Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

# AN EXTENDED CONTRACT-NET NEGOTIATION MODEL BASED ON TASK COALITION AND GENETIC ALGORITHM

**HAI-JUN TAO, YA-DONG WANG, MAO-ZU GUO**

School of Computer Science & Technology, Harbin Institute of Technology, Harbin, 150001, China
E-MAIL: hjtao@hit.edu.cn, ydwang@hit.edu.cn, maozuguo@hit.edu.cn

**Abstract:**

**Multi-agent negotiation has been one of the key problems in the multi-agent research area. An extended contract-net negotiation model based on task coalition and genetic algorithm is presented after analyzing the advantage and disadvantage of the classical contract-net negotiation model. Formalized definition method and coalition generation algorithm are given. A specialized genetic algorithm, which is optimized by optimized initial colony selection, optimized parent crossover/mutation and the using of Metropolis rule, is used to solve the task allocation in the coalition. The algorithm improves the efficiency of task allocation and reduces the communication cost. By testing and analyzing an example of a missile defense system, it is proved that the model can reduce the negotiation cost effectively contrast with the classical contract-net model on the basis of ensuring the negotiation quality.**

**Keywords:**

**Multi-agent system; Negotiation; Task coalition; Generic algorithm; Task allocation**

## 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 researches on it in different areas, such as DAI, social psychology and economy, etc.

The classical multi-agent negotiation model is the contract-net negotiation model presented by R.Davis and R.G.Smith in 1980 [1]. By adhibiting the market invite–bid-hit mechanism, it consigns the system tasks distribution in order to solve the conflict of resource and knowledge, and is applied to the research of MAS negotiation by many researches later. There are some disadvantages to solve multi-agent negotiation in the classical contract-net negotiation model, mainly exists that system load will rise rapidly with the expand of the issue's scale, the quality of the negotiation can't be guaranteed and the dynamic change of the task environment is hardly to adapt, etc. Therefore it has been detailed and extended in actual applications, such as Sandholm introduced the concepts of hiberarchy and boundary cost computation [2], [3], Chen Xueguang introduced the bid threshold [4] to the decision produce of contract-net bid and hit in order to restrict the computation and communication cost in negotiation, Fischer introduced the mechanism of temporal trust/refuse and simulation trade in contract-net to optimize tasks distribution [5], and Bouzouia used learning mechanism to optimize tasks distribution in contract-net [6]. Because of the characteristic of contract-net, there are still some problems not solved completely.

This paper presents an extended contract-net MAS negotiation model based on task coalition and genetic algorithm for the requirement of large-scale distributed MAS system. According to the different functions, agents are classified into three roles: management agent, command agent and execution agent. In the third and forth section, the formalized definition method based on task coalition and coalition generation algorithm are given in terms of the architecture. A specialized genetic algorithm, which is optimized by optimized initial colony selection, optimized parent crossover/mutation and the using of Metropolis discriminate rule, is used to solve the task allocation in the coalition, which improves the efficiency of task distribution and reduces the communication cost. In the fifth section, by testing and analyzing a missile defense system, it is proved that the model can reduce the negotiation cost effectively contrast with the classical contract-net model.

## 2. Coalition based on task

In some multi-agent systems, single agent is responsible for task solving, it is enough when the scale is small and the task is relatively simple. When the task scale is large and complex, it needs multi agents cooperated to

**1-4244-0973-X/07/$25.00 ©2007 IEEE**

implement the task. Therefore, in terms of the task requirement, we organize the agents with the capability to implement the task to form the coalition, take charge and execute the task by the unit of coalition, which improves the capability of the system to solve problems and reduces the communication cost during the system cooperation procedure.

### 2.1. Task coalition

Generating agent coalition through cooperation to take charge of the task can improve the solving efficiency of MAS and implement the tasks single agent can't accomplish independently. This paper presents a coalition strategy based on tasks. First some basic concepts needed are defined.

*1. Agent and its capability specification*

All agents in the system compose a set $A=\{a_1,a_2,\cdots,a_n\}$, every $a_i$ in the set has its capability vector $B_{a_i}=(b_1^{a_i},b_2^{a_i},\cdots,b_r^{a_i})$, and the cost vector $COST_{a_i}=(\cos t_1^{a_i},\cos t_2^{a_i},\cdots,\cos t_r^{a_i})$ agent needs to expend corresponding to capability set. $b_j^{a_i}$ denotes the quantitative capability when $a_i$ can accomplish some specified function, and $\cos t_j^{a_i}$ denotes the cost $a_i$ expend when accomplish $b_j^{a_i}$.

