Distributed Task Allocation in Dynamic Multi-Agent System

Vaishnavi Singhal
Department of CSE & IT
Jaypee University of Information and Technology
Waknaghat, H.P., India
singhal.vaishnavi92@gmail.com

Deepak Dahiya
Department of CSE & IT
Jaypee University of Information and Technology
Waknaghat, H.P., India
deepak.dahiya@juit.ac.in

Abstract— **from the last two decades, Software agents are playing an important role in the field of Artificial intelligence and the Distributed Problem Solving. The properties of software agents like autonomy, reactivity, pro-activity and their social ability make them more center of focus for the real world problems. The accomplishment of any complex task is required to be done by the agents autonomously without any user intervention in order to achieve high reliability and adaptability. This paper mainly concentrates on the task allocation problem in multi-agent systems. Task Allocation is an important and challenging problem. This can be defined as the problem of allocating tasks among agents within a multi-agent system. Main objective of the task allocation problem is to maximize the number of successfully completed task and overall system utility without any conflict. To accomplish any complex task, agents negotiate, cooperate and coordinate with each other. Many researchers are working in the field of task allocation in multi-agent systems. In this paper, various approaches of task allocation in multi-agent systems are discussed. The comparison of these approaches is also providing that lead to a discussion and motivation of the task allocation problem for multi-agent systems. This paper presents a hybrid approach for task allocation in dynamic multi-agent systems. The conclusion drawn from this survey is that, for dynamic multi-agent systems, the distributed task allocation is a better approach.**

Keywords— Agent coordination, communication and negotiation, Artificial Intelligence, Distributed Problem Solving, Multi-Agent system, System utility, Task allocation.

I. Introduction

To accomplish any task, it must have to be allocated to the agents forming multi-agent system. Task allocation is an important and challenging problem in Multi-agent systems. The task allocation problem can be defined as a problem of assigning a set of tasks to a set of agents in order to achieve the maximum number of successfully accomplished tasks where both the agents and the task can leave and enter into the system over the time. There can be more tasks than agents thus agents need to schedule themselves to attempt each task in turn [10]. In case of heterogeneous system, where each agent may have different capabilities and unable to perform the task individually, agents need to communicate and negotiate with other agents. The group of agents that collectively fulfill the task requirements is termed as coalition. Thus the initiator agents need to choose the best coalition which will accomplish the task more efficiently and effectively.

Task allocation can be done in two ways [1]:

- Centralized task allocation
- Distributed task allocation

In centralized approach, a central planner is used to allocate the tasks to cooperative agents. Here, single point of failure is usually inevitable which results in decreasing robustness of the system.

In Distributed approach, the task can be arrived at any agent and the agents communicate amongst themselves to complete that task assignment.

Example of task allocation problems include the allocation of sensing tasks to robots [2] and rescue tasks to ambulances [3]. For the dynamic multi-agent systems, the task allocation can be done in both centralized and distributed manner [2], [3], [4], [5], and [6]. In [2], [3] the task allocation had been done in centralized way, where they didn't consider the fact that agent or tasks may change over the time. The disadvantage of this approach is that if the task allocation problem gets changed due to the arrival of a new agent or task, it needs to be re-allocation of the agents from the scratch. A number of distributed task allocation algorithm for the dynamic multi-agent system have been developed: namely, OPGA [4] based on markov game theory, Auction and market based approaches [5], [6], DCOPs solution based approaches like LADCOP [7], SDPOP [8], distributed anytime algorithm [9] based on FMS, negotiation based approaches which include constraints optimization like CFSTP [10], and swarm intelligence based approach [11].

The objectives of each of the researchers were to find the solution of task allocation problem which gives maximum system utility and the successful accomplishment of task. System utility is the reward which is gained by every system after the successful accomplishment of the task [1]. The task is said to be accomplished successfully, if all the resources required by that task is available, the allocated agents are available and free to perform the task and the task is

ISBN:978-1-4799-8890-7/15/$31.00

completed within its deadline without any conflict.

The rest of the paper is structured as follows: section II contains the literature survey of the various approaches of task allocation in multi-agent systems, section III includes the critical review and motivation, section IV includes problem definition, section V includes proposed work and methodology and at last the section VI concludes this paper.

II. RELATED STUDY

In recent years, many centralized and decentralized algorithms have been proposed for task allocation in cooperative multi-agent environments. The problem of task allocation and its relationship with overall system performance is a major research issue in distributed multi-agent system. Many researchers gave various approaches for task allocation in multi-agent systems [2], [3], [4], [5], [6], [7], [8], [9], [10], [11].

