Proceedings of the 2011 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Beijing, China

Dynamical Task Allocation and Reallocation Based on Body Expansion Behavior for Multi-robot Coordination System

Guanghui Li, Yusuke Tamura, Hajime Asama

Department of Precision Engineering
The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan

guanghui@robot.t.u-tokyo.ac.jp, tamura@robot.t.u-tokyo.ac.jp, asama@robot.t.u-tokyo.ac.jp

***Abstract* - The inefficiency, exponential amount of communication and computational time are high undesirable for realistic applications under utilizing the existing distributed task allocation approaches, especially for the dynamical tasks that move randomly before assigned robots to execute them, and the condition of these tasks could vary over time. For such dynamical task assignment problem, we propose a dynamical task allocation and reallocation method for multiple robots coordination system based on multi-round negotiation and body expansion behaviour. For the first time round negotiation, robots sequentially negotiate and select tasks to perform according the proposed algorithm, and declare the information to other robots. When all robots have finished first time selecting, then the remaining un-selection robots choose the rest un-assigned tasks again sequentially. We set two distance thresholds for decision making so as to implement body expansion behavior. Based on the body expansion behavior, one robot can request, accept and refuse other robots to execute tasks by intention communication under the order of two distance thresholds. The advantages of dynamical task allocation and reallocation approach is demonstrated by comparing with existing task allocation algorithm in this paper. The simulation results show that the efficiency for whole multi-robot coordination system to accomplish all tasks is improved by utilizing our approach. Moreover, it is more conducive to reduce the numerous computational time and communication compare to the existing investigated task assignment methods.**

Index Terms - Multiple autonomous mobile robot. Task allocation. Task reallocation. Body expansion behavior.

I. INTRODUCTION

Distributed task assignment approaches address such problems that arise with centralized task allocation method, each robot operates largely independently, dealing with information that is locally available through its equipped sensors. A robot can coordinate with other robots to execute the assigned task separately, and perform the tasks which can not be accomplished by other robots, due to malfunction or limited capability of these robots, thus the system is more robust since the entire system's performance no longer depends on a single leader robot. Moreover, robots are better able to respond to unknown and dynamical environment, and the condition vary with time, that based on each robot can perceive the environment locally. However, the number of required communication which makes use of distributed task allocation approach is still excessively high communication requirements, and computational time required to compute an optimal solution is too long, which make the MRCS unable to keep up with real-time execution demands. Thus, neither communication nor computational time is high undesirable for realistic applications, especially for dynamical tasks that move randomly before the assigned robots to execute them, and the condition of these tasks could vary over time. The solution which robots have been assigned to the given tasks may not suit for the next solution when conditions of tasks are changing during time. Therefore, the task allocation system should reallocate robots to tasks so as to find the potential optimal solution to task assignment problem. For such dynamical task assignment problem, we propose a dynamical task allocation and reallocation method for MRCS based on multi-round negotiation and body expansion behaviour[1]. In this paper, we mainly utilize the proposed dynamical task allocation and reallocation approach to improve the efficiency for MRCS to accomplish all tasks. Moreover, it is more conducive to reduce numerous computational time and communication compare to the existing investigated task assignment methods.

The remainder of this paper is structured as follows. The next section discusses related works of task allocation. Section III presents a formal definition of task assignment problem. Section IV describes notion about body expansion behavior, setting two thresholds to make decision, and details the proposed algorithm for MRCS. Section V presents the implementation and discusses simulation results. Finally, section VI draws conclusions and sketches future work.

