http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 733-738, 2007
ISSN 1812-5638

Extended Contract Net Protocol for Multi-Robot Dynamic Task Allocation

Zhou Pu-Cheng, Han Yu-Sheng and Xue Mo-Gen
Department of Information Engineering, Artillery Academy of PLA, Hefei 230031,
AnHui Province, People's Republic of China

Abstract: One of most important aspects in the design of multi-robot systems is the allocation of tasks among the robots in a productive and efficient manner. This study addressed a kind of multi-robot distributed task allocation method under dynamic environments. The proposed approach is based on traditional contract net protocol and has been extended from three aspects, including task announcement strategy with case-based reasoning, selection of the contractor using multi-attribute decision theory and breach of contract mechanism. Simulation results of the pursuit problem have shown the feasibility and validity of the given algorithm.

Key words: Task allocation, contract net protocol, multi-robot system, cooperation and coordination

INTRODUCTION

One of the important recent trends in robotics is the study of teams of Multi-Robot Systems (MRS). The general idea is that teams of robots, deployed to achieve a common goal, are not only able to perform tasks that a single robot is unable to, but also can exceed systems of individual robot, in terms of efficiency and quality. As a result of the focus on MRS, Multi-Robot Task Allocation (MRTA) has been one of the key research issues for MRS. Generally speaking, the task allocation problem is that, given some resources, some tasks and a performance metric, the goal of the system is to maximize performance on the tasks, subject to the resource constraints.

Current task allocation methods can be roughly divided into three classes (1) centralized allocation (Mary *et al*., 2005), that is, some robot or centralized computer acts as the manager to assign given tasks to relevant robots (2) emergent allocation (Low *et al*., 2006), any robot among the system can be the manager. Besides, although the manager makes decision just by its local sensation and inner information, some task-allocated state emerges from the whole system and (3) distributed allocation, that is, according to some social laws and management rules, task allocation is performed by interaction of robots. Distributed task allocation methods include auction (Nanjanath and Gini, 2006), free market (Zlot and Stentz, 2006) and Contract Net Protocol (CNP) (Lemaire *et al*., 2004) and so on. Since CNP is the best-known scheme for task allocation, this study focuses on CNP-based MRTA.

CNP AND ITS LIMITS

CNP was firstly proposed by Smith (1980). According to this approach, each robot takes on one of two roles: Manager or contractor. A manager is responsible for monitoring the execution of a task and processing the results of its execution and a contractor is responsible for the actual execution of the task. CNP can be described as unfolding in five steps (1) request for bids: a manager requests a bidder to submit a bid (2) submission of bids: the bidder prepares a bid and submits it to the manager (3) awarding of contracts: the manager evaluates the bid, which could (or not) be awarded as a contract to the bidder (4) acceptance of contracts: if awarded, the bidder is requested to accept (or decline) and (5) the execution of the contract.

As a widely used task allocation method, CNP is one of the most popular research frameworks in distributed problem solving field. In spite of this, some limits still exists when apply CNP to specific problems. Here we sum up three limits for traditional CNP: 1) negotiation based on CNP among multiple robots needs much communication. During the task announcement phase, the manager will broadcast task announcement message to all robots. Any robot that satisfies the bid specification can join for bid. It is communication intensive due to the broadcast of task announcement. It runs into problems as the problem size, the number of communicating robots and the number of tasks announced by them increases. This limits its usability in large-scale MRS; 2) when the manager makes decision to pick up which robot as the contractor, usually

Corresponding Author: Zhou Pu-Cheng, Department of Information Engineering, Artillery Academy of PLA, Hefei 230031, AnHui Province, People's Republic of China

only by its tender price. It neglects some other attributes, such as the credit of each robot, the task load of each robot, the relevance of different tasks and so on (3) once a task is assigned to some robot that wins the bid, the robot must perform the task until it has finished the task. Since it doesn't take the dynamic change of the task into account, it's hard to optimize the task allocation solution.