Many researchers gave various approaches for finding the optimal task allocation in multi-agent systems. These are:

- Game-theory based approach
- Allocation based on markov decisions
- Auction based task allocation
- Negotiation based approach
- Distributed constraint optimization
- Swarm intelligence based approach

In next section we will discuss these approaches in detail.

A. Game-Theory based approach

Here, each agent will be treated as a player and the process of allocation task to the coalition is strategy. The goal is to find the best strategy in the nash-equilibrium condition. For each player, the aim is to choose the strategy which will give its best payoff [1]. When each agent will choose its best strategy, no one will wish to deviate from their current strategy because they can't do any better than that. This is called nash-equilibrium condition.

In [4], Chapman defines a game-theoretic technique for decentralized planning to address dynamic task allocation named as OPGA. They formulate the task allocation problem as Markov game. But due to this formulation the agent's utility function became difficult to derive. Thus they approximate the global utility using a series of static potential game and derive the agent's utility function. The global utility is the payoff gained by the whole system after performing the task successfully whereas the agent's utility is the individual payoff gained by the agents participated in the task accomplishment. They also used the Distributed Stochastic Algorithm to find equilibrium in these games. Implementation has been done on RoboCup Rescue simulator. The result shows that this approach outperformed the centralized task scheduler and is robust to restrictions on the agents' communication and observation range. But this algorithm requires the continuous negotiation and doesn't consider the environmental changes.

B. Allocation based on Markov Decisions

The agents take the decisions on the basis of markov theory. Given the current state at particular time instant, the agent must have to take the action which result in optimal next state. For the markov game approach, agent must have either global or the partial view of the system.

There are various researchers which discussed the Agent-based system by using Markov Decisions Processes.

In [14] the author presented a system designed for task allocation, staff management and decision support for scalable systems. The task is allocated to workers according to the user's requirements, different goals of the management, permanent staff and contractors. The system is designed on the basis of Contract Net protocol, belief theory and Markov Decision Processes

C. Auction Based Task Allocation

The task allocation can also be done on the basis of auction based market theory. The centralized task allocation is done by this approach. There is a central auctioneer that is responsible for the task handling and allocation. On task arrival, the central auctioneer starts auction for its accomplishment. Agents those are interested to perform that task sends their contribution to the central auctioneer. Then central auctioneer make the decision which maximizes the overall system utility. The winning agent or group of agents will be chosen for the task accomplishment and the task is allocated to them.

In [15], the market-based allocation of the heterogeneous tasks to the heterogeneous agents was discussed. The authors have presented a heterogeneous task model and the metric task coverage for generating good heterogeneous teams. They used the sequential auction with the TeamFit bidding mechanism.

D. Negotiation Based Approach

The agents negotiate with the other agents via some communication link for the efficient task allocation. The initiator agent if not capable to accomplish the received task individually then it negotiates with other agents in the system. Agents via negotiation form the coalition and then the coalition which maximizes the system utility has been chosen for that task accomplishment.

O. Shehory and S. Kraus presented an anytime algorithm in [3] for task allocation among computational agents via coalition formation. Here, the agent contacts with their neighbor agents and make some agreement on the required capability for the task. The coalition is formed i.e. the group of agents are collectively coordinate for the task accomplishment. The best coalition is chosen among disjoint and overlapping coalition. Disjoint coalition is the group of agents where no agent is part of more than one group whereas in case of overlapping agents, one agent can participate in more than one agent. They also considered the task precedence ordering and allocate the task only when all its' predecessor tasks have assigned some coalition. This approach was implemented on RETSINA. The actual performance was 0.9 time the optimal performance. In worst case, the actual performance declined fast to less than 0.5 times of optimal performance.

In [13], the author constrained the agents' cooperation domain within a community i.e. the agent can only negotiate with its intercommunity member agents. This approach was inspired by the social sites like twitter or Facebook. They presented their approach in three phases. First, task selection where the desirable task is to be selected preferentially. Second, allocation to community i.e. allocating the selected task to community based on significant task-first heuristics. Third, allocation to agents where the negotiation of resources for the selected task is done based on the non-overlap agent first and breadth first resource negotiation mechanism. In this community-aware model, because of dense intra-community connections, it is easy for a community member to cooperate, which will produce less system communication cost compared to the global-aware task allocation model. Because of the lower time complexity of the proposed heuristic algorithm, the community model can be exploited well in large-scale applications. In this paper, the community was fixed during the task allocation however in reality the communities can be dynamic.