II. THE RELATED WORKS

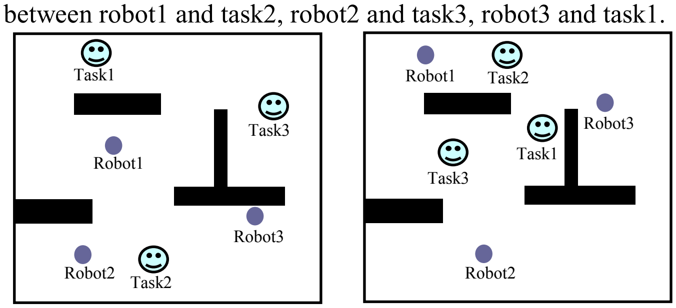
Task allocation for MRCS is a widely studied field. It can be broadly classified into two classes: one is centralized planner based systems, planners are often based on auction mechanisms in which robots bid for tasks, e.g. Gerkey's MURDOCH [2]. As [3] proposes a method for team-task allocation in a multiple robots transportation system, since such kind of system are that agents and tasks are still fixed, in addition capabilities and resources do not depend on time, while in real world application it is not very useful. Another problem is the systems which rely on individual robots to make individual task allocation decision without considering other team member and the optimization of whole system. Empirical results of an auction based algorithm for dynamic allocation of tasks to robots is proposed by [4]. In their researches, they propose a method of repeated auction for distributed tasks dynamically among a group of cooperative robot. The distinctive feature of this algorithm is its robustness to uncertainties and to robot malfunctions that happen during task execution.

Another kind of methodology is distributed task assignment, e.g. Asama et al. develop an autonomous and decentralized robot system called ACTRESS to address the issues of communication, task assignment, and path planning among heterogeneous robotic agents [5, 6]. This approach revolves primarily around a negotiation framework which allows robots to recruit help when needed. Parker formulated a related multi-robot task allocation problem called the ALLIANCE efficiency problem [7-9]. In [10], they introduce a broadcast of local eligibility (BLE) approach to multiple robots coordination. BLE mechanism involves a comparison of locally determined eligibility with the best eligibility calculated by a peer behavior on another robot. A distributed multi-robot cooperation framework for real time task achievement is proposed in [11, 12]. The framework integrates a distributed task allocation scheme, coordination mechanisms and precaution routines for multi-robot team execution. When initial assignments of tasks may become inefficient during real time execution due to the real world issues such as failures, and these allocations are subject to change if efficiency is a high concern, reallocations are needed and should be performed in a distributed fashion. They propose an online dynamic task allocation system for reallocation to achieve a team goal that can respond to and recover from real time contingencies. [13, 14] present a reasonable system that enables a group of heterogeneous robots to form coalitions to accomplish a multi-robot task using tightly coupled sensor sharing. The advantages of this new approach are that it enables robots to synthesize new task solutions using fundamentally different combinations of sensors and effectors for different coalition compositions, and provides a general mechanism for sharing sensory information across network robots. However, all the mentioned above mainly concern the computational performance, and tasks are static, they do not mention the dynamical tasks and method of task reassignment, additionally, they do not discuss the fault tolerance, flexibility and robust, moreover when a robot is failure, the system how to deal with it.

III. TASK DESCRIPTION

In this paper, task assignment problem among multiple, fully distributed, initially homogeneous mobile robots is studied, i.e., we develop a method of task allocation and reallocation which can deal with dynamical tasks. The formal definition of this problem is that assume such kind of environment included two kind of missions, one is initial mission, initial mission is that assign multiple dynamic tasks to robots reasonably and efficiently; another is final mission, the final mission is such system that robot guide task from the initial position to destination, which tasks should reach. For the initial mission, due to the dynamical tasks that move randomly before assigned robots to execute them, and the condition of these tasks could vary over time, we should assign and reassign tasks to robots propertly. We propose a dynamical task allocation and reallocation methodology based on body expansion behavior to solve this kind of problem. While the final mission, when robots move nearby tasks, tasks transmit its destinations to robots, then in each robot global coordinate system, robots find the destination's coordinate, plan the optimal path(e.g., we utilize the particle swarm optimization to motion planning which is proposed in [15]), finally, and guide tasks to its destinations.

