Proc. of IEEE International Conference on Robotics and Automation, Rome, Italy, 2007.

A Complete Methodology for Generating Multi-Robot Task Solutions using ASyMTRe-D and Market-Based Task Allocation

Fang Tang
Computer Science Department
California State Polytechnic University, Pomona
Pomona, CA, 91768
Email: ftang@csupomona.edu

Lynne E. Parker
Distributed Intelligence Laboratory
Department of Computer Science
University of Tennessee
Knoxville, TN 37996-3450
Email: parker@cs.utk.edu

Abstract— **This paper presents an approach that enables heterogeneous robots to automatically form groups as needed to generate both strongly-cooperative and weakly-cooperative multi-robot task solutions in the same application. The fundamental contribution of this work is the layering of our low-level coalition formation algorithm for generating strongly-cooperative task solutions, with high-level, traditional task allocation methods for weakly-cooperative task solutions. At the low level, coalitions that generate strongly-cooperative multi-robot task solutions are formed using our ASyMTRe-D approach that maps environmental sensors and perceptual and motor schemas to the required flow of information in the robot team, automatically reconfiguring the connections of schemas within and across robots to form efficient solutions. At the high level, a traditional task allocation approach is used to enable individual robots and/or coalitions to compete for weakly-cooperative task assignments through task allocation. We introduce the site clearing task to motivate the work, and then formalize the problem. We then present the approach of layering ASyMTRe-D with task allocation. We validate the approach on a team of robots with the site clearing task. We believe the resulting approach is a flexible system that can handle a broad range of realistic multi-robot applications beyond what is possible using other existing approaches.**

I. Introduction

This paper addresses the problem of synthesizing both strongly-cooperative and weakly-cooperative solutions in the same multi-robot application. In past work, most task allocation approaches in multi-robot teams have dealt with the assignment of *single-robot* tasks, which are tasks (or collections of tasks or subtasks) that can be accomplished independently by a single robot. We call these task solutions *weakly-cooperative*. Another important type of task in multi-robot teams is the *multi-robot* task, which typically requires a *strongly cooperative* solution [3], meaning that the task is not trivially serializable, and cannot be decomposed into subtasks that can be completed by individual robots operating independently. Instead, it requires robots to act in concert to achieve the task. This type of task is also called tightly-coupled or tightly-coordinated. While much prior work has addressed the allocation of weakly-cooperative tasks, and some recent work is beginning to address the allocation of strongly-cooperative tasks, almost no work has been done on combining approaches that can handle both types of task solutions in the same application. The objective of this paper is to present an approach that integrates the two mechanisms into a single framework. More specifically, our approach layers our ASyMTRe-D coalition-formation algorithm for strongly-cooperative task solutions [16] with an auction-based mechanism for achieving the allocation of multiple weakly-cooperative tasks. Robots first form coalitions at the low level to solve a single multi-robot task with a strongly cooperative solution. Coalitions, and possibly individual robots, then compete for tasks (or collections of tasks) at the high level, using the more traditional task allocator for weakly-cooperative tasks. By combining the benefits of coalition formation and task allocation mechanisms, we believe the resulting approach is a flexible mechanism for a broad range of realistic multi-robot applications, with the ability to generate both strongly cooperative and weakly cooperative solution strategies, as appropriate.

The remainder of the paper is organized as follows. Section II provides additional background on our approach. Section III describes an application example to motivate this work. The problem is formalized in Section IV. Section V describes our approach in detail. Experimental validation of the approach is discussed in Section VI. We provide brief concluding remarks in Section VII.

II. Background and Related Work

The task allocation problem is the problem of determining a suitable mapping between robots and tasks. The majority of the work in task allocation for multi-robot systems [14], [23], [2], [8], [4], [24] focuses on allocating *single-robot tasks* to *single-task robots* with either *instantaneous assignment* or *time-extended assignment* (using the taxonometric terms of [7]). Typically, a task is decomposed into independent subtasks [14], hierarchical task trees [24], or roles [19] either by a general autonomous planner or by the human designer. Independent subtasks or roles can be achieved concurrently, while subtasks in task trees are achieved in order according to their precedence constraints. The work of [24] also addresses "tightly-coupled" multi-robot tasks; however, their tasks can be decomposed into multiple single-robot tasks and thus falls into the category of weakly-cooperative tasks. A formal analysis comparing the computation, communication requirements and solution qualities of several well-known approaches is presented in [7].

Multi-robot *coalition formation* deals with the issue of how to organize *single-task* robots into subgroups to accomplish *multi-robot tasks* using a strongly cooperative solution approach (again, using the taxonometric terms of [7]). These multi-robot tasks requiring strongly cooperative solutions are sometimes referred to as *tightly-coupled tasks*. Some researchers refer to teams of robots performing these strongly-cooperative multi-robot tasks as *coalitions* [7].