E. DCOPs based Approach

DCOP stands for Distributed Constraint Optimization problems. In DCOPs problem, each agent is given with a variable which has some assigned value whose domain is the action that an agent can perform. The objective function is to optimize some global constraint. From the literature surveyed there are various constraints that can be used in dynamic multi-agent systems. Like spatial constraint, temporal constraint, Communicational constraint etc. there are various DCOPs approaches like max-sum, Fast-Max-Sum, ADOPT, La-DCOP etc.

A new Algorithm, Fast-Max-Sum (FMS) was proposed in [12]. The FMS algorithm is an extension of max-sum algorithm. It defines new function on variable and factor nodes. This reduces the number of states over which each factor has to compute its solution. Furthermore, the FMS algorithm allows each variable to decide when to send messages to other connected factor, when the factor-graph changes.

The author has further extended the FMS algorithm by applying online domain pruning and branch-and-bound methods as a novel approach in [9]. This novel approach achieved 23% more utility, 31% less time and 25% less messages than other existing approaches in dynamic environment.

In [10], Ramchurn et. al. builds the case for coalition formation with spatial and temporal constraint. They gave the MIP formulation for various constraints like completion constraint, deadline constraint, starting time, routing and service constraint etc. they also devised a new anytime heuristic for task allocation. They defined the set of feasible assignments and choose the best allocation which can accomplish the task is less time and can participate in more number of future task and allocate the task to such coalition. CFTSP completes 97% tasks for the larger problems having 20 agents and 200 tasks.

F. Swarm Intelligence based Approach

Swarm Intelligence has become a new field in the AI research, which is inspired by the social insect behavior that displays intelligence on the swarm level with simple interacting individuals. The swarm intelligence can be used for the task allocation in multi-agent system. In [11], the author presented the swarm based approach of task allocation. They implemented ant allocation algorithm for task allocation in random dynamic environment and perform task re-allocation when working condition changes. The author used hybridization of two approaches. For task selection, Honeybee model was used and then ant colony optimization is used. First of all, randomly initialize each agent with some response threshold. When task arrives at the system, the probability of selecting a task by the agent is calculated on the basis of response threshold. If Less response threshold then greater will be the chance of selecting that task. After finishing the task, the response threshold is updated similar to ant colony optimization.

Table 1- Survey of various approaches

S. No	Paper Title	Approach	Advantages/ Disadvantages
1	Decentralized Dynamic Task allocation: A practical Game theory approach, AAMAS, 2009 [4]	Overlapping Potential Game Algorithm	-Decentralized task allocation -consider future effect of agent's current action -Continuous negotiation -static environment.
2	Adaptive Task Allocation in multi-agent systems ACM, New York, 2001 [5]	Computational Market system	-Dynamic env. -Heterogeneous agents -Fair allocation - Considers the type, deadline & priority of tasks -Centralized approach -Communication & resource manager overhead

3	A Distributed Anytime Algorithm for Dynamic Task Allocation in MAS AAAI, 2011 [9]	Fast-Max-Sum approach	-Dynamic env. -Heterogeneous agents -Less communication and computation overhead -doesn't consider task preference -doesn't consider impact of future task
4	Coalition Formation with Spatial and Temporal Constraints AAMAS, 2010 [10]	Mixed Integer Programming	-include spatial & temporal constraints -consider future task -minimize comp. time of task and working time of agents -homogeneous agents -one coalition can perform only one task at a time -static env.
5	Task Allocation in Multi-Agent Systems with Swarm Intelligence of Social Insects (ICNC-2010) [11]	Hybridization of Honeybee Selection model & Ant colony optimization	-Random working env. -diff cost for each type of tasks -doesn't consider global maxima -time consuming approach for task completion
6	Community-Aware Task Allocation for Social Networked Multiagent Systems IEEE Transactions , 2014 [13]	Social Networked Multi-Agent Systems	-consider community constraint -significant-task first, non-overlap agent first and breadth-first heuristic is utilized -reduce comm. cost -cooperative agents -centralized Algorithm -static environment

III. Critical Review

Various approaches for the task allocation in multi-agent systems have been studied. There are some benefits as well as shortcomings of these approaches. These approaches have been critically reviewed and some conclusion has been drawn.

The game theoretic approaches outperformed the static applications rather than dynamic applications. The computational complexity in these approaches is also very high. Robustness, scalability and adaptability are difficult to achieve in game-theory approach [1]. Thus it is difficult to use game-based approaches for real-world problems those are dynamic in nature.