Previously, few researchers have done the domain about tasks which are dynamical and move randomly. In mentioned above, all these methods are suitable for the tasks which positions are fixed, while for the dynamic tasks, these methods are inefficient. And they don't discuss the task reallocation during robots are executing tasks except the robot malfunction, communication failure and partial system failure. Actually, for tasks in terms of position change, we should consider not only assign the tasks to robots successfully but also robots perform tasks efficiently for whole coordination system. For example, in previously mentioned above, if we consider homogeneous robots, efficiency of robots to perform tasks depend on time needed by a robot to reach the location of task. This measure depends on the task and robot's position which is a function of time. Therefore, the efficiency for a robot to perform a task varies with time, as a result, robots should select the optimal tasks which the time needed by robots to reach are shortest (i.e., the distance between robots and tasks are shortest.) to perform every time to improve the efficiency for the whole guidance system. As Fig. 1 shown, at T0, the system assign task1 to robot1, task2 to robot2 and task3 to robot3 according to the distances between robots and tasks are shortest. While at T1, since changing positions of tasks the system should reallocate tasks to robots reasonably, because of the values of distance between robot1 and task1, robot2 and task2, robot3 and task3 are large than the distance between robot1 and task2, robot2 and task3, robot3 and task1.



(a) At time T0 (b) At time T1

Fig. 1 Illustration of dynamical task allocation and reallocation

Except for, none of previous approaches explicitly address the problem of minimizing communication, time of path planning, computational time and computational memory. For example, market-based auction methods， ALLIANCE and BLE need each robot plan the path from location of itself to each task, calculate the distances between robots and tasks, when the positions of tasks change. Once the situation of tasks and robots vary, system should auction these tasks for all robots. After bidding tasks, robots which obtained profits are largest for the whole system execute these tasks. In other words, the efficiency of these methods is very low to deal with the dynamical task allocation and reallocation problem, it takes a long computational time to motion planning, distance calculation and tasks negotiation. Both BLE and ALLIANCE methods don't explicitly consider global efficiency, while these methods are satisfied with finding any feasible solution. A notable exception is the work by M. Nanjanath et al. [4], where they propose a method of repeated auction for distributed tasks dynamically among a group of cooperative robot, tasks not yet achieved are re-submitted for bids every time a task has been completed. The repeated auction comes closest to our approach. Main differences include our proposed system reallocation tasks for robots every time step, and we utilize body expansion behavior to reduce the communication when the distance between robot and task is large than a given threshold.

IV. Task allocation and reallocation method

Assume that all robots are homogeneous robots with the same speed, function, and structure, and can communicate with each other using radio frequency broadcast. One robot allocate only single task one time, execute only single task and guide the assigned task to its destination.

The tasks are randomly distributed in the environment, and can move freely with variable speed before robots reach around them. Each task does not know its destination where it is unless under the robot guiding to. And all tasks are waiting for guiding in the priority queue under the principle of "First In First executed". Robot always executes the relative most priority task regardless other tasks move around it.

A. Body expansion behavior

Body expansion behavior means that robot can transmit its own intention and the receiver executes the order, thus robot is capable of controlling the other's behavior. In addition this demonstrates the expansion of the robot's degree of freedom (D.O.F.).

Two distance thresholds for decision making are settled to implement body expansion behavior. One is the small distance threshold θ1, means that robot is about to guide the assigned task. Another is the large distance threshold θ2, means that robot have a long time to guide the assigned task (Fig. 2). If the distance is more thanθ2, robot can request other robots to guide the task, if the distance betweenθ2 andθ1, robot compare the distance and select the shorter distance task to guide, and if the distance is less thanθ1, then robot refuse all the request from other robots. For all robots, there are three working states: 1. *Free-robot*, robot has not been assigned task. 2. *Half-free-robot*, robot has been assigned task but is not guiding the task, or the distance between robot and task is less than θ2 but more than θ1. 3. *Busy-robot*, robot is guiding task, or the distance between robot and task is less than θ1. When robots find remaining un-guided tasks and there are free-robots in the environment, the robot can request the free-robot to guide the remaining un-guided tasks. Another condition, when the value of distance is more than θ2, robot can request other robot guide the task.

