International Journal of Database Theory and Application
Vol.8, No.1 (2015), pp.29-38
http://dx.doi.org/10.14257/ijdta.2015.8.1.04

Dynamic Coalition Formation Based on Multi-sided Negotiation

Lin Xiang[1] and Haijun Tao

College of Information Engineering, China Jiliang University, Hangzhou, Zhejiang, China,
{xianglin, hjtao }@cjlu.edu.cn

Abstract

The coalition formation is an important aspect of multi-agent negotiation and cooperation. Based on the Bilateral Shapley-Value, a multi-sided eigenvalue is presented by using the reference of n-person stochastic cooperative game. It is obvious when multi-sided eigenvalue have superadditivity, rational agents will combine to one coalition. A dynamic coalition formation algorithm is constructed based on the eigenvalue. Procedures of multi-sided negotiation, agent's negotiation and coalition condensing are introduced in detail. In the end, the complexity, validity, coalition stability and parameter's function of the algorithm is given. According to these, the correctness of the algorithm is proven.

Keywords: *multi-agent; eigenfunction; multi-sided eigenvalsue; multi-sided negotiation; coalition*

1. Introduction

Negotiation is the key issue for the multi-agent system to harmonize and solve the conflict of the MAS target, knowledge and resource, and it is an agent mutual mechanism based on the communication language. Through negotiation agents come to an agreement to some questions. More and more scholars pay attention to MAS negotiation problem and make research on it in different areas, such as DAI, social psychology and economy, *etc*.

As an important mechanism which can improve the multi-agent system's performance, coalition draws a lot of researchers' attention. Many methods are proposed. Klusch, *et. al.*, [1, 2] give an bilateral coalition formation algorithm by importing the BSV (Bilateral Shapley-Value), but it not adapt to the coalition that involve more than three agents. Kraus [3] presents a coalition formation algorithm based on queuing networks. Sandholm and Lesser [11] adopt a model of bounded rationality where computation resources are costly. The optimal coalition structure and its stability are discussed. In paper [12], they present an algorithm that establishes a tight bound with the minimal amount of search to form a coalition. Based on Sandholm's work, Hu Shan-Li and Shi Cun-Yi present an optimize coalition structure formation algorithm [4, 5]. All of these works are focused on maximizing the global reward. These are the coalition formation problem in DAI in essential. Shehory [6] presents an algorithm based on a set-covering algorithm. It depends on the global reward of each two agent forms the coalition, so global knowledge is required. On the other hand, many optimal models are proposed aiming at special application regions. The Tribase model [7] uses trust degree, acquaintance degree to form and adjust the coalition. Tong Xiang-Rong [8] presents a long term coalition maintenance algorithm based on fuzzy ally relation. However, the efficiency, quality and complexity of

[1] The corresponding author

Manuscript received ; revised , accepted

ISSN: 2005-4270 IJDTA

coalition formation which involves large range of participants still need more research works.

To improve the quality and efficiency of MAS cooperation, a multi-sided eigenvalue is presented by using the reference of n-person stochastic cooperative game and the bilateral shapley-value. We expound when multi-sided eigenvalue fulfils superadditivity, rational agents will form a coalition. According to the feature, we construct a dynamic coalition formation algorithm to find the coalitions which fulfill eigenvalue's superadditivity. Procedures of multi-sided negotiation, agent's negotiation and coalition condensing are introduced in detail. In the end, the complexity, validity, coalition stability and parameter's function of the algorithm is given. Potential coalition formation maybe success even when superadditivity can't be satisfied or some information can't be acquired. According to these, the correctness of the algorithm is proven.

2. Basic Definitions

Definition 1: Suppose t is a task, $A = \{a_1, a_2, \cdots\}$ is set of agent. For $S \subseteq A$, $v(S)$ is the max utility no matter what policy that agents outside S perform in the execution of task t. $v(S)$ is called the coalition's eigenfunction. $v(\phi) = 0$.