EXTENDED CONTRACT NETPROTOCOL

To deal with these problems mentioned we extend traditional CNP from the following aspects.

CNP with CBR: To reduce communication load on task negotiation with CNP, Ohko *et al*. (1997) given a method called addressee learning, which combined CNP with Case-Based Reasoning (CBR). Base on Ohko's work, we propose a CBR-based task announcement strategy for CNP, which let the manager gradually learn from the bid process using CBR, so as to shrink the scope of inviting public bidding.

Each case of the Case Base (CB) is denoted as an array. To record successful and failed bid activities, we introduce two kinds of CB: recommendatory case base (RCB) and Prohibitive Case Base (PCB). RCB is denoted as: RCB = $\{C_1, \ldots, C_n\}$ and each case of RCB is denoted as: c_RCB: = <caseID, task, ctor, C>. Where caseID is ID number of case, task = $\{A_1V_1, \ldots, A_mV_m\}$ is description of the bid task, A_k is the k-th attribute, V_k is value of A_k; ctor is contractor of the task; C is an effect coefficient which is updated as:

$$C \leftarrow \begin{cases} kC, \text{fail to finish the task} \\ 1, \text{succeed to finish the task} \\ C, \text{others} \end{cases} \tag{1}$$

Where, $k\in(0, 1)$ is decay coefficient. If C is less than a threshold, corresponding case will be deleted from RCB.

PCB is denoted as PCB = $\{S_1, \ldots, S_k\}$, each case of PCB is expressed as: c_PCB: = <caseID, task, rer, R>. Where rer is the robot that refuses to bid for the task, R is prohibitive coefficient which is updated as:

$$R \leftarrow \begin{cases} 1, \quad \text{refuse to bid for the task} \\ eR, \text{others} \end{cases} \tag{2}$$

Where, $e\in(0, 1)$ is decay coefficient. If R is less than a threshold, corresponding case will be deleted from PCB.

When a task T_j appears, the manager calculates the similarity of T_j with tasks which have been finished by robots within CB. Then, the manager can choose some robots to bid for task T_j by the suitability of each robot for task T_j. Detailed procedures are followed:

- Calculate the distance between T_j and T_i under each task attribute:

$$D(A_{jk},A_{ik}) = \frac{1}{1+|V_{jk}-V_{ik}|} \tag{3}$$

Where, V_{jk} and V_{ik} are values of the k-th task attribute of task T_j and T_i, respectively.

- Evaluate the similarity of T_j and T_i

$$\begin{cases} Sim(T_j,T_i) = \sum_{k=1}^{N} w_k D(A_{jk},A_{ik}) \\ \sum_{k=1}^{N} w_k = 1, \quad w_k \ge 0, k = 1,\cdots,N \end{cases} \tag{4}$$

Where, w_k is the weight of the k-th task attribute.

- Establish the similar task set of task T_j:

$$S(T_j) = \{T_i | Sim(T_j,T_i) \ge \Delta\} \tag{5}$$

Where, T_i is the task within the case base, $\Delta\in(0, 1)$ is a threshold of task similarity.

- Evaluate the suitability of each robot for T_j:

$$S(Robot_i,T_j) = \frac{1}{|S(T_j)|} \cdot \sum_{T\in S(T_j)} E(Robot_i,T) \tag{6}$$

Where, $|S(T_j)|$ is the number of the similar task set of T_j, $E(Robot_i, T)$ is the suitability when $Robot_i$ undertakes task T and it is evaluated by:

$$E(Robot_i, T) = C\,(1-R) \tag{7}$$

- Select suitable robots to bid for T_j

According to Eq. 8, the manager evaluates each robot and selects suitable robots to bid for T_j:

$$Candidate = \{Robot_i | S(Robot_i, T_j) \ge \theta\} \tag{8}$$

Where, $\theta\in[0,1]$ is a threshold.