Some recent work [9], [10], [11], [6], [22], [16] has begun to address this problem of the allocation of tightly-coupled tasks, or the forming of coalitions. The Hoplites approach [10] focuses on the selection of an appropriate joint plan for the team to execute by incorporating joint revenue and cost in the bid. The work in [9] achieves multi-robot task allocation through matching roles with robot capabilities. The work in [11] also matches task required capabilities with robot capabilities and accomplishes multi-robot tasks through combinatorial bids. The approach by Vig and Adams [22] forms robot coalitions by adapting the coalition formation techniques developed by Shehory and Kraus [18] for multi-agent systems to the domain of multi-robot systems.

None of the above approaches, however, are able to autonomously generate joint plans for how robots should work together to solve a multi-robot task. In the general case, robots will have different sensor, effector and computational capabilities. Thus, a team of resource-bounded robots may not individually possess all of the required capabilities to accomplish a task. However, they could work with other robots as a coalition to effectively accomplish the task objectives, if they were able to autonomously form a joint plan (although the solution approach to achieving a joint plan does not necessarily have to use traditional planning approaches). Challenges exist in how coalitions can be automatically formed that efficiently use the sensory and computational resources distributed across different robots.

Our recent work on ASyMTRe-D [16] generates flexible techniques for automating the formation of coalitions (i.e., generating joint plans) to generate strongly-cooperative multi-robot task solutions, which may involve the sharing of sensory, perceptual, and computational capabilities across heterogeneous team members. These solutions are generated on a more fine-grained *schema* level instead of on the traditional sensor/task level. Although ASyMTRe-D provides a way of generating flexible robot coalitions, it is only designed to handle a single strongly-cooperative multi-robot task at a time. For missions of multiple tasks, we would like to also achieve task allocation amongst coalitions and/or individual robots for weakly cooperative tasks, thus combining the benefits of low-level coalition formation with those of high-level, more traditional, task allocation. The purpose of this paper is to describe a complete methodology for generating both strongly-cooperative and weakly-cooperative task solutions in the same multi-robot team application.

III. Motivating Example: The Site Clearing Application

To motivate the need for the combination of strongly-cooperative coalitions with task allocation for weakly-cooperative tasks, we introduce a site clearing application. This site clearing application is a simplified version of the site preparation task [15], which has been identified by NASA as an important prerequisite for human missions to Mars. The site clearing application requires a specific area to be cleared of obstacles, which we simplify to be boxes with different weights or sizes. The objective of the application is to clear the site in as little time as possible while minimizing the cost to the robots (e.g., energy consumption or computational requirements). For the purposes of this discussion, we assume that a map is available to enable the robot team to determine the positions of the obstacles in the area. We assume that the obstacles to be removed from the site can either be pushed outside the area, or can be pushed to a common collection point, as indicated by a beacon. We further assume that a partial-order planner exists to determine the ordering constraints of removing the obstacles, in case certain obstacles need to be removed before other obstacles can be cleared. For example, one obstacle may block the path to another obstacle.

The site clearing application can be decomposed into a series of tasks with ordering constraints. Each task is aimed at removing one obstacle from the site, which we call "Remove Obstacle". Since only some tasks have ordering constraints, the system can allocate a subset of the tasks to the robots for concurrent execution. Thus, when making a task allocation decision, robots are considering more than one task at a time. Since obstacles (boxes) have different weights or sizes, a "Remove Obstacle" task may require a varying number of robots to form coalitions to accomplish the task in a strongly cooperative manner that efficiently uses the available robot capabilities. Additionally, when multiple coalitions are available, the system must determine which coalition is the best fit to the current task.

Note that from our perspective, an individual task cannot technically be categorized in advance as a multi-robot task or a single-robot task. Instead, whether or not the task requires single or multiple robots depends upon the capabilities of the robot team members. Some robots may be able to perform a given task on their own (thus making the task a single-robot task), while other robots may require help from teammates to accomplish that same task (thus making that same task a multi-robot task). Our ASyMTRe-D approach for generating strongly-coupled task solutions is able to find combinations of robot capabilities that can accomplish the task in either the single-robot case or the multi-robot case, depending upon the team capabilities. It is challenging to predefine solutions for the team when the team capabilities are unknown at design time or they change during the task execution.

IV. Formalism of the Problem

The problem we address can be formally defined as follows:

- $R = \{R_1, R_2, \cdots, R_n\}$ is a collection of n robots, where each robot R_i is represented by its available environmental sensors (ES), and its corresponding perceptual schemas (PS), motor schemas (MS), and communication schemas (CS).
- T is the team-level task to be accomplished, which is denoted as $T = \{t_1, t_2, t_3, \cdots\}$.
  - A set of *ordering constraints* defines a proper partial order of tasks. $t_i \prec t_j$ means that task t_i must be executed sometime before task t_j.
  - A set of *open preconditions*. A precondition is open if it is not achieved by some task in the plan.
  - A subset T^i of T can be allocated to robots concurrently if the tasks in T^i do not have ordering constraints and their preconditions are not open.
  - Each task t_i is further defined as a set of motor schemas that need to be activated in certain ways in order to accomplish this task.
- To accomplish a subset of tasks T^i, a collection of m coalitions, denoted $C^i = \{C^i_1, C^i_2, \cdots, C^i_m\}$, needs to be generated based on the task requirements of T^i and the robot capabilities [16].
- With multiple solutions available, we define a *cost* function for each robot, specifying the cost of the robot performing a given task, and then estimate the cost of a coalition performing the given task. We consider two types of cost:
  - A *robot-inherent* cost measures the inherent cost (e.g., in terms of energy consumption or computational requirements) of using particular capabilities on the robot (such as a laser or a mapping algorithm). We denote robot R_i's inherent cost by *robot_cost*(R_i).
  - A *task-specific cost* measures cost according to task-related metrics, such as time, distance, success probability, etc. We denote the cost of R_i performing task t_j by *task_cost*(R_i, t_j).
  - The *cost* function of R_i performing t_j is represented by *cost* (R_i, t_j), which is a weighted combination of both the robot-inherent cost and task-specific cost, normally in the form of a linear function. Other type of costs can also be incorporated when necessary.
  - The cost of a coalition C_i performing a task t_j is the sum of individual cost of robots that are in the coalition, which is denoted as:

$$cost(C_i, t_j) = \sum_{R_k \in C_i} cost(R_k, t_j) \qquad (1)$$

The problem we address here is: Given (T, R), assign a set of tasks T^i to coalitions of R such that the sum of the coalition costs $\sum_{t_k \in T^i, C_j \in C^i} cost(C_j, t_k)$ are minimized.

V. The Approach: Layering Coalition Formation with Task Allocation

To allocate multi-robot tasks to a team of robots, we propose an approach encompassing four main steps as shown in Table I. Figure 1 describes a general procedure that first decomposes a team-level task to a set of tasks with ordering constraints. At the low level, coalitions from the team of robots are formed to address the given tasks. These coalitions are not distinct, but may share same team members. The coalitions then compete for the assignment of tasks using a traditional task allocation approach.

TABLE I

Allocating Multi-Robot Tasks to a Team of Robots

Input: (T, R)

1) Find the set of tasks T^i up to a constant number[a], such that both the ordering constraints and the preconditions of tasks are satisfied.
2) Configure solutions for each task t_j in T^i by forming a set of coalitions C^i, based on t_j's objective and the current team capabilities.
3) Allocate tasks in T^i to coalitions in C^i, such that:
   - The task-specific cost and the robot-inherent cost are minimized for the set of tasks.
   - A coalition can win at most one task at a time. Assuming $C' \subseteq C^i$ is the set of coalitions selected to perform the tasks in T^i, then the following condition must be satisfied: $\forall_{C'_j, C'_k \in C', j \neq k}, C'_j \cap C'_k = \emptyset$.
4) Monitor the execution of tasks. If the entire task is not complete, start the allocation process (go to step 1) when robots are within Δt time of completing their current tasks (i.e., begin planning the next task as the current task is nearing completion). Otherwise, exit.

[a]Note that the maximum number of tasks allowed for allocation is limited to a constant number b to decrease the computational complexity of the allocation of multiple tasks at once.

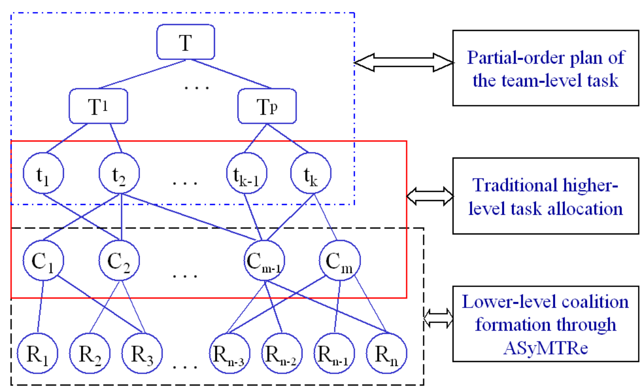

Fig. 1. The relationships between tasks, coalitions and robots.

A. Low-Level Coalition Formation

To better understand the integrated system, we now describe our previous work on coalition formation, called ASyMTRe-D [16]. The ASyMTRe-D approach has been developed for addressing the formation of heterogeneous robot coalitions that solve a single multi-robot task using a strongly-cooperative task solution. Even though we are not using the traditional definition of coalition by calculating payoffs as in game theory [12], we share the same motivation behind coalition formation as mentioned in [18]; that is, robots in a coalition should work together to share resources and cooperate on task execution due to their decision that they would benefit more from working together as a coalition than they would working individually. More generally, this approach deals with the issue of organizing robots into subgroups into a strongly cooperative solution that accomplishes a task collaboratively based upon their individual capabilities.

The fundamental idea of ASyMTRe-D is to change the abstraction that is used to represent robot competences from the typical "task" abstraction to a biologically-inspired "schema" abstraction and provide a mechanism for the automatic reconfiguration of these schemas to address the multi-robot task at hand. To achieve this, we view robot capabilities as a set of environmental sensors that are available for the robot to use, as well as a set of perceptual schemas, motor schemas, and communication schemas that are pre-programmed into the robot at design time.