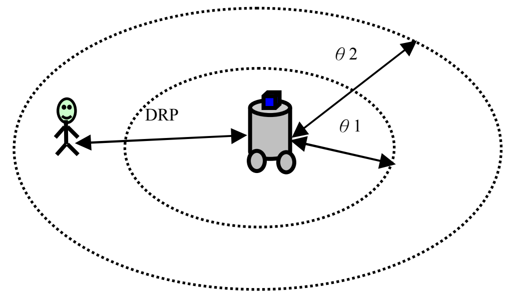


Fig. 2 Distance threshold

B. Proposed task allocation and reallocation algorithm

We propose a task allocation method which can reallocate tasks to robots according to the shortest distance. In the environment, Ri {R1, R2, ・・・, Rn} denotes the i th robot, Tj {T1, T2, ・・・, Tm} denotes the j th task, The DRiTj denotes the utilizable distance from Ri to Tj, and m≥n. Assumption that tasks Ti Є{T1, T2, ..., Tm, iЄN} are distributed in the environment randomly at the same time, and can move randomly, all tasks need robots guide them to their destinations due to these tasks don't know the path to their destinations. In the initial state, the working statuses of all robots are free-robot, and wait for executing tasks.

Tasks broadcast the request information include task IDs and coordinates to all robots every time pulse. In the initial time step (Shown in Fig. 3), there are two times round negotiation and selection for each robot. For the first time round, all robots receive request information from tasks, then plan paths to all tasks and calculate the distances between all tasks in the robot's global map of environment. Robots are priority according to the subscript of robots, such as robot R1 has the largest priority than other robots, the priority of robot R2 is less than robot R1, and so on. That is the priority of robot which subscript is small is larger than the priority of robot which subscript is large. From robot R1 to robot Rm sequentially select tasks to perform according our algorithm, and declare the information to other robots. When all robots have finished the first time selecting, then the remaining un-selection robots choose the rest un-assigned tasks again sequentially in the second time round. In other words, robot R1 receives the request information including robot IDs and coordinates from the first m number of tasks(i.e., m is the number of robot), plans the paths from its location to the positions of all tasks and calculates the distances between them. Robot R1 selects the task to execute under comparing the calculated distances with the given distance thresholds. If there is/are one or more than one distance(s) which between robot R1 and tasks is(are) less than the small distance threshold θ1, R1 selects the task which distance is smaller and broadcasts the execution information(i.e., the execution information is the number of task which is assigned to R1) to other robots. Otherwise, R1 doesn't select any task, requests other robot to execute these tasks and broadcasts the execution information(i.e., none task is assigned to R1). Robot R2 plans the paths from its location to the positions of the rest tasks and calculates the distances between them once R2 receives the execution information from R1. Like the selecting process of robot R1, R2 selects the rest task to execute under comparing the calculated distances with the given distance thresholds. If there is/are one or more than one distance(s) which between the robot R2 and the rest tasks is(are) less than θ1, R2 selects the rest task which distance is smaller and broadcasts the execution information(i.e., the execution information is the number of task which is assigned to R2) to other robots. Otherwise, R2 doesn't select any task, requests other robot to execute these tasks and broadcasts the execution information (i.e., none task is assigned to R2). Robot R3 should receives both execution informations from R1 and R2, that is the later robot Rm should receive all the execution informations from the former m-1 robots, then the robot does the similarly task selection process of R1. Until the last subscript robot do the task selection process, all robots have finished the first time round negotiation.