*2. Coalition and its capability specification*

Coalition is an agent set $C=\{a_1,a_2,\cdots,a_m\}$ which can cooperate and accomplish a task together. The commander of the coalition is the manager responsible for the unification action of the whole coalition, and controls the correlative information, such as the member agent's capability in the whole coalition, etc.

The capability of coalition C is denoted by vector $B_C=(b_1^c,b_2^c,\cdots,b_r^c)$. The capability of coalition C is the sum of every agent's capability in the coalition, $B_c=\sum_{i=1}^{m}B_{a_i}$.

*3.Task and its capability specification*

There are independent tasks $T=\{t_1,t_2,\cdots,t_m\}$in the system, the priority of task t is denoted by set $PRI_t$, the priorities of all tasks in the coalition compose set $PRI_T=\{pri_{t_1},pri_{t_2},\cdots,pri_{t_m}\}$. The capability needed to solve task $t$ is denoted by $B_t=(b_1^t,b_2^t,\cdots,b_r^t)$, and the essential condition of coalition C can take charge of task t is $B_t[i]\le B_C[i], i\in\{1,2,\cdots,r\}$.

*4. Degree of Coalition C Trust to Agent*

The degree of trust denotes the degree of the coalition trust some agent $a_i$, and adjusts the members in the coalition by this mechanism. There are two adjustment rules.

①degree of trust increase: if $a_i$ accomplish the task, then $R_{a_i}=\min(1,R_{a_i}+\delta)$, $\delta\in(0,1)$ calls the cooperate factor.

②degree of trust decrease, there are two situations:

If $a_i$ not accomplish the task, then $R_{a_i}=\max(0,R_{a_i}-\delta)$, $\delta\in(0,1)$ calls the punish factor.

If $a_i$ not take part in the task, then $R_{a_i}=\max(0,R_{a_i}-\lambda)$, $\lambda\in(0,1)$ calls the forget factor.

According to the degree of trust, coalition adjustment rules are defined as follows:

① member remove rule: If $R_{a_i}<R_{Thresh}$, then remove $a_i$ from coalition, $R_{Thresh}$ is the remove threshold.

②member enter rule: If $a_i$ matches the requirement, then add $a_i$ to the coalition, $R_{a_i}=R_{init}$, $R_{init}$ is the initial degree of trust.

### 2.2. Generation and maintenance of task coalition

Task coalition is generated in terms of the tasks arrived at the first time, during the later procedure, the coalition adjusts continuously according to the degree of trust to agent, and comes to stable gradually. The method generating initial task coalition is described as follows:

(1) System initializes, and all agents are independent, there are no coalitions in the system.

(2) The first task T arrives, MA decomposes T to $T=\{t_1,t_2,\cdots,t_m\}$, and gives the priority set $PRI_T$ of T's sub-tasks.

(3) MA sends task bulletins to all idle CAs, gives the invite time limit $Time_T$, and waits for CA to bid.

(4) In terms of the request received, MA chooses the highest priority of $|CA|$ sub-tasks to allocate to CA randomly.

(5) CA sends the bulletin of task $t_j$ to all AAs, and

gives the reply time limit $Time_{t_j}$. AA compares the capability $B_{t_j}$ the task required and the capability $B_{a_i}$ itself owns, and in terms of the coalition strategy, sends the request to CA in the limit time. If AA can accomplish many sub-tasks $T^{'} = \{t_1, t_2, \cdots, t_k\}$, choose CA to send the request according to the following rules:

If $pri_{t_j} = pri_{t_i}, \forall t_i, t_j \in T^{'}$ then send the request to CA obtained the task $t \in T^{'}$ at random

else send the request to CA obtained the task $t_i : pri_{t_i} = \max(PRI_{T^{'}})$

(6) CA assigns the AA's initial degree of trust $R_{init}$, and sends reply to AA to notice it already join the coalition. The coalition generated.

This paper defines two member join strategies, choose according to different application areas:

① the largest capability strategy: if $Count(B_T[i] \le B_a[i]) \ge T\_Count_{thresh}$, then Agent $a$ decides to join;

② average capability strategy: if $\sum_{i=1}^{r} B_a[i] > \eta * \sum_{i=1}^{r} B_T[i]$, then Agent $a$ decides to join.