P_S is the policies that agent in S can take. $P_{A/S}$ is the policies that agent in A/S can take, we have:

$$v(S) = \max_{x \in P_S} \min_{y \in P_{A/S}} \sum_{a_i \in S} e_i(x, y) \qquad \text{Formula 1}$$

We can get theorem 1 when use eigenfunction to express the coalition's utility.

Theorem 1: To coalition $S, T \in A$, $S \cap T = \phi$, eigenfunction fulfill superadditivity, that is: $v(S) + v(T) \leq v(S \cup T)$ (Ref[11] shows the proving process).

We suppose that the agent is rational when forms a coalition (they are driven to form a coalition only when they can get more reward). Utility vector $u(S) = \sum_{a_i \in S} p_i$ is used to define coalition S's utility function. Ref [10] defines the bilateral eigenfunction $\Psi_{S \cup T}(S) = \frac{1}{2}u(S) + \frac{1}{2}[u(S \cup T) - u(T)]$ based on $u(S)$. It also proves that S and T hope to form a coalition when S, T are rational and $\Psi_{S \cup T}(S) \geq u(S)$ (that is $u(\square)$ have superadditivity).

A multi-sided eigenfunction is defined by extending the bilateral eigenfunction:

Definition 2(Multi-sided eigenfunction):

$$\Psi_C(C_i) = \frac{m-1}{m}u(C_i) + \frac{1}{m}\left[u(C) - \sum_{C_j \in C, i \neq j} u(C_i)\right] \qquad \text{Formula 2}$$

where $C = \bigcup_{j=1}^{m} C_j, C_i \cap C_j = \phi$ when $i \neq j$.

According ref [1], we can get the following theorem easily.

Theorem 2: For a set $A = \bigcup_{i=1}^{n} a_i$ of rational agents, $u(\cdot)$ have superadditivity based on the definition of multi-sided eigenfunction, $\forall a_i \in A$ want to form cooperation coalition with $\bigcup_{j=1, i \neq j}^{m} a_j$.

The eigenfunction have superadditivity. Agents in coalition A would like to form a coalition when $u(\cdot)$ is eigenfunction and agent is rational. It is hard to get the eigenfunction according to formula 1. This requires the agent have global knowledge of the environment. For a MAS in dynamic environment, the problem becomes more complex. Now we present a dynamic coalition formation algorithm based on multi-sided eigenfunction by using the contract net procotol [10] as a reference.

3. Coalition Formation Algorithm based on Multi-Sided Negotiation

We suppose the agents involved are rational and the agent should know how much resource it can save when cooperate with other agents.

The basic work mode of the system is: An agent receives a task T that needs to cooperate with other agents, it broadcasts the task to all of the others as a manager. Agents evaluate if they can participate in the task and the minimal reward they want according their capabilities. Agents report these information to the manager of task T. The manager combines agent that have different types into sets $A_1, A_2, \cdots A_m$ in a given time, each set can perform the task without the help of others. Then the manager sends all possible combinations to every agent in the set. The agents decide if they can form a coalition by multi-sided negotiation and send the negotiation result to the manager. The manager selects a final coalition which would take the task T from all the coalitions.

A coalition formation algorithm is presented (shown as Table 1) based on the contract net protocol

Table 1. Dynamic Coalition Formation Procedure based on Multi-sided Cooperation Algorithm

- The manager receives task T and the given reward r, analyzes the agent types h that is needed and then broadcasts these information to all of the agents.
- The manager waits for the agent's reply in a given time t_1, then goto next step.
- Supposing there are n agents reply the task T, the manager makes completely combinations of them, gets sets $A_1, A_2, \ldots A_m$, for each A_k, $\mathrm{type}(\mathrm{A}_k) = |A_k| = h$.
- For each A_k ($k = 1, 2, \ldots m$), the manager sends the A_k's member list and the minimal reward $\{u(\{a_1\}), u(\{a_2\}), \ldots u(\{a_n\})\}$ that the agent announced to all $a_i \in A_k$.
- The manager waits the results of multi-sided negotiation in given time t_2, signed them as $A_{pot}(1), A_{pot}(2), \ldots,$ broadcast $coalition\_potential(A_{pot}(q))$, $q = 1, 2, \ldots$ to all the agents.
- The manager selects final coalition according to $A_{final} = \min_{A_{por}(q)} \sum_{a_i \in A_{pot}(q)} u(\{a_i\})$, then notifies agents in A_{final} to execute task T.