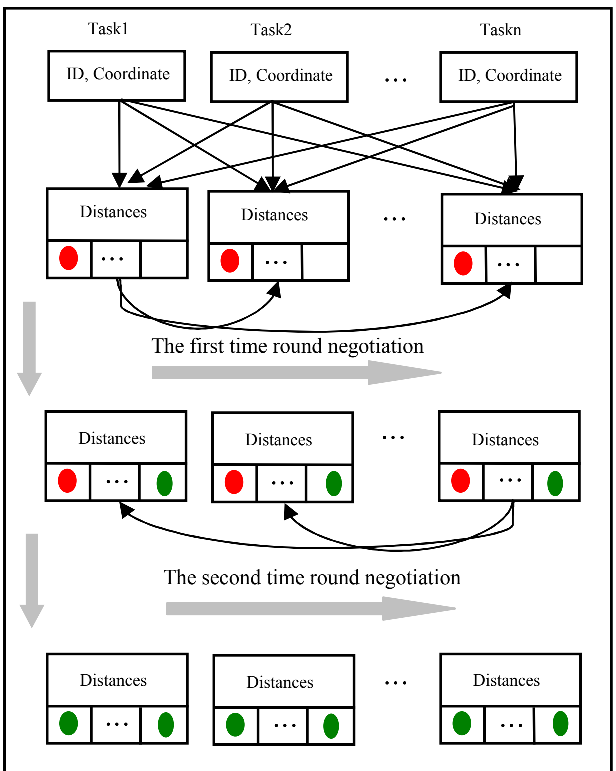


Fig. 3 The illustration of initial step

In the first time round negotiation, there are some robot don't select tasks to execute and some tasks are not assigned to robot. Thus, robots need do second time negotiation. For the second time negotiation, based on the priority of that the smaller subscript of robot firstly does the task selection process, the later subscript of robot should receives the all task selection information from the former robot then can carry out the task selection process, the remaining un-selection robot sequentially selects the un-assigned task which distance is shortest in the un-assigned tasks to perform, even though the distance between them is more the large distance threshold θ2 . The algorithm of initial time step is as follows:

Tasks broadcast the request informations include task ID Ti and the coordinate to each robot Rj.

For *Rj (j=1, j<=n, j++)*

i. *Robot Rj Plan the path for the first n tasks, calculate the distances DRjTi Є{DRjT1, DRjT2, ..., DRjTn, i, j, n ЄN} between robot Rj and all task Ti.*

ii. *Task selection model in robot Rj: Compare the distances DRjTi Є{DRjT1, DRjT2, ..., DRjTn, i, j, n ЄN}.*

If *There are some distances DRjTi Є{DRjT1, DRjT2, ..., DRjTn, i, j, n ЄN} which are small than or equal θ1.*

Then *Select the task Ti to Rj which distance DRjTi is the Shortest.*
Request other robots to execute tasks expect task Ti.
Broadcast the selection information to other robots.
The working status of robot Rj change to busy-robot.

Else *All of the distances DRjTi Є{DRjT1, DRjT2, ..., DRjTn, i, j, n ЄN} are more than the large distance threshold θ1.*
Request other robot to execute the first n tasks Ti.

For *Rj (j=1, j<=n, j++) except busy-robot*
Task selection model in robot Rj:Compare the distances DRjTi Є{DRjT1, DRjT2, ..., DRjTn, i, j, n ЄN}.
Select the Ti to Rj which distance DRjTi is the Shortest.
Broadcast the selection information to other robots.
The working status of Rj change to half-free-robot.

Due to the dynamical tasks that move randomly before the assigned robots to reach around and execute them, the condition of these tasks could vary over time, the distances between robots and the corresponding assigned tasks may vary. The MRCS should reallocate tasks to robots every seconds time based on utilizing body expansion behavior during the implemental period, in order to improve the efficiency of which robots execute tasks for whole system. If the distances between robot and corresponding assigned task is more thanθ2, then robot request other robots to execute this task and broadcasts the information to all other robots. For the other robots which from R1 to Rm sequentially, if the distance between robot and task which assigned in the latest time step is less thanθ2, the working state of robot change to busy-robot and refuse any requests from other robot. Otherwise, robot selects task which distance is shorter and broadcasts the task selection information to other robots. The other robots can make a decision about accept/refuse the request according its calculated distance and the received task selection information. If all other robots refuse the task, the robot should continue select the task to perform despite the distance is moreθ2. Note that robot also request other robot to execute the assigned task when robot is failure. At the next time step, robots continue move toward the assigned tasks which are allocated in the latest time step before the system assigns the new task to robot.

V. SIMULATION AND RESULTS

To demonstrate the validity and efficiency of our approach, a variety of experiments are carried out by computer simulation. The simulation environment without obstacles is built up with the setting of 400*400m^2. Three robots and six tasks are employed. At the initial time step, tasks and robots are randomly distributed in the environment. During the simulation tasks move with the variable speed over time which are shown in Fig. 4, while the speed of robot is constant which is 0.76m/s. The small distance threshold θ1 is 4m. The large distance threshold θ2 is 40m. To compare our approach, we simulate a kind of general approach market-based repeated auction method in the same situation.