The auction based approach depends on the communication link used for the negotiation between the auctioneer and the other agents. It leads to slow decision-making in case of unreliable communication line [1]. It also results in more computation and communication cost as each agents need to communicate with the central auctioneer.

Markov Decision processes results in more time consuming approach. As it searches all the possible states which give exponential time complexity. MDP also requires the complete view of the system which can't be possible in dynamic environment.

DCOP approaches require less communication overhead as compared to auction-based approach and MDP-based approach. On applying spatial, temporal or communicational constraints, it may results in efficient, scalable and adaptable task allocation.

Swarm-intelligence based approaches consider the ant-colony like behavior. They basically perform the task allocation on the basis of insect-behavior. These approaches are better for dynamic environment as they can work under dynamic working set. But these approaches only consider the local maxima whereas in our problem we require the optimization of global maxima.

The main research challenge in task allocation problem is to get optimal task allocation in multi-agent systems where both agents and tasks can be dynamic in nature. Optimal task allocation is needed for dynamic task allocation in order to:

- Improve the performance of MAS
- Lower the overall task execution time
- Reduce the cost incurred for task allocation
- Minimize the resource and communication between the agents

Thus the research challenge is to find the optimal task allocation for dynamic environment in distributed manner.

IV. Problem Definition

Task allocation problem in dynamic multi-agent systems can be defined as the problem of assigning set of complex task to set of agents where both the agents and the tasks can enter and leave the system over the time. Rescue operation during disaster scenario and automatic target recognition are examples of dynamic real world problem of task allocation. The mechanism to find the solution of task allocation problem in such environment must be adaptable and update the solution as any change occurred in the environment. Against this background, the problem can be formulated in the following manner.

Consider a system consisting of a set of agents **A= {a_1, a_2, a_3….. a_n}**. Each agent is 2-tuple variable i.e., **a = <a_i, R>**, where **a_i** represents the name of agent and **R_i** represents the resources available at agent a_i. The set of tasks that these agents can perform be represented by T i.e. **T= {t_1, t_2, t_3….. t_m}**. The task t_j can be represented by **<t_i, R_t^j, init-agent, deadline(t_j), exec(t_j), reward>** where **t_j** is the task-id, **R_t^j** is the required instances of resources for the completion of task t_j, **init-agent** is the initiator agent at which the task would be arrived, **deadline(t_j)** is the deadline of performing the task t_j, **exec(t_j)** is the execution time taken by the task t_j, **rew(t_j)** is the pay-off given by the task to the agents who are involved in its execution after its successful completion.

As the task arrived at the system, the agent a_i will check its' R_i with the R_t^j of the arrived task t_j. If the agent is capable to perform this task individually then it starts its accomplishment. Otherwise it will negotiate with the other neighbor agents. After getting the responses from those agents, the initiator agent will form the coalition. Coalition Formation is the process of making a group of agent that caters the need of arrived task and accomplish the task successfully and efficiently. The agents will check their capabilities and the reward gained due to the accomplishment of this task. After that agent will check the deadline of that task if it is able to complete that task within the deadline then that agent will be added to the coalition.

Consider the C_k^j represents the possible coalitions C_k for the accomplishment of task t_j. Now Coalition C_k^j which results in maximum system utility will be chosen for that task.

$$\mathbf{C^* = arg\ max_C \Sigma_{C_k \in C}\, \gamma\, (C_K , t_j)}$$

Here, $\gamma(C_k, T_j)$ is the coalition value gained after performing the task T_j by the coalition C_k. The task is then assigned to that coalition for its accomplishment.

When the task will get completed, the capability of each agent will be updated and reward is assigned to the corresponding agents according to their contributions.

V. Proposed Methodology

There are various current research issues in multi-agent system like problem decomposition, task distribution, communication, coordination, conflict resolution. The problem of distributed task allocation in dynamic multi-agent system has been discussed in this paper.

The hybrid approach is used for the task allocation in dynamic multi-agent systems. The problem is divided into 4 phases i.e. **task selection, resource negotiation, coalition formation, task execution.**

In task selection phase, if there is more than one task at a time than the agent will choose the task having more profitability. This can be calculated as:

$$\mathbf{W_{tj} += Req(t_j, R_k)} \quad (1)$$

//sum of resources required by the task t_j

$$\mathbf{P(t_j) = rew(t_j) / W_{tj}} \quad (2)$$

The task with maximum value of $p(t_j)$ will be selected.