The ASyMTRe-D approach extends prior work on schema theory [1], [13] by autonomously connecting schemas at run time instead of using pre-defined connections. According to information invariants theory [5], the information needed to activate a certain schema or to accomplish a task remains the same regardless of the way that the robot may obtain or generate it. We can therefore label inputs and outputs of all schemas with a set of information types, such as *laser range data*, *self global position*, etc. Two schemas can be connected if their input and output information labels match. Thus, schemas can be autonomously connected within or across robots based upon the flow of information required to accomplish a task. With the run time connection capabilities, task solutions can be configured in many ways to solve the same task or can be reconfigured to solve a new task.

We have implemented the ASyMTRe-D approach using a distributed negotiation protocol [16] inspired by the Contract Net Protocol [20]. We validated this approach through simulation and physical experiments and analyzed its performance in terms of robustness, scalability, and solution quality. These experimental results allowed us to conclude that the ASyMTRe-D approach provides beneficial mechanisms for multiple robots to: (1) synthesize strongly-cooperative task solutions using different combinations of robot sensors and effectors, (2) share information across distributed robots and form coalitions as needed to assist each other in accomplishing the task, and (3) reconfigure new task solutions to accommodate changes in team composition and task specification, or to compensate for faults during task execution. Thus, the ASyMTRe-D approach can serve as the low-level solution generator for strongly-cooperative task solutions in our approach.

B. High-Level Task Allocation through Auctions

Although ASyMTRe-D provides the mechanism for a heterogeneous robot team to generate a strongly-cooperative task solution by forming coalitions, it can only handle one multi-robot task at a time. We therefore propose the use of an auction mechanism to provide a high-level task allocation approach on top of ASyMTRe-D for handling multiple weakly-cooperative tasks. Note that the intent of this approach is not to develop a new auction mechanism, but instead to layer existing auction mechanisms with the ASyMTRe-D approach to achieve the allocation of both strongly- and weakly-cooperative tasks in the same application. The following high-level auction process is similar to [9], although, as we have stated, the techniques for strongly-cooperative coalition formation are different. Additionally, in the general case, we allow the simultaneous allocation of multiple tasks at a time instead of only one.

The auction process is described as follows:

1) *Task announcement*: Initially, the human operator introduces the site clearing task T to the system. Each task t_i in T is embedded with task-specific information, such as the size and the position of the obstacle to be removed. The human operator has an interface "Auctioneer" that interacts with the other robots in the system (similar to OpTrader in [4]). This auctioneer holds the partial-order plan for T, selects a subset of tasks T^i that satisfies the ordering constraints and the preconditions, and makes an auction call of all tasks in T^i to the robots.
2) *Coalition formation*: Robots that receive T^i start negotiating with others to generate solutions for accomplishing tasks in T^i. These solutions may be either strongly-cooperative solutions involving multiple robots, or solutions that require only a single robot, in the case that such a capable robot is available. For each task t_j in T^i:
   a) Each robot tries to find a list of coalitions (up to a constant number c) that it can join to accomplish t_j. The revised ASyMTRe-D negotiation protocol returns the top c coalitions given a task. The size of a coalition is limited to a max coalition size d assuming robots work in a non-super-additive environment [18][1].
   b) Coalitions are not arbitrarily formed, but are selected based on the combination of the robot-inherent cost and the task-specific cost (please refer to *Formalism of the Problem* Section for details of cost estimation.).
3) *Bid submission*: Once coalitions are formed for each task t_j, a randomly selected coalition leader submits a bid to the auctioneer, including information such as the list of coalition members, the cost of this coalition performing t_j, the leader of the coalition, etc.
4) *Winner determination*: Once bids for all tasks in T^i are collected or a timeout has expired, the auctioneer then determines the winner coalition for each task. The goal for the auctioneer is to find a coalition C_j for each task t_j, such that the total cost of performing the tasks in T^i is minimized and there is no overlapping

[1]Due to the similarity between our configuration algorithm and the coalition formation algorithm presented in [18], we plan to analyze the bounds on our solution quality in future work. It has been proved in [18] that similar algorithms are of low logarithmic ratio bounds to the optimal solution.

of coalition members assigned to the tasks. If no such coalition C_j exists for task t_j and C_k for t_k such that $C_j \cap C_k \neq \emptyset$, then one of the tasks (either t_j or t_k) is auctioned again in the next round. If there is no coalition to accomplish task t_i, t_i is set aside and will not be auctioned off again. The problem of determining the winner is equivalent to the combinatorial auction where multiple tasks are offered and each coalition can bid a subset of tasks. Existing combinatorial auction clearing algorithms (such as [17]) can be applied here with a constraint that the assigned coalitions do not overlap for different tasks.

5) *Award acceptance*: Once winner coalitions are determined, the auctioneer awards each task to the leader of the selected coalition. The leader robot then contacts the other coalition members for the acceptance of the task. Once responses from other coalition members are received, the leader robot accepts the award by sending a task acceptance message to the auctioneer and the coalition members commit themselves to the task until the task is complete. If there is no acceptance message received after certain timeout, the auctioneer awards the next available coalition in the list. If no coalition responses within certain time, the award is rejected and the task will be auctioned again.