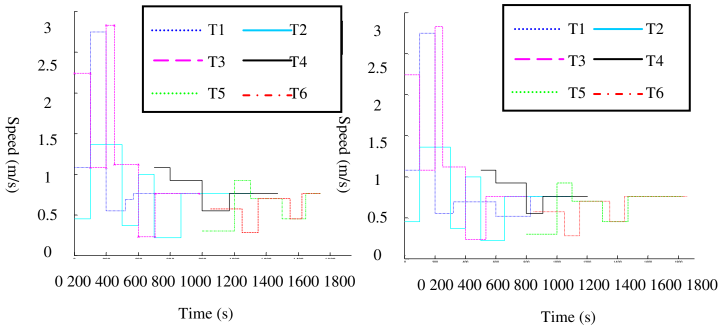


Fig.4 The speed of tasks

Three robots are distributed in the simulation environment, three tasks enter into the simulation environment differential positions at the initial time step. The purposes of tasks are that reach the destinations, while all of them don't know where are the destinations of themselves. Therefore, tasks request robots to guide them to the destinations. But during the time of that robots reach around tasks, tasks can move randomly instead of standing in the specified location for waiting. As the Fig. 5 shown, task T1 is assigned to robot R1 due to the distance between T1 and R1, which is shorter than distances between T1 and R2, T1 and R3 at the initial time step, task T2 and T3 are assigned to R3 and R2, respectively. Robots will move along the planed path to capture tasks, while tasks move randomly.

Fig. 5 is the situation that utilize our approach in every time step, each robot compare the large/small distance threshold to distance DTR which is from location of itself to task, if DTR is large thanθ2, robot request other robot to execute it. At T=62, 110, 230, 475 and 1179, tasks are reallocated to robots because of distances between them are vary. At T=371, robot R3 arrive at T1 and will guide T1 to destination D1. In such situation, R3 will refuse all requirements from other robots, since distance is short thanθ1, T=507, 667, 969, 1422 and 1448 are the same as T=371. T4 walks into the environment at T=500(The same as T=800 and 850 are distributed into the environment). T4 will move randomly under the un-assigned state, due to each robot can assign to only single task to guide each time, until there is a free-robot that like T=705, T1 has arrived at D1 under the R3 guiding, in the next time robot will check whether there is un-assigned task. Robot will assign to the un-assigned task if there is a un-assigned task in the environment like T=705, 1120, 1273 and 1789, or like the situation that robot will move freely due to there is no un-assigned task (as T=782).

Result shows the condition that which robot assigns which task during the simulation time. From the figure, we can see that robots often changed the task to perform according to the shorter distance, but not so frequently as we expected. Simulation results show that the total number of time steps for robots reach around tasks is 3234, while for the first three tasks it only needs 1545 time steps. The total time steps which robots guide the first three tasks and all tasks to the destination are 2607 and 7505.