Decision-making for selecting contractor: While making decisions, if the manager considers synthetically some factors such as credits of robots, the relevancy of different tasks and so on, then it's possible to improve the task allocation solution.

Definition 1: The Credibility Value (CV) depicts the degree that the manager believes a robot should complete the given task. After each decision, the manager updates CV of the contractor: if it completes the given task, then its CV will increase. Otherwise, it will decrease. We use $Credit_{ij} \in [0, 1]$ to denote CV of the robot i to task T_j and the update rule of CV is:

$$Credit_{ij} = \begin{cases} \min(\frac{C_a}{C_t} + \xi_1, 1), & \text{success} \\ \max(1 - \frac{C_q}{C_t} - \xi_2, 0.1), & \text{failure} \end{cases} \tag{9}$$

Where, C_a is the total number of robot i undertakes T_j, C_q is the number that i failed to finish T_j and C_t is the total task numbers of i, ξ_1 and ξ_2 are thresholds.

Definition 2: The relevancy value (RV), denoted as Con_{ij}, means the relevant degree between task T_j and tasks that have been done by robot i. Here the update rule of RV is defined as:

$$Con_{ij} = \begin{cases} 1 - \frac{\Delta t}{T_V}(1 - \frac{C_s}{C_t}) & \Delta t \le T_V \\ \frac{C_s}{C_t} & \Delta t > T_V \end{cases} \tag{10}$$

Where, $\Delta t = t - t_0$, t is current time (decision point), t_0 is the latest time spending for completion of T_j, T_V is the deadline of T_j be finished and C_s is the total numbers that robot i succeeds to complete task T_j.

Base on CV, RV and some other factors, such as task load of each robot, the cost when execute given task and tender price et al., the manager can make decision to choose a suitable robot to be the contractor of the given task. Since the decision of the manager can be regarded as a kind of multi-attribute decision-making problem, we can choose TOPSIS (Hwang and Yoon, 1981) to resolve. Suppose that candidate set of the contractor of the given task is B = {$bidder_1$, ..., $bidder_m$} and attribute vector to evaluate B is Y = {y_1, ..., y_n}. Then, n attribute values of any $bidder_i \in B$ (i = 1...m) forms a vector Y_i = {y_{i1}, ..., y_{in}}. Apparently, n attribute values of the m candidates form a $m \times n$ matrix, we call it Support Decision-Making Matrix (SDMM). Based on SDMM, the manager can obtain the order of all candidates using TOPSIS and picks up the most suitable one to be the contractor of the given task.

Breach of contract and management: For traditional CNP, once a contract is awarded, it won't be terminated until the task is finished, we call this Doesn't Permit Breach of Contract (DPBC) mechanism. But during the processing of the task, in some cases the contractor maybe can't finish the task even after a long time. If the contractor continues to execute the task, it will lose more, also this will make the task deadlocked. So, we propose a kind of Permit Breach of Contract (PBC) mechanism.

According to PBC mechanism, once the contractor finds that it can obtain more interests if it performs other tasks, then it will choose to abort current contract. To ensure the stability of the system, we introduce the concept, fine for breach of contract (FBC), which means that the contractor who wants to breach the contract must pay penalty to the task manager. So if a contractor thinks that it will get more interests when performs another task, even it is fined because of its breach of current contract, in this case it will choose to breach current contract; otherwise, it will continue to implement the current task. FBC is evaluated as:

$$F(T_j) = \begin{cases} k_1 S_{Ej}(1 - e^{-k_2 t^2}) + e & t \le E_T \\ k_1 S_{Rj} & t > E_T \end{cases} \tag{11}$$

Here, $k_1 \ge 0$ is the ratio of fines, $k_2 > 0$ is an adjustment coefficient, t is the passed time after the contract becomes effective, $\varepsilon \ge 0$ is the lowest fines, E_T is the expected finished time of task T_j.