VI. Experiments

To date, we have implemented a basic auction-based system that performs task allocation with instantaneous assignment[2], and integrated it with our ASyMTRe-D algorithm. This integrated system is validated through the site clearing task as described in Section III with the difference that tasks are auctioned off on a first-come-first-served basis. To minimize the time spent to clear the site, a greedy algorithm is applied, meaning that the current task under consideration is allocated to the coalition that could accomplish this task with the least cost. The cost is a weighted combination of the task- and coalition-related cost and the inherent cost of the coalition performing the task. The inherent cost is determined by the sensing and computational costs that are required to accomplish the task (see [16] for details). The task- and coalition-related cost is decided by the task completion time $t_{complete}$, as follows:

$$t_{complete} = t_{nav} + t_{push} \quad (2)$$

$$t_{nav} = \max_{R_j \in coalition_i} dist(R_j, box)/speed(R_j) \quad (3)$$

$$t_{push} = \max_{R_j \in coalition_i} dist(box, goal)/speed(R_j) \quad (4)$$

Given the above functions, a robot can incorporate its speed and position information into the bid and share this information among other coalition members to calculate the overall coalition cost. The time for a coalition to accomplish a "Remove Obstacle" task depends on the slowest robot

[2]In instantaneous assignment, only one task is considered at a time, which is a special case of considering b tasks at a time, where (b = 1). The time-extended assignment ($b \geq 1$) remains as a future work.

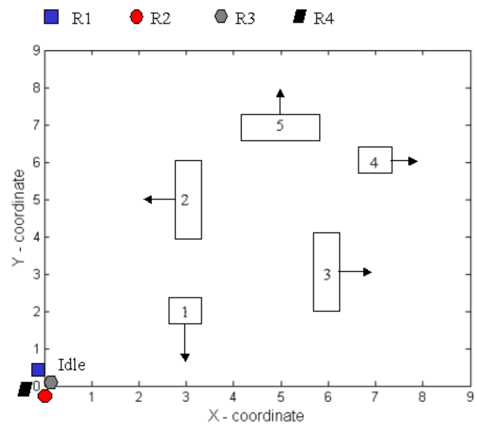

Fig. 2. The site setup in simulation. For the trajectory of a sample run in the above simulation, please see Figure 7.12 in [21].

TABLE II

Robot capabilities in the site clearing task.

Robot	Available Sensor(s)
R_1 and R_4	`sonar, laser, camera, comm`
R_2	`laser, camera, comm`
R_3	`sonar, camera, comm`

in the coalition. In the following sections, we describe the simulation results and physical experimental results.

A. Simulation setup and results

In the simulation setup, four heterogeneous robots are required to clean a $10 \times 10m^2$ area with five boxes scattered in the area, as shown in Figure 2, in which three large boxes need to be pushed on both ends (i.e., require a strongly-cooperative solution) and two small boxes require only a weakly-cooperative solution (i.e., can be pushed by a single robot). The arrows on the boxes represent the desired pushing directions. The types of robots we use are shown in Table II. We reuse the schemas of our multi-robot navigation and multi-robot box pushing tasks [21] for the robots to navigate in the environment, identify boxes, and push boxes. Different robot capabilities result in different solutions (i.e., different coalitions between robots). For example, a robot with sonars can use schemas that enable it to move along the side of a box, and thus push on both ends in a sequential manner. On the other hand, a robot with a laser scanner is not programmed with this schema, and therefore needs to strongly-cooperate with another robot to push a long box. When correctly configured (which is done automatically using the ASyMTRe-D approach), the schemas enable either a single robot or a robot coalition to independently or cooperatively push a box towards the goal.

We ran over 10 logged trials of the site clearing experiment with the above environmental setup, and with random task sequences. The robot team is able to accomplish the site clearing task in an average of 151.2 seconds with a standard deviation of 9.1 seconds, with robot speed varying from .5m/s to 1m/s. To demonstrate the entire task allocation and coalition formation process, we kept a record for the major

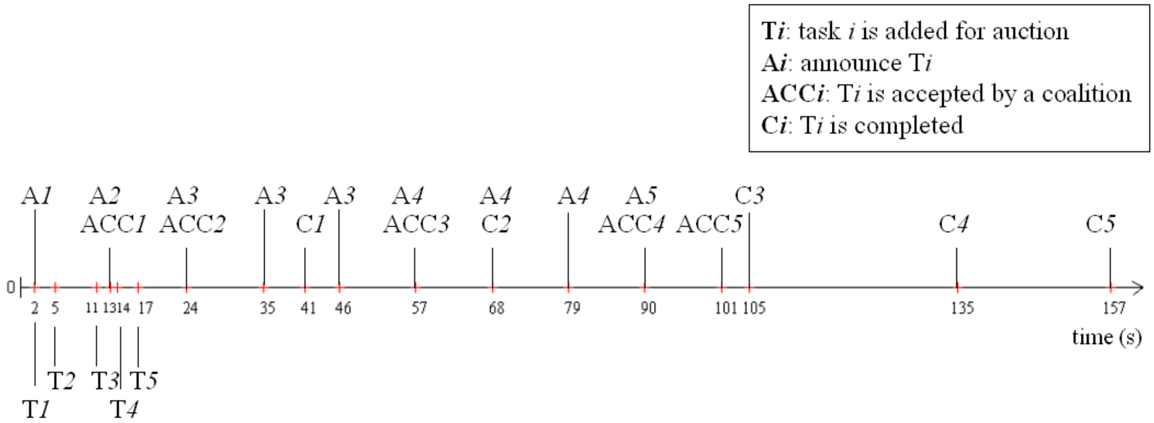

Fig. 3. The timeline of major events from the auctioneer's point of view.