Agent monitors the task information, judges if it will join a coalition according to multi-sided negotiation and multi-sided eigenvalue. The concrete procedure is shown as Table 2:

Table 2. Multi-sided Negotiation Algorithm of Agent

- To the task and relate information that received, the agent compares its own type and the required types, judges if it want to join the task T .
- If the agent want to join T , it gives the minimal ideal reward $u(\{a_i\})$, returns $u(\{a_i\})$ and its type to the manager.
- The agent receives member list and minimal reward list from the manager.
- For set $A_k(a_i \in A_k)$, agent a_i calculates the resource $\delta_C(A_k)$ it can save when cooperates with the other members in A_k according to its own knowledge and experience.
- For every set $A_k(a_i \in A_k)$, a_i calculates the cooperation reward $\Psi_{A_k}(a_i) = \frac{h-1}{h}u(\{a_i\}) + \frac{1}{h}\left[r + \delta_C(A_K) - \sum_{a_k \in A_k, i \neq j} u(\{a_j\})\right]$, sorts all A_k ($\Psi_{A_k}(a_i) \geq u(\{a_i\})$) as $A_1^{(i)}, A_2^{(i)},... A_{m_i}^{(i)}$ by incremental. $L(A_k^{(i)}) = k$ means the place of $A_k^{(i)}$ in the list.
- Gets into the multi-sided negotiation.
- Returns the negotiation result $A_k^{(i)}$ to the manager, waits the final coalition that selected by the manager decision in a given time.

After the agent generates the potential cooperate coalition $A_1^{(i)}, A_2^{(i)},... A_{m_i}^{(i)}$, it decides the possible coalition through multi-sided negotiation in sequence. The multi-sided negotiation relies on the calculation of multi-sided eigenvalue. When multi-sided eigenvalue is superadditivity, the agent will form a coalition. Otherwise, the algorithm will use some strategies to guarantee the generation of potential coalition. Multi-sided negotiation is shown as Table 3:

Table 3. Multi-sided Negotiation Algorithm

- **Step1:** Initialize, set $k = 1$;
- **Step2:** Agent $a_i(i = 1,2,..., n)$ sends coalition investigation $coalition\_offer(a_i, A_k^{(i)})$ to all agent $a_j \in A_k^{(i)}$, $coalition\_offer(a_i, A_k^{(i)})$ means a_i wants to make coalition with the members in $A_k^{(i)}$;
- **Step3:** The agent waits for the other agent's agree reply $coalition\_agree(a_j, A_k^{(i)})$ or deny reply $coalition\_deny(a_j, A_k^{(i)})$, $a_j \in A_k^{(i)}$. The agent also receives other agent's coalition investigation $coalition\_offer(a_j, A_k^{(j)})$, $a_j \in A_k^{(j)}$ or the potential negotiation result $coalition\_potential(A_{pot}(q))$ sent by the manager;
- **Step4:**
  - The agent marks a_j in $A_k^{(i)}$ to $agree$ when receives $coalition\_agree(a_j, A_k^{(i)})$;
  - When receives $coalition\_offer(a_j, A_k^{(j)})$, the agent searches the place of $A_k^{(j)}$ in a_i list- $L(A_k^{(j)})$. If $L(A_k^{(j)}) \leq k$, mark a_j in $A_k^{(i)}$ to $agree$, replies a_j with $coalition\_agree(a_i, A_k^{(j)})$, else replies a_j with $coalition\_deny(a_i, A_k^{(j)})$;