Definition 3: Lifecycle of Contract (LC), denoted as C_{life}, is a measurement of confidence that the contractor can complete the task specified by the current contract. C_{life} is evaluated as:

$$C_{life} = \begin{cases} e^{-k_3 t^2} & t \le T_{max} \\ 0 & t > T_{max} \end{cases} \tag{12}$$

Here, T_{max} is deadline of task T_j that should be finished and $k_3 > 0$ is an adjustment factor.

According to Eq. 12, when the contractor begin to fulfill the contract, $C_{life} = 1$. With the process of the task, C_{life} decreases gradually. When the task cannot be finished after the deadline, C_{life} decreases to zero and then the contractor begins to decide whether to breach current contract or not.

Task allocation algorithm based on ECNP: Based on extension of CNP (ECNP) discussed above, we propose a kind of multi-robot dynamic task allocation algorithm, shown as follows:

- When task T_j appears, the manager decides the scope of inviting public bidding according to historical bid activities of CB and sends Task Announcement Message (TAM) to relevant robots.
- When a robot receives a TAM, it evaluates reward, cost, difficulty and some others aspects of the task specified by the TAM and choose an appropriate task to bid for, then submit a bid to the manager.
- After deadline, the manager evaluates bids, picks up a suitable contractor using TOPSIS and then sends Awarding of Contract Message (ACM) to it.
- When a robot receives a ACM, it sends confirm message to the manager, acts as the contractor and then starts to perform the contract.
- If the contract is finished, goto step 10.
- If $C_{life} = 0$, goto step 11.
- If the contractor wants to breach current contract, then go to Step 8. Otherwise, goto step 9.
- Judge whether breach of current contract or not. If not, go to Step 5; else, go to step 8.
- Perform the contract, go back to step 5.
- The contractor obtains some reward. The manager updates CBs and parameters of SDMM.
- The contract is aborted. The manager updates CBs and parameters of SDMM.
- The contract is aborted. The contractor pays some fines to the manager. The manager updates CBs and parameters of SDMM.

The algorithm mentioned above can be divided into two phases, from step 1 to 4 is the first phase: task allocation based on ECNP and from step 5 to 12 is the second phase: task performance and management.

SIMULATIONS

To testify the proposed task allocation algorithm, we use the pursuit problem as a test-bed.

The pursuit problem: A variant of the pursuit problem is considered. In a 25×30 grid world, m robots (hunter) want to capture n robots (prey). Both hunters and preys are divided into four types: I, II, III and IV. Type of a hunter indicates its capture ability and type of a prey indicates required types of hunters that to capture it. The hunter can only capture preys whose type isn't more than the hunter's, e.g., a hunter whose type is III can capture type I, II or III preys, but cannot capture type IV preys. The value of each prey indicates rewards will be paid to the relevant hunters after the prey is captured. Both hunters and preys have global field of view, the same speed (one grid per move) and ideal communication (no loss or delay of communication). Each hunter can only bid one pursuit task. A prey is captured when the distance of the prey and its responding hunter less than two grids. If prey E_j is captured, corresponding hunter will obtain rewards with the value is the cube of the type of E_j, i.e., Score = $(R_{Ej})^3$, where R_{Ej} is the type of E_j. At time t, preys move randomly, but hunters use greedy pursuit strategy to capture their target prey, that is:

$$x_{Pi}(t+1) = \underset{x_{Pi}(t+1)\in Cm_{Pi}(t)}{\operatorname{argmin}} D(x_{Pi}(t+1), x_{Ej}(t)) \qquad (13)$$

Where, $x_{Pi}(t)$ Position of hunter. P_i, $x_{Ej}(t)$ Position of prey E_j. $Cm_{Pi}(t)$ Accessible grids set of P_i in one step at time t. $D(.,.)$ The distance of P_i and E_j.

Experimental results and analysis: To evaluate the proposed task allocation algorithm, experiments from four aspects have been done.

COMMUNICATION LOAD: CNP VS. ECNP

To compare communication of CNP with ECNP, we adopt Utility of Communication (UC) as a criterion. Here UC is defined as scores of hunters divided by total numbers of messages during task negotiation.