After the end of every task execution, first adjust the degree of trust of Agent, then adjust the coalition in terms of the degree of trust of the members.

### 2.3. GA-based negotiation algorithm

Task allocation in the coalition is an important problem in the negotiation model. It should minimize the cost of allocation. A GA-Based task allocation algorithm is present to solve the task allocation problem in coalition. Tie algorithm can solve the problem better than traditional algorithms. The following content shows the key issues of the algorithm.

1. CA of Coalition C decomposes the received task $T$ into $T = \{t_1, t_2, \cdots, t_n\}$. CA sends tender invitation to all the AAs in coalition, sets the reply time $Time_{reply}^{T}$.
2. When Agent $a$ receives the invitation, it sends the task set $T_a = \{t_1^a, t_2^a, \cdots, t_m^a\}$ it can execute and the corresponding cost vector $Cost_a$, that is $B_{t_i^a} \le B_L, i \in \{1,2,\cdots,r\}$.
3. CA generates the initial colony by coding all tenders of AA. Coding method:
4. CA generates a stochastic binary coded allocation string according to all $T_a$, called M: $p_1^1, p_2^1, \cdots, p_n^1, p_1^2, p_2^2, \cdots, p_n^2, \cdots, p_1^l, p_2^l, \cdots, p_n^l$, $l = |C| - 1$ (except CA). If Agent $a_k$ can't execute $t_i$, set $p_k^i$ as 0, else set $p_k^i$ as 0 or 1 by stochastic. One task can only be allocated to one agent and every task must be allocated when possible. After the coding step, $Colony(0)$ is generated.
5. Applying the refined GA algorithm that is introduced in the following part to $Colony(0)$. The fitness function is defined as $f(a) = -\sum Cost_T^M$ (that is the whole cost of task execution defined by $M$). The algorithm can find a optimized solution by using self-adaptive aberrance which needs fewer cost.
6. If P is empty, goto 8, else goto 6.
7. Send task allocation solution to AA and wait the AA's confirmation in the given time;
8. End.

An non-optimized GA not only have a slow convergence speed but also easily producing a premature convergence when applied in task allocation. These features have a bad influence of the efficiency and result. We use the optimal initial colony selection, optimal parent crossover, Metropolis rule and multi-point crossover/ mutation to optimize the GA.

The termination condition of the algorithm uses the conjunction of two methods: 1)reach specified iterative times $K$; 2) the fitness reach the following condition: $\exists m \in M, f(m) \ge -\lambda * \sum_{i=1}^{n} \min\{Cost_{a_k} \mid t_i \in T, a_k \in \{a \mid B_{t_i} \le B_a\}\}, \lambda \ge 1$ is a predefined const value according to the problem.

Mutating operation must follow the capability that AA reports to CA. That is if $p_i^k$ mutates, agent $a_i$ must have the capability to execute $t_k$.

The optimized GA is shown in the following table.

Table 1. Refined Task Allocation Algorithm

- Generate $n$ initial members at random, set the selecting threshold $k$, select $k$ members that have the highest fitness from $M$, select $l-k$ members from the left.
- Set iterative number $j=0$;
- $while$ appropriate solution not found
  - $j=j+1$, $n=0$;
  - Put the member of $Colony(j-1)$ which has the highest fitness into $Colony(j)$;
  - $n=n+1$;
  - $while$ $n$ not reach the predefined colony count
    - Select two pairs from $Colony(j)$ at random, select the member that have higher fitness from each pair, called $a,b$;
    - Generate a stochastic number $S \in [0,1]$ of uniform random probability;
    - $if$ $S < Crossprobability$

      Produce $c,d$ by applying multi-point crossover on $a,b$. Applying Metropolis rule on $a,c$ and $b,d$: (suppose $x$ is parent, $y$ is successor)

      (1) if $f(y) > f(x)$, $y \to Colony(j)$;

      (2) if $f(y) \le f(x)$, generate a stochastic number $\xi \in [0,1]$ of uniform random probability, if $\xi < \exp\left[\frac{f(y)-f(x)}{Temperature}\right]$, select $y$, else $x \to Colony(j)$;

      $else$ put $a,b$ into $Colony(j)$;
    - $n=n+2$;
  - Apply multi-point mutation operator on every member in $Colony(j)$, use the same Metropolis rule on the mutated members;
- Recalculate $Temperature$ according to the temperature decent policy, begin the next iteration;
- Return the task allocation solution;

Here we adopt a uniform temperature decent policy, that is $t_{k+1} = \lambda * t_k$, $\lambda \in (0.5,1)$, select a const close to 1 in general.