- When receives $coalition\_deny(a_j, A_k^{(i)})$, the agent marks a_j in $A_k^{(i)}$ to $deny$, goto Step6;
- When receives $coalition\_potential(A_{pot}(q))$, the agent gets into abridgement procedure;

- **Step5:** If $\forall a_j \in A_k^{(i)}$ was marked as $agree$, $A_k^{(i)}$ is the a_i's potential coalition, exit the negotiation procedure;
- **Step6:** If $\exists a_j \in A_k^{(i)}$ not marked as agree, goto Step3;
- **Step7:** If $k < m$, let $k = k + 1$, goto Step2 else goto Step3;

Table 4 shows the potential coalition abridgement procedure of a_i.

Table 4. Abridgment Procedure of Agent a_i

- **Step1:** Set $A_{temp} = \phi$;
- **Step2:** $while\ (A_k^{(i)} \cap A_{por}(q))! = \phi\ \ and\ \ k > 1)\ k--$;
- **Step3:** $if\ (k > 1)\ A_{temp}^i = A_k^i$;
- **Step4:** Delete $A_k^{(i)}$ from $A_1^{(i)}, A_2^{(i)}, \ldots A_{m_i}^{(i)}$ where $A_k^{(i)} \cap A_{por}(q) \neq \phi$;
- **Step5:** If the set is not empty, sort the members in the set according $\Psi_{A_k}(a_i)$ by decrement, else exit the abridgement procedure;
- **Step6:** If $A_{temp}^{(i)}! = \phi$, let $k = L(A_{temp}^{(i)})$, goto Step3 else let $k = 1$;
- **Step7:** Exit abridgement procedure;

4. Analyses of the Algorithm

We will analyze the complexity, validity, parameters and the coalition's stability in the following section.

A. Complexity

The algorithm's complexity centralizes on the step1 and step3 in Table 1. The other steps' complexity are $O(n)$. In step3, the manager needs to make completely combinations of n agents which reply the task T on the $\text{type}(A_k) = |A_k| = h$ constraint. The constraint guarantees the coalition can execute the task T and the sub-tasks can be executed by one agent without overlap. This will decrease the competition in coalition.

There are two factors that influence the complexity in step3: 1) agent count n that reply the task T; 2) agent types that needed to perform the task T. The simplest status is $n = h$ or $n - h + 1$ of n agents are the save type whereas the other $h - 1$ agents are different types. In this status, the complexity is linear-$O(n)$. When the distribution of agent's type is uniform and $n = lh\ (l = 2,3,\ldots.)$, the complexity maximized-$O(l^h)$. When h is constant and l is variable, the complexity of step3 is polynomial time. If l is constant and h is variable, the complexity becomes exponential time. This problem occurs when the agent types which are needed to solve the task T extremely increase. The problem can be solved by restricting the range of agent that participate in the cooperation or filtering the agent according some factors (such as less minimal reward $u(\{a_i\})$ of the agent) to decrease the complexity. Further researches of these mechanisms are expected.

In the multi-sided negotiation, the worst case is the negotiation continues to $k = m_i$. The worst case of m_i will reach h^{l-1}, so there are $h \cdot h^{l-1} = h^l$ *offers* be sent. In this case, the complexity will also reach $O(h^l)$. When using the negotiation algorithm in a large MAS, it is necessary to use some mechanisms to restrict the complexity.

B. Validity of the Algorithm

The validity we discussed means if the negotiation can generate at least on potential coalition.

We suppose the negotiation procedure does not generate a potential coalition when $k = \max_{i=1,2,..n} m_i - 1$. That is to $\forall a_i, l \le k$, no $A_l^{(i)}$ exist which members are all marked as *agree*. From the negotiation we can see that the algorithm will not get into abridgement procedure. Supposing $m_j = \max_{i=1,2,...,n} m_i$, m_j will send $coalition\_offer(a_j, A_{last})$ to the other members in $A_k^{(j)}$ (we mark $A_k^{(j)}$ as A_{last} to avoid confusion). For $\forall a_i \in A_{last}(a_i \ne a_j)$ that receives the offer, it will get $L(A_{last}) \le m_i \le m_j = k$ when judging the place of A_{last}. It will mark a_j in A_{last} to *agree* according step3 in Table **3**.