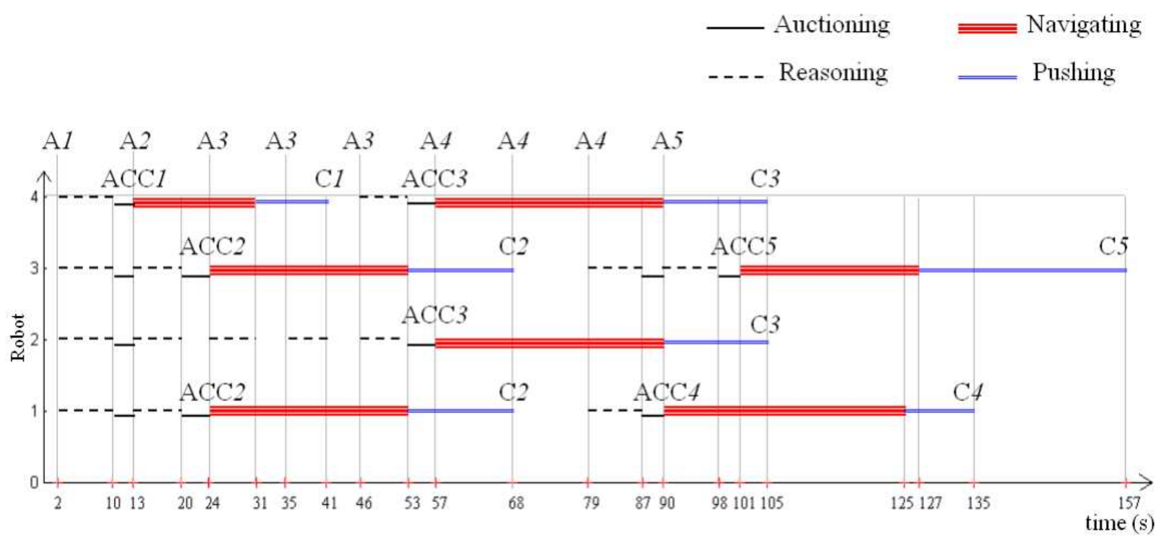

Fig. 4. The timeline showing the state of each robot during the task execution.

events of the auctioneer and each robot. Each event record consists of the time and a description of the event, such as in the example illustrated in Figure 3. For example, at time 2, task 1 is added for auction (T_1) and the auctioneer announces task 1 (A_1). This begins the process in which robots reason to determine their bids. The process is completed at time 13, at which time task 1 is accepted by the winning coalition (ACC_1). In the meantime, tasks 2 and 3 are added to the queue for auction (T_2, T_3), at times 5 and 11, respectively. Once the winning coalition is determined, the auctioneer announces the next task in the queue at time 13 (A_2), and the process repeats.

During execution, robots are always in one of the following states: reasoning, auctioning, navigating, pushing, and idle. A robot starts reasoning with ASyMTRe-D when it receives a task announcement. A robot is in the auctioning state when it is communicating with the auctioneer to bid for a task, or to accept the task award. A robot is idle when it is waiting for incoming tasks. Figure 4 illustrates a typical example, showing each robot's current state during execution. In this example, at time 2, all robots receive the task announcement of removing box 1 and start reasoning to form coalitions. At time 10, coalitions are formed and robots start to bid for the task and wait for the award. At time 13, the task is assigned to R_4, and the rest of the team starts reasoning on the next available task. Note that at times 24 and 35, R_2 continues to reason on task T_3, but fails to generate any solution because of its limited sensing and/or computational capability. At time 53, R_2 finally forms a coalition with R_4 to accomplish T_3. In the end, we can see that boxes 2 and 3 are both pushed by two robots; however box 5 is pushed only by R_3 since it is the only robot available at that time that is capable of pushing the box on both sides.

In the above simulation, the reasoning and auctioning times are decided by the various timeouts in the distributed ASyMTRe-D negotiation protocol and in the auction algorithm. The following timeout parameters are used in the auction algorithm: (1) the time to wait for incoming bids for a specific task announcement (10 seconds), (2) the time for assigning the task to the winning robot coalition (6 seconds),

