Market-based Approaches for Coordination of Multi-robot Teams at Different Granularities of Interaction

Anthony Stentz, M. Bernardine Dias, Robert Zlot, Nidhi Kalra
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{axs,mbdias,robz,nidhi}@ri.cmu.edu

Abstract--*Multi-robot teams can improve safety and increase human productivity for operations in hazardous environments. To be effective, a control scheme is needed to decompose a task, assign subtasks to individual robots, and synchronize execution. We have developed a market model for this control scheme that realizes the best of both centralized and distributed approaches. In the market approach, robots coordinate opportunistically to meet team constraints and to optimize the team solution. In this paper, we illustrate how the market is used to coordinate at the task decomposition, assignment, and execution phases, depending on the requirements of the given application. We present results from simulation and from actual robots for the applications of mapping, area reconnaissance, and perimeter sweeping.*

I. INTRODUCTION

Mobile robots are useful tools for inspection, maintenance, and clean-up of hazardous environments. Robots minimize human exposure to hazardous agents and can increase a human's productivity through semi-autonomy. Multi-robot teams further increase productivity through execution concurrency. Moreover, they make it possible to perform more complex tasks that require a heterogeneous group of specialists, such as assembly or material handling.

In order to perform a task with a team of mobile robots, three operations must be performed. First, the task must be decomposed into subtasks that can be executed by individual robots. Second, the subtasks must be assigned to the robots. Third, the subtasks must be interconnected and synchronized so that constraints are satisfied and the execution of the entire team task is optimized.

One approach to the problem is to use a centralized decision maker [1] [2] that decomposes the task, assigns subtasks, and dictates how synchronization will occur. The advantage of this approach is that it is optimal. The disadvantages are that it is intractable for many problems and sluggish to change or unexpected situations. Another approach is to fully distribute the system [3] [4]. In this approach, each robot decomposes the task as it sees fit, performs part of it, and at most minimally synchronizes with the other robots. The advantage of this approach is that it is flexible and robust to change. The disadvantage is that it is suboptimal, since several robots can perform the same subtask, and some robots end up performing subtasks better suited for other team members.

To realize the best of both approaches, we have developed a market based approach for multi-robot teams called Traderbots [5] [6]. In this approach, the robots are paid revenue for accomplishing subtasks and incur costs for consuming team resources. The robots bid on subtasks, attempting to maximize their revenues and minimize their costs. The robots seek to maximize their individual profits, leading the team toward the most cost effective solution. This effect occurs because a robot can increase its profit by computing a better team solution and using part of the cost savings to buy team participation. The market behaves more like a centralized system when ample time, computing resources, and good intra-team communication permit. In these cases, individual robots compute optimal solutions to the entire team task or to large parts of the task and bid them off against alternative plans produced by their team mates. The market behaves more like a distributed system when time is short, computing is limited, and communication is sparse. In these cases, individual robots perform simple decompositions and subtask assignments that are globally suboptimal but still accomplish the team task.

We have applied the market approach to a number of tasks, including mapping, reconnaissance, and perimeter sweeping. As noted above, the robot team can typically improve the efficiency of its solution through intra-team coordination. The type and granularity of coordination possible is heavily dependent on the application. For example, consider the task of efficiently mapping a building by minimizing the sum of the distances travelled by all of the robots. For this problem, the subtasks are just portions of the map to build. Each subtask is performed by a single robot. The robots need to coordinate only to determine which robot should map which portion of the building. Thus, coordination at a coarse scale, for the assignment of independent tasks, is sufficient.

For other applications, more than one robot is needed to perform a given subtask. For example, in an assembly task, one robot needs to hold one component while another robot attaches another component. In area reconnaissance, multiple robots may be needed to simultaneously observe an area of interest. Alternatively, a task may be so complex that intermediate levels of task decomposition are needed, perhaps for assigning subtasks to a subset of the robots. Thus, robots must coordinate in order to jointly assume responsibility for a given subtask.