Because $L(A_{last}) \le m_i$, $\forall a_i \in A_{last}$ already sent $coalition\_offer(a_i, A_{last})$ to other members $a_l \in A_{last}(a_l \ne a_l)$ in A_{last} when $k_i \le m_i$. According to the algorithm, a_l has two cases of the offer:

(1) When $L(A_{last}) \le k_l$, mark a_i in A_{last} to *agree*, then send the agree reply $coalition\_agree(a_l, A_{last})$ to a_i;

(2) When $L(A_{last}) > k_l$, send deny reply $coalition\_deny(a_l, A_{last})$ to a_i;

For the second case, because a_l will increase k_l when not get the negotiation result and k_l will reach $k_l = L(A_{last})$ finally, a_l will send $coalition\_offer(a_l, A_{last})$ to a_i. For a_i, it is obvious that $L(A_{last}) > k_l$, so a_i will mark a_l as *agree* and sends the $coalition\_agree(a_i, A_{last})$ to a_l. After this step, a_i and a_l in A_{last} are marked to *agree*. Because a_i and a_l are arbitrary beyond a_j, so other members in A_{last} except a_j are all marked as *agree* when a_j sends $coalition\_offer(a_j, A_{last})$ to other members in A_{last}. a_j in a_l's A_{last} will be signed as *agree* because $\forall a_i \in A_{last}(a_i \ne a_l)$ fulfils case (1) and $coalition\_offer(a_i, A_{last})$ is sent, so all members in a_j's A_{last} are marked as *agree*. We get the potential coalition A_{last} of the negotiation procedure. To this extremely case, A_{last} is the only one potential coalition and the final negotiation result.

C. Analysis of the parameter

There is another important parameter $\delta_C(A_k)$ in the algorithm outsides the h and n in the complexity analysis.

$\delta_C(A_k)$ depends on the agent's experience and knowledge. It influences the A_k place in the set list. $\delta_C(A_k)$ varies with the A_k 's member's experience and knowledge. In general, the more cooperation experience and knowledge A_k 's member have, the larger $\delta_C(A_k)$ is. The $\Psi_{Ak}(a_i)$ calculated by formula 2 varies by this reason.

From the algorithm we can know that if A_k has a former place in the set list, it will also get a higher priority of being the potential coalition. This matches the case that agent want to maximize its own reward. The algorithm finds a acceptable coalition on the basis of agent's desirability of maximizing personal reward. There is a extreme circus: none of the agents has knowledge and experience about any coalition, that is $\delta_C(A_k) = 0$. Formula 2 becomes:

$$\Psi_{A_k}(a_i) = \frac{h-1}{h} u(\{a_i\}) + \frac{1}{h}\left[r - \sum_{a_j \in A_k, i \neq j} u(\{a_j\}) \right] \qquad \text{Formula 3}$$

r is a const value given by the manager. For agent a_i, $u(\{a_i\})$ is also constant. When sorting the set list according to $\Psi_{A_k}(a_i)$ that calculated by formula3, A_k will be the first member in all $a_i(a_i \in A_K)$'s set list if $\sum_{a_i \in A_K} u(\{a_i\}) = \min_{i=1,2,\dots m} \sum_{a_i \in A_L} u(\{a_i\})$. A_k will become a potential coalition immediately according to the multi-sided negotiation procedure. At the same time, the manager will get $A_{final} = A_k$ when selecting task T 's final coalition by the reference of $A_{final} = \min_{A_{por}(q)} \sum_{a_i \in A_{pot}(q)} u(\{a_i\})$. The coalition that has minimal sum of minimal anticipation reward will be the final coalition.