## 3. Extended contract net based on task coalition

Traditional contract net is reconstructed by using of task coalition, GA-Based task allocation algorithm. Coalition is the basic bid unit. The negotiation process is shown as below(biding and the bid-winning algorithm is described in the author's another paper [9]):

(1) MA receive task TG and stores it in task queue.

(2) When TG is the head of queue, TA check out TG and decompose it into separate sub-tasks: TG=$\{T_1, T_2, \cdots, T_n\}$. Set TA as the system's manager.

(3) If TG is the first task of the system, execute the initial coalition generation according to 3.2. Goto 13;

(4) The manager queries the agent coalition in the blackboard, for each $T_j (j = 1,2\cdots,n)$, send bid inviting message to the active CAs according coalition state and capability information in the blackboard, set the reply time $T_{time-out}$.

(5) According their capability, coalitions that received decides bidding or not. If agent believe that it can execute $T_j$, then calculate the biding value $V(i,T_j) = b_i$, send the biding value to manager. Otherwise, send invite to the free AAs in the system. If the new

coalition can fulfill $T_j$ 's require, send bid information to manager, else tell the manager that it can't finish the task.

(6) If the manager receive no tender of $T_j$ in $T_{time-out}$, goto 12, else select a CA as the winner according to award function $F$, bid value $V^{'}(a_i, T_j)$ and send award to it.

(7) The winner sends confirmation to manager if it plans to execute the task. If the manager doesn't receive the confirmation in $T_{time-out}$, goto 11.

(8) The manager send $T_j$ to the winner, monitor the executing state;

(9) CA invokes the task decomposition module, decomposes $T_j$ to $T_j = \{t_{j1}, t_{j2}, \cdots, t_{jm}\}$, then invokes the task allocation algorithm to allocate the sub-tasks in coalition. When all AAs return result, report the task result to manager. According to the task executing state, regulates the belief value of AA, adjusts the coalition's members;

(10) If the coalition didn't finish all the sub-tasks, the left tasks is called $T_j'$, let $T_j = T_j'$, CA becomes the new manager, goto 11, else goto 13;

(11) The current negotiation round is finished.

(12) The manager decides whether to begin a new bid round. If yes goto 4;

(13) Negotiation process ends.

## 4. Experiments and Analyses

Automatic missile defense system is a typical MAS. It consists of a command center, radar system, several campaign armies. Campaign army consists of several battle units. Each battle units consists of several aerial defense missiles. A prototype of the system is designed to test the performance of our negotiation model.

In the system, MA corresponds to the command center. CA corresponds to a campaign army. AA corresponds to a battle unit. The system includes one MA, six CA and thirty AA. The AA's capability consists of attack type, attack range, attack precision, can be described as a five dimension vector $(type, latitude, longitude, range, precision)$, such as $(air, 100, 85, 100, 0.8)$.

Three models are involved: traditional contract net model (CNM), un-optimized GA-based contract net model (UEM) and optimized GA-based contract net model(EM).

*1. Average time of search optimization*

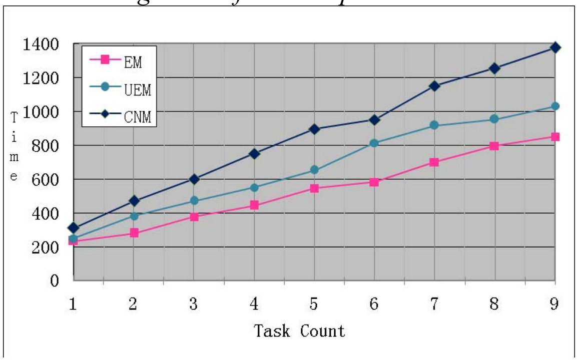


Figure 1. Average-time of search optimization

Each result is the average value of 10 tests. From figure 1, it is obvious that UEM and EM can find the task allocation solution than the CNM. EM overcomes the shortage of slowly convergence than UEM.

*2. Performance Comparison*

We compare the three model from times of finding optimize result and local optimize result. Every model executes 100 times. Task count of each experiment is random.

