probability of $P(s_{0,3}, J, s_{1,31}) = 0.17$, and $s_{1,3} = s_{1,31}$, $P(s_{0,3}, J, s_{1,3}) = P(s_{0,3}, J, s_{1,31}) = 0.17$, $\nu(s_{1,3}) = 3$.

3) We can obtain the results as follows by the same methods:

a) When $i = 2$, that is agent 3 orients the 2nd dimensional capability of each task, agent 3 transits to state $s_{2,3} = \left\langle (0, 0, 8), \left\{ b^1_{T^1}, b^2_{T^2} \right\} \right\rangle$ in the state transition probability of $P(s_{1,3}, J, s_{2,3}) = 0.1$, and the value of state $s_{2,3}$ is $\nu(s_{2,3}) = 8$.

b) When $i = 3$, agent 3 orients the 3rd dimensional capability of each task, agent 3 transits to state $s_{3,3} = \left\langle (0, 0, 1), \left\{ b^1_{T^1}, b^2_{T^2}, b^3_{T^3} \right\} \right\rangle$ in the state transition probability of $P(s_{2,3}, J, s_{3,3}) = 0.11$, and the value of state $s_{3,3}$ is $\nu(s_{3,3}) = 15$.

The main states and the state value of agent 3 are shown in Table 1 (see next page).

4) Each agent submits its final state to the system. The possible coalitions for task T^1 are $\{3, 4, 7\}$, $\{3, 4, 9\}$, $\{8, 4, 7\}$ and $\{8, 4, 9\}$. The possible coalitions for task T^2 are $\{2, 3, 1\}$. The possible coalitions for task T^3 are $\{5, 6, 3\}$ and $\{7, 6, 3\}$.

5) The controller decides the final coalitions for tasks according to (15) (see Table 2).

6) The utility of each agent is obtained according to (4) and (5) (see Table 3).

Table 1 The main states and the state values of agent 3

State s	State values
$\langle <3,5,8>, \emptyset \rangle$	0
$\left\langle <0,5,8>, \left\{ b_{T1}^{1} \right\} \right\rangle$	3
$\left\langle <0,0,8>, \left\{ b_{T1}^{1}, b_{T2}^{2} \right\} \right\rangle$	8
$\left\langle <0,0,1>, \left\{ b_{T1}^{1}, b_{T2}^{2}, b_{T3}^{3} \right\} \right\rangle$	15

Table 2 The task-oriented coalitions

Task	T^1	T^2	T^3
Coalition	$\{3,4,7\}$	$\{2,3,1\}$	$\{7,6,3\}$

Table 3 The agents′ utility

Agent	1	2	3	4	6	7
Utility	3	2	15	4	6	11

As can be seen, the strategy in this paper can effectively and form in parallel optimal coalitions for multi-task, avoid coalition lock and resource conflict, adequately consider agent behavior when maximizing system′s utility, and satisfy each agent′s need. Moreover, the payoff distribution accords to work, and ascertains coalition stability and tasks solving efficiency.

4 Conclusion

In this paper, we consider situations where multiple tasks should be performed in parallel by group of agents, and a multi-task coalition lock parallel formation strategy is proposed. First, the conclusion is testified theoretically that the process of multi-task coalition formation is a Markov decision process. Second, reinforcement learning is used to solve agents′ behavior strategy, and the process of coalitions formation is given. Finally, an example is shown to illuminate that the strategy can effectively and parallel form multi-task coalitions in multi-task oriented domains.

References

1 Ketchple S. Forming coalitions in the face of uncertain rewards. In: Proceedings of the 12th National Conference on Artificial Intelligence. Seattle, USA: AAAI Press, 1994. 414−419

2 Zhang Guo-Fu, Jiang Jian-Guo, Xia Na, Su Zhao-Pin. Solutions of complicated coalition generation based on discrete particle swarm optimization. *Acta Electronica Sinica*, 2007, **35**(2): 323−327 (in Chinese)

3 Zoltkin G, Rosenschein J S. Coalition, cryptography, and stability: mechanisms for coalition formation in task oriented domains. In: Proceedings of the 12th National Conference on Artificial Intelligence. Seattle, USA: AAAI Press, 1994. 432−437

4 Zhang G F, Jiang J G, Xia N, Su Z P. Particle swarms cooperative optimization for coalition generation problem. In: Proceedings of the 6th International Conference on Simulated Evolution and Learning. Hefei, China: Springer, 2006. 166−173

5 Shehory O, Kraus S. Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents. In: Proceedings of International Conference on Multi-agent Systems. Kyoto, Japan: MIT Press, 1996. 330−337

6 Conitzer V, Sandholm T. Computing shapely values, manipulating value division schemes, and checking core membership in multi-issue domains. In: Proceedings of the 19th National Conference on Artificial Intelligence. California, USA: AAAI Press, 2004. 219−225

7 Luo Yi, Shi Chun-Yi. The behavior strategy to form coalition in agent cooperative problem-solving. *Chinese Journal of Computers*, 1997, **20**(11): 961−965 (in Chinese)

8 Jiang Jian-Guo, Xia Na, Yu Chun-Hua. The coalition formation strategy based on capability vector contribution-rate and auction. *Acta Electronica Sinica*, 2004, **32**(S1): 215−217 (in Chinese)

9 Kraus S, Shehory O, Taase G. Coalition formation with uncertain heterogeneous information. In: Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems. Melbourne, Australia: ACM Press, 2003. 1−8

10 Hosam H, Khaldoun Z. Planning coalition formation under uncertainty: auction approach. In: Proceedings of the 2nd International Conference on Information and Communication Technologies. Damascus, Syria: IEEE, 2006. 3013−3017

11 Klusch M, Gerber A. Dynamic coalition formation among rational agents. *IEEE Journal on Intelligent Systems*, 2002, **17**(3): 42−47

12 Kaelbling L P, Littman M L, Moore A P. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 1996, **4**: 237−285

13 Song Mei-Ping, Gu Guo-Chang, Zhang Guo-Yin. Survey of multi-agent reinforcement learning in Markov games. *Control and Decision*, 2005, **20**(10): 1081−1090 (in Chinese)

14 Tan M. Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the 10th International Conference on Machine Learning. Alberst, USA: Morgan Kaufmann Publisers, 1993. 330−337

JIANG Jian-Guo Professor at the School of Computer and Information, Hefei University of Technology. His research interest covers sensor and intelligent control, signal and information processing.
E-mail: jjg@ah165.net

SU Zhao-Pin Ph. D. candidate. Her research interest covers distributed artificial intelligence and evolving computing. Corresponding author of this paper.
E-mail: szhpin@163.com

QI Mei-Bin Associate professor, Ph. D. candidate at the School of Computer and Information, Hefei University of Technology. He is a Ph. D. candidate. His research interest covers computer control and image processing.
E-mail: qimeibin@163.com

ZHANG Guo-Fu Ph. D. candidate. His research interest covers swarm intelligence and MAS theory.
E-mail: zhang197933@163.com