D. Coalition's Stability

Agent in MAS wants to maximal its own reward with unconcern for global reward. So it is a problem that if the agent will maintain the coalition. From theorem 2, all agents would like to form a coalition when $u(\cdot)$ is superaddivity and allots the reward according to formula 2. The coalition is stable. When $u(\cdot)$ can not fulfil superaddivity, there are many agents of same type in a large coalition. The task is divided into more sub-tasks with the increasing of same agents. This will raise the complexity of task cooperation and resource allocation. So a large coalition makes no sense. A stable coalition is not the only guideline. Minimizing and simplifying of the coalition is an important guideline.

The embodiment of task's complexity is not only the workload but also the pluralism and relation. According to this, agent of same type should be minimized in a coalition which takes a complex task. The complete combination of agent which replies the task T in Table **1** should minimizes the coalition's size. The coalition contains only one agent of certain type, this avoids the conflict of agents.

5. Experiment

Let's take region F1 handling typesetting for example as algorithm testing, assume there are three agents in F1, which are A, B and C, whose sub-plan is $Plan\ (A) = ACT\ 1\ ,\ ACT\ 2\ ,\ ACT\ 3$ =go to printing house, fetch sample manuscript, return. $Plan\ (B) = ACT\ 4\ ,\ ACT\ 5\ ,\ ACT\ 6$ =go to printing house, typesetting, return. $Plan\ (C) = ACT\ 7\ ,\ ACT\ 8\ ,\ ACT\ 9$ =go to printing house, binding, return. There are pairwise orthogonal among A, B and C. when A accepts the commission of B and executes its sub-plan, modify $Plan\ (A) = ACT\ 1\ ,\ ACT\ 2\ ,\ ACT\ 5\ ,\ ACT\ 3\ ,\ ACT\ '$ = go to printing house, fetch sample manuscript, typesetting, return, give back to B, here $ACT\ '$ has extra cost $C_{add}(A)$.

Assume each cost is:

$C(A) = C(B) = C(C) = 0$

$C_{add}(A) = C_{add}(B) = C_{add}(C) = 1$

$C_1 = C_3 = 8$, $C_4 = C_6 = 9$, $C_7 = C_9 = 7$, $C_2 = 2$, $C_5 = 4$, $C_8 = 3$

Then：

$C_{exe}(A) = C_1 + C_2 + C_3 = 18$

$C_{exe}(B) = C_4 + C_5 + C_6 = 22$

$C_{exe}(C) = C_7 + C_8 + C_9 = 17$

$C_{con}(A) = 2$, $C_{con}(B) = 4$, $C_{con}(C) = 3$

We compared the standard multi-sided negotiation algorithm, region based multi-sided negotiation algorithm and the algorithm proposed in this paper, the testing results are shown in Table 5.

Table 5. Comparison of Efficiency and Cost at Some Times

Cooperation Times	1	2	3	4	5	6	7	8	9
Multi-Negotiation Algorithm	20,14	26,19	32,21	44,25	56,28	57,30	59,31	60,31	61,31
Region Based Algorithm	17,15	25,25	30,32	33,41	36,48	38,52	38,54	38,55	39,57
Our Algorithm	19,15	25,24	32,33	42,40	48,44	52,48	56,50	58,54	59,56

The item in the table is the cooperation efficiency percentage and the system overhead reduction percentage. It can be seen at the beginning, the cooperation efficiency of the algorithm in this paper slightly lower than the multi-sided negotiation algorithm, but higher than the region based multi-sided negotiation algorithm, and reducing overhead it is basically the same as the region based multi-sided negotiation algorithm, but higher than the multi-sided negotiation algorithm. With the increase of cooperation times, the cooperation efficiency of the multi-sided negotiation algorithm increases rapidly, but the effect on reducing system overhead is not obvious. The region based multi-sided negotiation algorithm has satisfied effect on system overhead reduction with the increase of cooperation times, but not satisfied on improving cooperation efficiency. The algorithm improved in this paper has both satisfied effects on cooperation efficiency and overhead reduction with cooperation times increases.