Still other applications require coordination in a tight manner even after the task has been decomposed and assignments have been made. For example, consider a team of robots conducting a security operation using an expanding perimeter. The robots must sweep through an area while ensuring that all points along the perimeter can be seen by at least one robot at all times. For this problem, the team must coordinate at every step to prevent occlusions in the perimeter that would enable an adversary to enter the secured area undetected. Thus, the robots need to coordinate during execution of their subtasks.

This paper describes how to apply the market approach to multi-robot control for a variety of applications with different types and granularity of coordination. Section II describes our first formulation, Traderbots, applied to the mapping problem. Section III extends Traderbots to handle more complex problems, such as area reconnaissance, where multiple robots are needed to solve a given subtask. Section IV explores coordination in the subtask execution phase that enables tightly-coupled applications such as perimeter sweeping.

II. INDEPENDENT TASK DECOMPOSITION

Many tasks such as mapping an unknown area, distributed sensing tasks, etc., can be accomplished more efficiently if the task is decomposed into independent subtasks and executed by different robots in parallel. This coordination granularity is coarse because all required coordination occurs in the task decomposition stage and none in the task execution stage. In the TraderBots approach [5] [6], coordination of task allocation is accomplished through trading.

II.A. Auctions and Clustering

Trading is a key component of the TraderBots approach. A “RoboTrader” agent assigned to each robot is responsible for opportunistically optimizing the tasks the robot commits to executing. An “OpTrader” agent serves as an interface between the operator and the robot team. Each trader maintains a portfolio in which it keeps track of its commitments, schedule, currently executing tasks, and tasks it trades to others. Two forms of contract types are allowed during trading: subcontracts and transfers. If the contract type is a subcontract, it implies the auctioneer is interested in monitoring the progress of the task and will hence expect a report when the task is completed; payment is made only after the subcontracted task is completed. If, on the other hand, the contract type is a transfer, payment is made as soon as the task is traded, and no further communication concerning that task is necessary between the auctioneer and bidder. Each trader has an internal alarm that prompts it to auction all tasks in its schedule periodically. Note that tasks under execution are removed from the schedule and hence cannot be traded.

A trader initiates an auction by sending out a call for bids. Traders that are within communication range compute and submit bids to this auction. Once the specified deadline expires, the auctioneer resolves the call by making an allocation based on the bids it received. If a trader receives an award for a bid it submitted, it accepts or rejects that award based on its current state. Note that an award is binding after it has been accepted. Two methods of call resolution are used in the current implementation of TraderBots. The RoboTraders award at most the single most profitable bid submitted to the auction. The OpTrader uses a greedy algorithm for resolving calls so that tasks are allocated more rapidly and team execution begins. The greedy algorithm assigns the most profitable bid submitted by each trader for each task while ensuring that no task is assigned more than once and no bidder is assigned more than one task during each auction. In order to participate in an auction, robots need to calculate the costs of tasks. A robot announcing an auction must determine its reservation price, i.e. the highest price it is willing to pay to subcontract or purchase a task. A robot bidding in an auction must calculate the expected cost of the tasks being offered. These valuations are based on marginal costs – the difference in between the cost of the current schedule with those tasks and the cost of the schedule without those tasks. For a single task, an auctioneer’s valuation is the savings resulting from removing that task from its schedule. A bidder’s marginal cost for a single task is the estimated cost of inserting the task into its schedule.

Further optimization can be introduced through the development of a “leader” role that allows a robot with the necessary resources to assess the current plans of a group of robots and provide more optimal plans for the group. The leader can gain knowledge of the group’s current state via communication or some form of observation. A prospective leader can use the profits generated by an optimized plan to bid for the services of the group members, and retain a portion of the profit for itself. The leader may bid not only against the individuals’ plans, but also against group plans produced by other prospective leaders. Note there are many ways in which the leader can reduce the cost within the group (and thereby the global cost). Leaders can use different mechanisms to re-distribute tasks among the group and even generate new tasks to coordinate the group more efficiently.