Figure 1 is average result for 4 hunters to capture different numbers of preys in 30 experiments, where hunters include all the four types, each type has one hunter and the numbers of the four types of evaders are 5, 10, 15, 20 and 25, respectively. From Fig. 1 we can see that CU of ECNP is more than that of CNP. Namely, when obtain the same scores, ECNP need less communication than that of CNP, so it reduces communication load of the task negotiation.

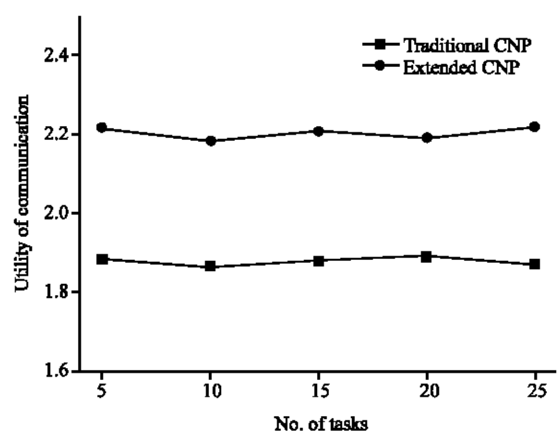

Fig. 1: Comparison of CU for CNP and ECNP

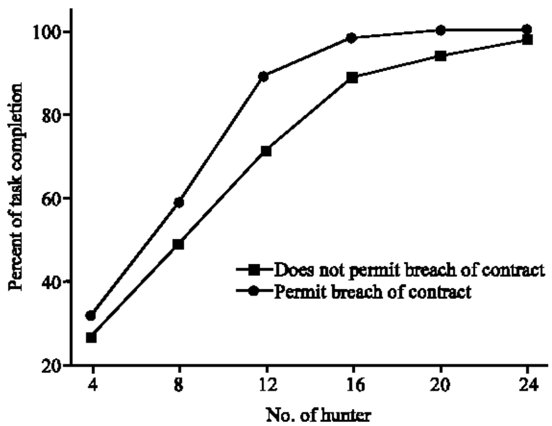

Fig. 2: Performance comparison of PBC mechanism with DPBC mechanism

PERFORMANCE COMPARISON: PBC VS. DPBC

To compare the performance of PBC mechanism with DPBC mechanism, another 30 experiments have been done. The maximum simulation cycle is 500 steps for each experiment. For PBC, main simulation parameters are: $k_1 = 1$, $k_2 = k_3 = 1$, $\varepsilon = 0.2$* Scores of the prey. The Task Completion Ratio (TCR) is chose as a criterion, i.e., when numbers of preys are fixed, the proportion of task numbers been finished to total task numbers for all hunters during a certain period of time. Figure 2 is the result of TCR for different number of hunters to capture 16 preys (4 preys per type). From Fig. 2 we can find that, compared with DPBC mechanism, PBC mechanism can increase the whole performance of the hunters group.

PBC MECHANISM: EFFECT OF DIFFERENT RATIO OF FINES

To test PBC mechanism under different ratio of fines, simulation experiments under different ratio of fines (let k_1 from 0 to 10, with increment of 0.5) were done, other parameters are the same as mentioned earlier. Results Fig. 3 show that when $k_1 = 0$, means a contractor can breach its current contract freely, then performance of hunters is dissatisfactory. Best performance of the hunters achieved when $k_1 = 1$, with average reward of hunters is up to 144.8. After that, with the increase of k_1, rewards decrease gradually. When $k_1 = 10$, average rewards is near to the result of DPBC mechanism.