6. Conclusion

For the construction requirement of large scale MAS, the multi-sided eigenvalue is presented by using the reference of n-person stochastic cooperative game based on the Bilateral Shapley-Value. It is obvious when multi-sided eigenvalue have superadditivity, rational agents will combine to one coalition. A dynamic coalition formation algorithm is constructed based on the eigenvalue. Procedures of multi-sided negotiation, agent's negotiation and coalition condensing are introduced in detail. In the end, the complexity, validity, coalition stability and parameter's function of the algorithm is given. According to these, the correctness of the algorithm is proven.

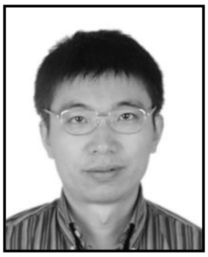
With the increasing of system and the task's complexity, the algorithm takes exponential time. Even though we discuss some clues to solve the problem, it still needs further research. To a large extent, the efficiency and effect of the algorithm rely on the agent's minimal reward and cost reduction of agent when cooperating with others. These require the knowledge of other cooperators. Such information is difficult to get in an open MAS. It also needs further research.

References

[1] M. Wooldridge and P. E. Dunne, "On the Computational Complexity of Qualitative Coalitional Games", Artificial Intelligence, vol. 158, **(2004)**, pp. 27–73.
[2] M. Klusch and O. Shehory, "Coalition Formation Among Rational Information Agents", Lecture Notes in Artificial Intelligence Series, vol. 1038, **(1996)**, pp. 204-217.
[3] S. Kraus and T. Plotkin, "Algorithms of distributed task allocation for cooperative agents", Theoretical Cmputer Science, vol. 242, **(2000)**, pp. 1-27.
[4] M. Cao, "Decision Making Model for Intelligent Agent in Automated Negotiation", Journal of Software, vol. 6, no. 8, **(2011)**, pp. 1537-154.
[5] B. Liu, "Efficient Trust Negotiation based on Trust Evaluations and Adaptive Policies", Journal of Computers, vol. 6, no. 2, **(2011)**, pp. 240-245.
[6] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation", Artificial Intelligence, vol. 101, **(1998)**, pp. 165-200.
[7] Y.-I. Lin, K.-W. Tien and C.-H. Chu, "Multitarget Hierarchical Negotiation for Distributed Design in Collaborative Product Development", IEEE Transactions on Automation Science and Engineering, vol. 11, no. 3, **(2014)**, pp. 881-892.
[8] C. Bartolini, C. Preist and N. R. Jennings, "A software framework for automated negotiation", Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications, Springer Verlag LNCS, vol. 3390, **(2005)**, pp. 213-235.
[9] J. Zhan and X. Li, "Flocking of Multi-Agent Systems Via Model Predictive Control Based on Position-Only Measurements", IEEE Transactions on Industrial Informatics, vol. 9, no. 1, **(2013)**, pp. 377-385.
[10] R. G. Smith, "The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver", IEEE Transactions on Computers, vol. C-29, no. 12, **(1980)**, pp. 1104-1113.
[11] T. Rahwan and N. Jennings, "Coalition Structure Generation: Dynamic Programming Meets Anytime Optimisation", Proc 23rd Conference on AI (AAAI), Chicago, USA, **(2008)** October 156-161.
[12] T. W. Sandholm, K. Larson and M. R. Andersson, "Anytime coalition structure generation with worst case guarantees", Proceedings of the 15th National Conference on Artificial Intelligence, Menlo Park, CA, **(1998)** May, pp. 46-54.

Authors

Lin Xiang, born in 1975, is a lecturer in China Jiliang University. She received master degree from Harbin Institute of Technology, and her major is computer application. Her research interests are in the areas of computer architecture and multi-agent system theory.



Tao Haijun, born in 1975, is Ph.D. and associate professor in China Jiliang University. He received PhD degree from Harbin Institute of Technology, and his major is computer application. His research interests include multi-agent system theory and machine learning technology.