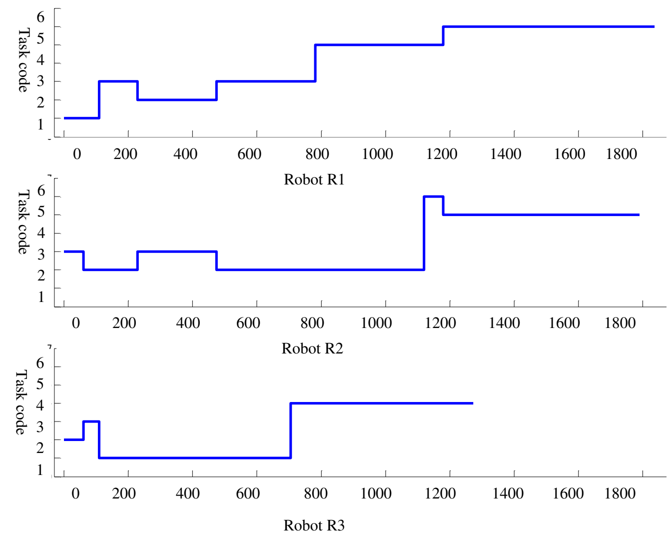


Fig. 5 Task allocation and reallocation based on our approach

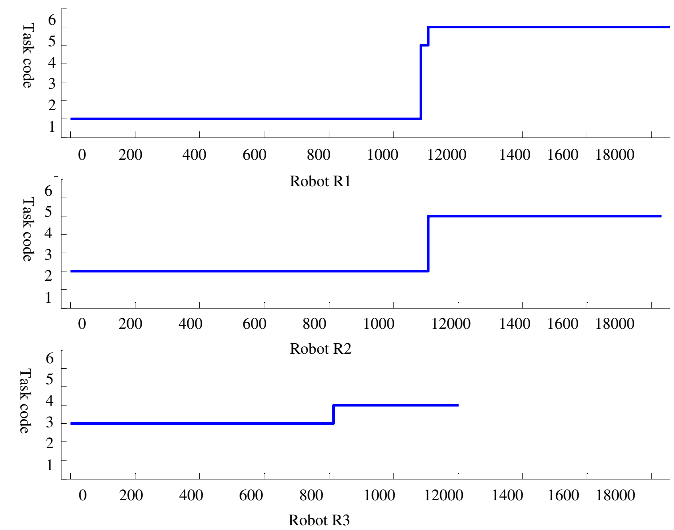


Fig. 6 Task allocation and reallocation based on repeated auction approach

To compare our approach, we simulate another kind of task allocation method market-based repeated auction at the same simulation environment, the simulation results are shown in Fig. 6. At the initial time step, T1, T2 and T3 are allocated to R1, R2 and R3 respectively, since the sum of distances between robots and tasks are shortest by this allocated way for the whole system (T=0). As the result shows that system will not reallocate tasks to robots until any one task is guided to arrive at destination. Such as T=814, 1084 and 1109, the system does not reallocate tasks to robots, due to the sum of distance between robots and tasks are the shortest when T3 and T1 are guided to the destination by robot. However, when T2 arrive at its destination, the system change the latest task allocation strategy, before robot guide T2 reach at its destination, R1 execute T5, and R3 execute T4. After T2 is guided to its destination, robot changes to perform T6, while R2 executes T5.

Fig. 6 shows the condition that which robot assigns which task during the simulation time. The simulation results show that the total number of time steps for robots reach around tasks is 3692. The increased time steps are 458 more than utilizing our approach. While for the first three tasks it needs 2018 time steps. The total time steps which robots guide the first three tasks and all tasks to the destination are 3007 and 7897. The improved performance of our approach relative to the market-based repeated auction method is shown in table1. From the table I proved that for the whole multi-robot coordination system, the efficiency to accomplish the tasks is greatly improved by employing the proposed dynamical task allocation and reallocation method.

TABLE I
THE IMPROVED PERFORMANCE

	Total time step (Reach around tasks)	Total time step (Arrive at destination)	Total time step (Robots reach around the first three tasks)	Total time step (The first three robots arrive at its destination)
Market-based repeated auction	3692	7897	2018	3007
The proposed dynamical task allocation and reallocation method	3234	7505	1545	2607
Improved performance	458(14.16%)	392(4.96%)	473(23.44%)	400(13.3%)

One of the greatest strengths of our dynamical task allocation and reallocation method is its ability to deal efficiently and successfully with the changing conditions. Since our approach does not rely on the initial task allocation and it can task reallocation according to the variable solutions, the MRCS is highly robust to changes with the environment, including malfunctioning robots. Thus, the presented method in this paper allows robots to deal with dynamical environments in an opportunistic and adaptive manner. Moreover, if the distance between robot and task is less than the large distance threshold$\theta 2$, robot only plan the path to the assigned task, thus it is more conducive to reduce the numerous computational time for the entire system. Since the communication between robots is task selection information, communication is greatly decreased comparing to the existing investigated methods.