Table 2. Comparisons on performance of the three models

| | CNM | UEM | EM |
|---|---|---|---|
| Optimal solution | 52 | 70 | 81 |
| Local optimal solution | 48 | 30 | 19 |

From the result, we can see that EA has a stronger search capability than UEA and CNM. EA overcomes the shortage of getting into local optimization. It has a higher probability of finding global optimal solution than UEA.

*3. Comparison of communication cost*

In MAS, communication is an important part of negotiation. It has a great influence on the system efficiency. We compare the communication times of EM and CNM (UEM equals EM). EM already executed several tasks and the coalition in it relatively stabled.

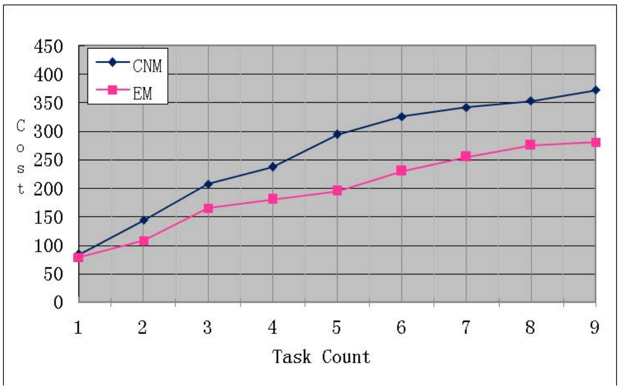


Figure 2. Comparison of communication cost

From the figure, EA needs few communication times than CNM. The system communication load is reduced.

## 5. Conclusion

A task coalition and GA-based extended contract net negotiation model is presented in the paper. We give the definition of task coalition. Coalition generation and maintenance policy is given upon the definition. A special GA is proposed, which is optimized by using of optimize initial colony selection, optimize parent crossover, metropolis rule and multi-point crossover/mutation, to solve the task allocation problem in coalition. The traditional contract net model is refined according to task coalition an optimized GA. The experiments and analyses of missile defense system prove that the model effectively decreases the negotiation cost, increase the efficiency and performance.

There are some shortages to be solved: 1) Task is stochastically allocated at the first time. This mechanism results in the coalition structure rely on the first task. In some circumstance, it has a bad influence on the stability of coalition, increases the additional cost; 2) when the task's property varies greatly, the coalition's stability needs more research; 3) the league policy may result in the imbalance resource allocation of the system, this factor should to be considered in the future research.

## Acknowledgements

The research work is supported by National 863 Hi-Tech Project (2003AA118030), National Defense Foundation (41315.8.1, 513150804) and Science and Technology Fund for Returnee of Heilongjiang Province (F2004-16).

## References

[1] R. G. Smith. The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver. In IEEE Transactions on Computers, Vol C-29, No.12, pp.1104~1113, December 1980.

[2] Sandholm T W.Lesser V R. Coalition among computationally bounded Agents. Artificial Intelligence. Vol 94, No.1, 9~137, Jan 1997.

[3] T. Sandholm, V. R. Lesser. Advantages of a Leveled Commitment Contracting Protocol. AAAI 1996, pp.126~133, July 1996.

[4] Chen Xueguang, Song Haigang. Further Extensions of FIPA Contract Net Protocol: Threshold plus DoA. Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, pp.1182~1188, Jan 2004.

[5] Klaus Fischer, Jörg P. Müller, Markus Pischel, Darius Schier. A Model for Cooperative Transporation Scheduling. Proceedings of the First International Conference on Multiagent Systems, San Francisco, California, pp.451~463, March 1995.

[6] Bouzouia, B.; Bouchemma, R.. Performance Enhancement of a Contract Net Protocol Based System Through Instance-Based Learning. ICRA 2004, Barcelona, Spain, Vol 1, pp.523~528. May, 2004.

[7] FIPA. FIPA Agent Management Specification. FIPA Experimental Component specification number 23, 2002.

[8] Chao Deng, Maozu Guo, Yadong Wang. Research on knowledge representation and inference based on extended weighted fuzzy logic, Proceeding of ISIITA2003 conference, Beijing, pp. 93-96, October 2003.

[9] Haijun Tao, Yadong Wang, Maozu Guo and Hanlun Wang. A Multi-Agent Negotiation Model Based on Acquaintance Model and Extended Contract Net Protocol. Journal of Computer Research and Development, 43(7), 1155~1161, July 2006.