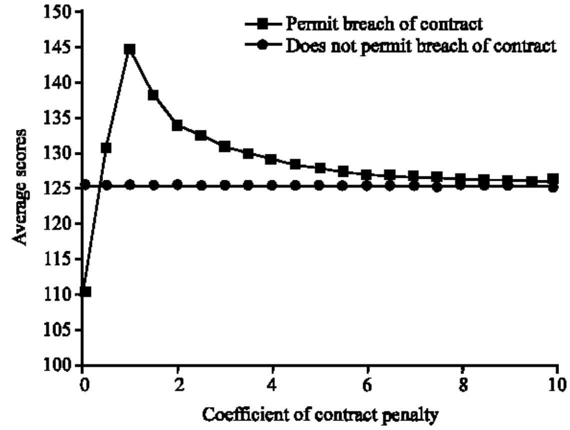

Fig. 3: Results of different ratio of fines

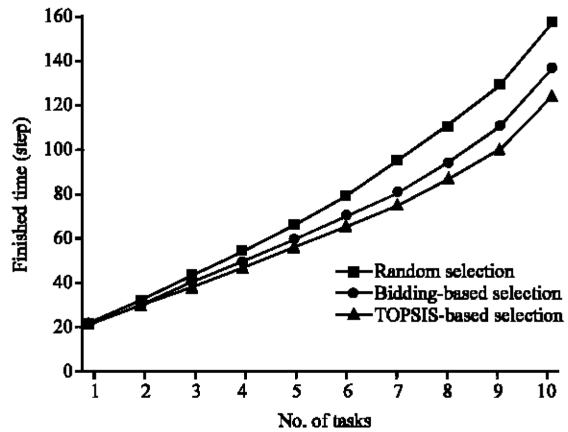

Fig. 4: Results of different decision-making methods

COMPARISON OF METHODS FOR SELECTING CONTRACTORS

Finally, to test the performance of methods for selecting contractors, three typical approaches are chose: pick up the contractor by random (Random selection), pick up the contractor by tender price (Bidding-based selection) and pick up the contractor by TOPSIS method (TOPSIS-based selection). The number of hunters is 4 and one type per hunter. Types of preys are randomly produced by the computer and the number of them is increased from one to ten. The total tasks finished time is selected as a criterion.

From the Fig. 4 we can find that the solution obtained by TOPSIS has best performance, which shows that if the manager can take more factors relevant to the given task into account, then it will be helpful to improve the efficiency of the task.

CONCLUSIONS

In this study, a kind of multi-robot dynamic task allocation method is addressed. The proposed method is based on the extension of contract net protocol from three aspects: using case-based reasoning to reduce the scope of inviting public bidding, adoption of multi-attributed decision theory to support the task manager to pick up the contractor and introducing breach of contract and management mechanism to optimize the task allocation solution. Simulation results on the pursuit problem have shown the feasibility and validity of the given algorithm.

REFERENCES

Hwang, C.L. and K. Yoon, 1981. Multiple Attribute Decision Making: Methods and Applications. New York: Springer-Verlag.

Lemaire, L., A. Rachid and L. Simon, 2004. A distributed tasks allocation scheme in multi-UAV context. Proceeding of IEEE ICRA., pp: 3622-3627.

Low, K.H., W.K. Leow and J. Anq, 2006. Autonomic mobile sensor network with self-coordinated task allocation and execution. IEEE Trans. SMC Part C., 36: 315-327.

Mary, K., N. Illah and S. Katia, 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. Proceedings of AAAI., pp: 1292-1297.

Nanjanath, M. and M. Gini, 2006. Dynamic task allocation for robots via auctions. Proceeding of IEEE ICRA., pp: 2781-2786.

Ohko, T., K. Hiraki and Y. Anzai, 1997. Addressee Learning and Message Interception for Communication Load Reduction in Multi-robot Environments. Distributed Artificial Intelligence Meets Machine Learning. Springer-Verlag, Berlin, pp: 242-258.

Smith, R.G., 1980. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Trans. Comput., 29: 1104-1113.

Zlot, R. and A. Stentz, 2006. Market-based multirobot coordination for complex tasks. Int. J. Robotics Res., 25: 73-101.