After the selection of task, the initiator agent will check its resources. If the required resources are available at the init-agent and it is able to accomplish the task within its deadline then it will start its execution. Otherwise it will negotiate with the neighbor resources. Here the agents are assumed to be cooperative in nature i.e. if they are capable to perform that task then they will immediately starts its accomplishment.

In resource negotiation phase, the init-agent will send the message <R-reqd^i_t, rew(t_j), sender-name, receiver-name> and sets a time threshold for receiving the response from the receivers. Here, R-reqd^i_t is the resources required to accomplish that task i.e.

R-reqd^i_t = $R_t^i - R_i$.

Where R_t^i is the total resources required by the task T_j arrived at agent a_i whereas R_i is the resources that are available at agent a_i and required by task T_j.

On receiving the request, the receiver-agents will check their capabilities and send the available resources to the init-agent. On receiving the responses from the neighbor agent within the time-threshold, it form coalitions and choose the best coalition which will give maximum system utility and complete the task in lesser time.

After the coalition selection, the init-agent informs the members of selected coalition for task execution with the starting time. Before reaching to the start time, if the participating agent receives more beneficial task then it will send the prone message back to init-agent and switches to the more-beneficial task otherwise, an OK-message will be send.

On receiving the prone message form any of the participating agents, the init-agent aborts this allocation and again starts resource negotiation for a new allocation for that task until its deadline met. After sending the OK-message the agent can't be switched to any other task before it completion. In this way the task is accomplished.

VI. Conclusion

Task allocation is major research challenge in multi-agent environment. The task allocation can be done in centralized and distributed manner. Many researchers gave the various approaches for the efficient and effective task allocation. We have discussed various approaches given by many researchers. For the real time scenario, we must refer to distributed task allocation because centralized approach results in single point of failure i.e. if the central agents get failed then the allocation have to be done from the scratch. For the dynamic environment where both the task and the agents can enter or leave the system at any point of time, certain conditions must have be considered. The task must be allocated according to some priority basis. Coalition of agents is formed on the basis of constraints like spatial, temporal or communication constraints. The capabilities of the agent must also be taken into account when allocating the task to them. Thus we are going to design a hybrid approach for task allocation with temporal constraints according to the agent's capabilities. This approach will be implemented and the comparison of this approach will be done with other approaches in future. Spatial constraints will also be applied in future. The Energy-efficient task allocation can also be a fruitful thought which will be applied and used in future.

References

[1] Kathryn S. Macarthur, "Multi-Agent Coordination for Dynamic Task Allocation", PhD [Dissertation], Southampton: University of Southampton, Dec. 2011.

[2] Zheng X, and Koenig S ,"Reaction functions for task allocation to cooperative agents", In Proceedings of the 2008 7th International conference on Autonomous Agents and Multi Agent Systems, AAMAS 2008, May, 12-16., 2008, Estoril, Portuga, pp. 559–566.

[3] Shehory O and Kraus S, “Methods for task allocation via agent coalition formation” Artificial Intelligence, vol. 101, pp. 165–200, May 1998.

[4] A.C. Chapman, R. A. Micillo, Ramachandra Kota and N.R. Jennings, “Decentralized dynamic Task Allocation: A Practical Game- Theoretic Approach,” In Proceedings of 8th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2009,May 10-15, 2009, Budapest, Hungary, pp. 915-922

[5] S. Shaheen Fatime, Michael Wooldridge, “Adaptive task resource allocation in multi-agent systems”, In Proceedings of 5th International Conference on Autonomous Agents, AGENTS’ 01, ACM, New York, NY, USA, 2001, pp.537-544

[6] yan kong, Minjie Zhang, Dayong Ye, “A Group Task Allocation Strategy in Open and Dynamic Grid Environments”, presented at 7th International workshop on Agent based Complex Automated Negotiations, ACAN 2014, Paris, France, May 5-6, 2014.

[7] Scerri P., Farinelli A., Okamoto S. and Tambe, “Allocating tasks in extreme teams”, In Proceeding of 5th International Conference on Autonomous Agents and Multi Agent System, AAMAS 2005,Utrecht, Netherlands, 2005, pp. 727–734.

[8] Petcu, A., and Faltings, B. 2005. S-DPOP: Super stabilizing Fault containing Multiagent Combinatorial Optimization”, In Proceedings of the National Conference on Artificial Intelligence, AAAI-05,Pittsburgh, Pennsylvania, USA, July 2005, pp. 449–454.