TABLE III

AVERAGE COMPLETION TIME IN THE BOX PUSHING TASK

Team size	Obstacles	T/R ratio	Avg. Time	Std. Dev.
3	6	2	244.2s	16.6s
3	10	3.3	341.2s	37.9s
6	6	1	143.2s	17.1s
6	10	1.67	202s	20.6s

and (3) the time to wait for confirmations from all coalition members before giving up and assigning the task to the next available coalition (4 seconds). We can see that each robot spends a reasonable amount of time bidding and reasoning on the task compared with the execution time. Fine tuning of the timeout parameters may also result in a shorter task completion time. We have also tested the system with more complex setups. We have varied the number of robots from 3 to 6 and the number of boxes from 6 to 10. Results from 20 runs with random task sequences (reported in Table III) have shown that the robots can successfully form strongly- or weakly-cooperative solutions to accomplish each task. The simulations results also show that the scalability of the integrated system is directly proportional to the task/robot (T/R) ratio. Theoretically, the ASyMTRe-D coalition formation algorithm scales linearly with the increasing team size [16]. The auction-based task allocation algorithm scales linearly with the increasing number of tasks, since tasks are announced sequentially in the current experiments. Our future work includes improving the current auction-based algorithm such that multiple tasks can be announced to robots concurrently, saving time for configuring solutions sequentially.

The auction-based task allocation with instantaneous assignment is very similar to MURDOCH [8] or TraderBots [4]. As has been analyzed in [7], the solution quality for such a task allocation approach is 3-competitive, meaning that the approach is able to find a solution whose utility is never less than $1/3$ of the optimal utility. In our simulation example above, the best plan would be for R_1 and R_4 to move box 2 and R_2 and R_3 to move box 3 simultaneously, and then for R_1 and R_4 to move box 5 cooperatively, and finally for R_2 and R_3 to move box 1 and box 4 separately. The estimated time of completion for the best plan is about 100 seconds, based on the typical auction and task execution times. When the simulation result (151.2 seconds) is compared with the optimal result (100 seconds), we can see the solution quality for our task allocation approach is about 3-competitive, but at least 2-competitive. Again, our purpose here is not to re-invent a new auctioning capability, but to take advantage of these existing algorithms to create a complete methodology that can generate both strongly-cooperative and weakly-cooperative task solutions in the same application.

The simulation results illustrate the success of layering ASyMTRe-D for low-level coalition formation (for generating a single, strongly-cooperative multi-robot task solution), with a higher level, auction-based task allocation approach (for solving a set of weakly-cooperative tasks). The resulting approach provides flexible mechanisms for a broad range of realistic multi-robot applications, with the ability of the robot team to generate both strongly cooperative and weakly cooperative solution strategies without predefined solutions, plans, or roles.

B. Physical robot experiments

The integrated system has also been tested on physical robots. The upper left subfigure of Figure 5 shows the environment setup for the task. The robot team includes two Pioneer robots, each with a laser and a camera. The site is a $4 \times 5m^2$ area with 3 boxes scattered inside. Two of the boxes (1 and 2) are small boxes and the other box is a long box that needs to be pushed from both ends. The objective of the task is to push the boxes to several predefined collection points, which are represented by red flags. Figure 5 shows a series of snapshots taken during one run of the site clearing task. Tasks are introduced to the system in the sequence of push box 1, push box 2, and push box 3. When T_1 (i.e., push box 1) is announced, both robots configure their solutions to move box 1. Since R_1 takes a shorter time to accomplish this task, it wins the task at the end of the auction. R_2 then configures its solution to remove box 2 and wins the task at the end. After R_1 completes its current task, it starts to generate solutions to push box 3; however, it does not possess the capability to push the box alone. Thus it waits until R_2 is completed with box 2 and then forms a coalition with R_2 to push box 3 cooperatively. All three boxes are moved to their nearest collection points at the end of the task.

VII. CONCLUSION AND FUTURE WORK

We have described our approach for layering our coalition formation mechanism for generating strongly-cooperative task solutions with a traditional auction mechanism for assigning weakly-cooperative tasks. The low-level ASyMTRe-D approach automatically forms coalitions according to the task objective, without using any pre-defined plans for how that task will be achieved. The high-level auction-based task allocation provides the mechanism for the team to allocate sets of weakly-cooperative tasks (any of which may itself require a strongly-cooperative solution), holding auctions to assign tasks to the best-fitting individual robots or coalitions.

Our ongoing work includes improving the high-level auction-based approach such that it enables a set of multi-robot tasks to be allocated simultaneously (time-extended assignment) instead of instantaneous assignment. We also believe that the ASyMTRe-D approach for coalition formation can be merged with other, non-auction-based approaches to task allocation, such as the motivation-based approach of ALLIANCE [14]. We believe it would be interesting to investigate the combination of ASyMTRe-D and ALLIANCE, as an alternative approach for achieving applications requiring both strongly-cooperative and weakly-cooperative task solutions.