VI. CONCLUSION

The MRCS based on the method of dynamic task allocation and reallocation is developed in this paper. We propose a task allocation method which can reallocate tasks to robots for dynamical task assignment problem according to the shortest distance. The method for multiple robots coordination system is based on multi-round negotiation and body expansion behavior. To demonstrate the validity and efficiency of the proposed approach, a variety of experiments are carried out by computer simulation. We simulate a kind of general approach market-based repeated auction method in the same situation compare with our approach. The simulation results show that our method is very validity and efficiency than the general task allocation method. Moreover, it is more conducive to reduce the numerous computational time and communication compare to the existing investigated task assignment methods. In the future work, we will implement our approach to the real robot.

ACKNOWLEDGMENT

The present research was supported (in part) through the Global COE Program, “Global Center of Excellence for Mechanical System Innovation” by the Ministry of Education, Culture, Sports, Science and Technology.

REFERENCES

[1] T. Fujiki, K. Kawabata, and H. Asama: “Adaptive Action Selection of Body Expansion Behaviour in Multi-Robot System Using Communication”, Journal of Advanced computational Intelligence and Intelligent Informatics, Vol.11, No.2, pp.142-147, 2007.
[2] B. P. Gerkey and M. J. Mataric: “Auction Methods for Multi-robot Coordination”, IEEE Transactions on Robotics and Automation, Special Issue on Multi-Robot System, Vol.18, No.5, pp.758-768, 2002.
[3] J. Wawerla and R. T. Vaughan: “A Fast and Frugal Method for Team-Task Allocation in a Multi-robot Transportation System”, IEEE ICRA2010, pp.1432-1437, 2010.
[4] M. Nanjanath and M. Gini: “Repeated Auction for Robust Task execution by a Robot Team”, Robotics and Autonomous System, Vol.58, pp.900-909, 2010.
[5] H. Asama, K. Ozaki, A. Matsumoto, Y. Ishida and I. Endo: “Development of Task Assignment System Using Communication for Multiple Autonomous Robots”, Journal of Robotics and Mechatronics, Vol.4, No.2, pp.122-127, 1992.
[6] K. Ozaki, H. Asama, and I. Endo: “Distributed and cooperative object pushing by multiple mobile robots based on communication”, Advanced Robotics, Vol.11, No.5, pp.501-517, 1997.
[7] L. E. Parker: “ALLIANCE: An Architecture for Fualt Tolerance, Cooperative Control of Heterogeneous Mobile Robots”, Proceedings 1994 IEEE/RSJ/GI, International Conference of Intelligent Robotics Systems (IROS 94), pp.776-783, 1994.
[8] L. E. Parker: “Heterogeneous Multi-Robot Cooperation”, Ph. D. Dissertation, Massachusetts Institute Technology, Artificial Intelligence Lab, 1994.
[9] L. E. Parker: “ALLIANCE: An Architecture for Fault Tolerance Multi-Robot Cooperation”, IEEE Transactions on Robotics and Automation, Vol.14, No.2, pp.220-240, 1998.
[10]B. B. Werger and M. J. Mataric: “Broadcast of Local Eligibility for Multi-Target Observation”, Distributed Autonomous Robotic Systems 4, pp.347-356, 2000.
[11]S. Sariel and T. Balch: “A Distributed Multi-Robot Cooperation Framework for Real Time Task Achievement”, Distributed Autonomous Robotic Systems 8, pp.187-196, 2008.
[12]S. Sariel, T. Balch and J. Stack: “Empirical Evaluation of Auction-Based Coordination of AUVs in a Realistic Simulated Mine Countermeasure Task”, Distributed Autonomous Robotic Systems 8, pp.197-206, 2008.
[13]L. E. Parker and F. Tang: “Building Multi-Robot Coalitions Through Automated Task Solution Synthesis”, Proceedings of IEEE, Vol.94, No.7, pp.1289-1305, 2006.
[14]F. Tang and L. E. Parker: “A Complete Methodology for Generating Multi-Robot Task Solutions using ASyMTRe-D and Market-Based task Allocation”, 2007 IEEE International Conference on Robotics and Automation, pp.3351-3358, 2007.
[15]E. Masehian and D. Sedighizadeh: “An Improved Particle Swarm Optimization Method for Motion Planning of Multiple Robots”, Distributed Autonomous Robotic Systems 2010, pp.1-14, 2010.