[9] K.S. Macarthur, R. Stranders, S.D. Ramchurn and N.R. Jennings, “A Distributed Anytime Algorithm for dynamic Multi-Agent Systems”, In proceeding of 25th National Conference on Artificial Intelligence, AAAI-2011, Aug. 7-11, 2011, San Francisco, pp. 701-706.

[10] S.D. Ramchurn, M. polukarov, Alessandro Farinelli, Cuong Troung, “Coalition formation with spatial and temporal constraint”, In Proceeding of 9th International Conference on Autonomous Agents and Multi-agent System, AA-MAS 2010, May 10-14, 2010, Toronto, Canada, pp-1181-1188

[11] Jipeng Du, Ling Zhou, Peng Qu, Zhen Shi, Yang Lin, “Task Allocation in Multi-Agent Systems with Swarm Intelligence of Social Insects”, In proceedings of 6th International Conference on Natural Computation, ICNC-2010, Aug. 10-12, 2010, Yantai, China, pp-4322-4326.

[12] S.D. Ramchurn, A. Farinelli, K.S. Macarthur, N.R. Jennings, “Decentralized coordination in RoboCup rescue", Published in The Computer Journal, Vol. 53, pp. 1-15, 2010.

[13] Wanyuan Wang, Yichuan Jiang, “Community-Aware Task Allocation for Social Networked Multiagent Systems”, IEEE Transactions on Cybernetics, vol. 44, No. 9, pp. 1529-1543, September 2014.

[14] Philip H.P.Nguyen, Minh-Quang Nguyen and Ken Kaneiwa, “A Belief-Based Multi-Agent Markov Decision Process for Staff Management”, In Proceedings of International Journal of Energy, Information and Communication, Vol. 2, Issue 2, pp.23-39, May 2011

[15] George Thomas, Andrew B. Williams, “Sequential Auction for Heterogeneous Task Allocation in Multiagent Routing Domains”, In proceedings of the 2009 IEEE Conference on Systems, Man and Cybernetics, SMC-2009, Oct. 11-14, 2009, San Antonio, TX, USA, pp. 1995-2000

[16] Nick Jennings, Michael Wooldridge, “Software Agents”, IEEE Review, January 1996, pp. 17-20

[17] Michael Wooldridge, “Introduction to Multi-agent Systems”, England: John Wiley & Sons Ltd., Aug. 2002.

[18] James Odell, “Agent Technology: What is it and why do we care? “, Enterprise architecture Advisory Services, Executive Report, vol. 10, No. 3, 2007. [Online]. Available: http://www.jamesodell.com/Cutter-Exec_Rpt-July_2007.pdf.

[19] Prithviraj Dasgupta, “A Multi-agent swarming System for Distributed Automatic Target Recognition Using Unmanned Aerial Vehicles”, IEEE Transaction on Systems, Man and Cybernetics, vol. 38, No. 3 , pp. 549-563, May, 2008.

[20] Sofia Amador, Steven Okamoto an Roie Zivan, “Dynamic Multi-agent Task Allocation with Spatial and Temporal Constraints”, In proceeding 13th International Conference on Autonomous Agents and Mutli Agent System, AAMAS-2014, May 5- 9,2014,paris, France, pp. 1495-1496

[21] Yongcan Cao, Wenwu Yu, Wei Ren and Guanrong, “An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination”, IEEE Transaction on Industrial Informatics, vol. 9, No. 1, pp. 427-438, Feb. 2013

[22] Frank R. Kshischang, Brendan J. Frey and Hans-Andrea Loeliger, “Factor Graphs and the Sum-Product Algorithm”, IEEE Transactions on Information Theory, Vol. 47, No. 2, pp. 498-519, Feb. 2001

[23] Zheng Xiao, Shengxiang Ma, Shiyong Zhang, “Learning Task Allocation for Multiple Flows in Multi-agent Systems”, In proceedings of International Conference on Communication Software and Networks, ICSSN’09, Feb. 27-28, 2009, Macau, pp. 153-157.

[24] Xing Su, Minjie Zhang, Dayong Ye, “A Dynamic Approach for Task Allocation in Disaster Environments under Spatial and Communicational Constraints”, AAAI workshop on Multiagent Interaction without Prior Coordination, MIPC 2014,July 2014.

[25] Yichuan Jiang, Zhaofeng Li, “Locality-sensitive Task Allocation and load balancing in networked multi-agent systems: Talent versus centrality”, Journal of Parallel Distributed Computing, vol. 71, no. 6, pp. 822-836, June 2011.