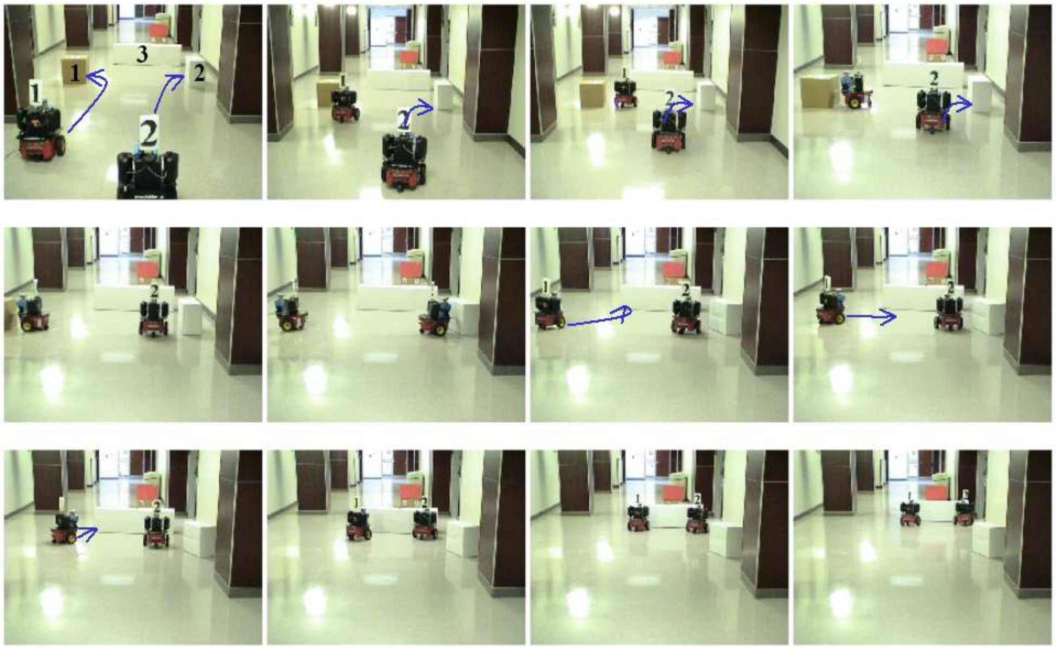

Fig. 5. A series of snapshots taken during one run of the site clearing task. The arrows represent the directions of motion of the robots.

REFERENCES

[1] R. C. Arkin. Motor schema based navigation for a mobile robot: an approach to programming by behavior. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 264–271, 1987.

[2] S. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1234–1239, 1999.

[3] Russell G. Brown and James S. Jennings. A pusher/steerer model for strongly cooperative mobile robot manipulation. In *Proceedings of 1995 IEEE International Conference on Intelligent Robots and Systems (IROS '95)*, volume 3, pages 562–568, 1995.

[4] M. B. Dias. *TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2004.

[5] B. R. Donald. Information invariants in robotics. *Artificial Intelligence*, 72:217–304, 1995.

[6] C.-H. Fua and S S Ge. COBOS: Cooperative backoff adaptive scheme for multirobot task allocation. *IEEE Transactions on Robotics*, 21(6):1168–1178, 2005.

[7] B. Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[8] B. P. Gerkey and M. J. Mataric. Sold! auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, 2002.

[9] E. Jones, B. Browning, M. B. Dias, B. Argall, M. Veloso, and A. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *Proceedings of IEEE International Conference on Robotics and Automation*, May 2006.

[10] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: a market-based framework for complex tight coordination in multi-robot teams. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2005.

[11] L. Lin and Z. Zheng. Combinatorial bids based multi-robot task allocation method. In *Proceedings of IEEE International Conference on Robotics and Automation*, April 2005.

[12] R. D. Luce and H. Raiffa. *Games and Decisions*. John Wiley and Songs, Inc, 1957.

[13] D. M. Lyons and M. A. Arbib. A formal model of computation for sensory-based robotics. *IEEE Transactions on Robotics and Automation*, 5(3):280–293, 1989.

[14] L. E. Parker. ALLIANCE: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2), 1998.

[15] L. E. Parker, D. Jung, T. Huntsberger, and P. Pirjanian. Opportunistic adaptation in space-based robot colonies: Application to site preparation. In *Proceedings of World Automation Congress*, 2000.

[16] L. E. Parker and F. Tang. Building multi-robot coalitions through automated task solution synthesis. *Proceedings of IEEE*, 94(7):1289–1305, 2006. Special Issue on Multi-Robot Systems.

[17] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: a fast optimal algorithm for winner determination in combinatorial auctions. *Management Science*, 51(3):374–390, 2005. Special issue on Electronic Markets.

[18] O. Shehory. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

[19] R. Simmons, S. Singh, D. Hershberger, J. Ramos, and T. Smith. First results in the coordination of heterogeneous robots for large-scale assembly. In *International Symposium on Experimental Robotics*, December 2000.

[20] R. G. Smith. The Contract Net Protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12), December 1980.

[21] F. Tang. *ASyMTRe: Building coalitions for heterogeneous multi-robot teams*. PhD thesis, The University of Tennessee, Department of Computer Science, 2006.

[22] L. Vig and J. A. Adams. Multi-robot coalition formation. *IEEE International Conference on Robotics and Automation*, 22(4):637–649, 2006.

[23] B. B. Werger and M. J. Mataric. Broadcast of local eligibility for multi-target observation. In *Distributed Autonomous Robotic Systems 4*, pages 347–356, 2000.

[24] R. Zlot and A. Stentz. Market-based multirobot coordination for complex tasks. *International Journal of Robotics Research*, 25(1), January 2006. Special Issue on the 4th International Conference on Field and Service Robotics.