The capability to negotiate multi-task deals greatly enhances the market approach because it allows a robot to escape some local minima in task allocation solutions. However, if the robots bid on every possible combination of tasks, the number of bids submitted will grow exponentially with the number of tasks. Consequently, processing these bids will be impossible for more than a few tasks. Hence, some form of a clustering algorithm is necessary to determine the groups of tasks on which to bid, and some form of auction-clearing algorithm is required to process multi-task bids. A leader can provide these capabilities to robots that cannot reason about clusters. A combinatorial exchange (a market where many bidders can jointly buy and sell a combination of goods and services within a single bid) can be another instantiation of a leader that enables multi-party optimizations for a team. A combinatorial exchange enables a leader to locally optimize the task assignments of a subgroup of robots and to potentially achieve a greater global cost reduction. Many researchers have presented valuable insight on how to efficiently implement and clear combinatorial exchanges for E-commerce applications [7].

II.B. Experimental Results

We validated the TraderBots approach using a distributed map-building application. This translates into a version of the traveling salesman problem (TSP) with the robots being represented by multiple salesmen following paths instead of tours (i.e. without the requirement that robots need to return to their starting locations) and where all the robots can start from different base locations. This is known as the multi-depot traveling salesman path problem (MD-TSPP). The tasks can be modeled as cities to be visited where the costs are computed as the time taken to drive between cities. A task is completed when a robot arrives at a city. The global task is complete when all cities are visited by at least one robot. The global cost is computed as the summation of the individual robot cost, and the goal is to complete the global task while minimizing the number of robot-hours consumed. Each robot is responsible for optimizing its own local schedule.

Figure 1: The mapping application was implemented using a team of Pioneer robots equipped with laser rangefinders. The scene depicts the area mapped during the experiments. Note that the environment is very cluttered.

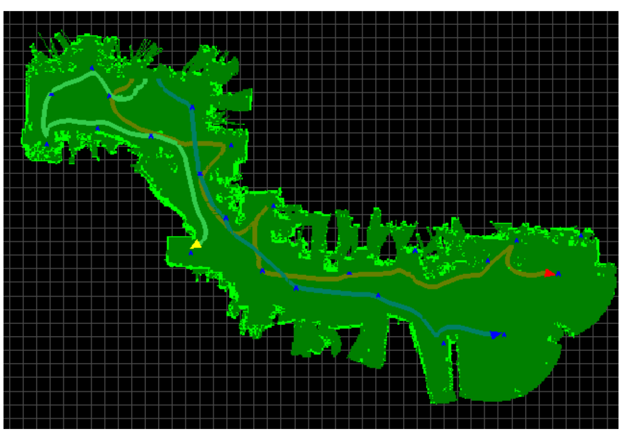


Figure 2: An overhead view of the area mapped. The dark areas are open space; the light areas are structures in the environment; the gridded areas are unsensed.

The mapping application was implemented using a team of Pioneer II-DX robots equipped with SICK laser rangefinders oriented to scan horizontally. Figure 1 shows the robots and the area mapped, which was quite cluttered. Figure 2 shows an overhead view of the area mapped. The dark areas are open space; the light areas are obstacles, walls, and other structures; the gridded areas are unsensed.

Further implementation details are reported in earlier publications ([8], [9]).

III. COORDINATED TASK DECOMPOSITION

For some problems, robots need to coordinate to decompose a task into subtasks that are executed by more than one robot. For example, in an area reconnaissance scenario, robots need to jointly cover a region composed of several named areas of interest (NAI), which may be accomplished by visiting multiple observation points (OPs). At one level of abstraction, robots can jointly execute area coverage tasks. However, at a lower level, the execution of the individual OP subtasks does not require coordination. We present a task tree-based market mechanism, in which robots coordinate both task decomposition and task allocation, to arrive at efficient solutions to complex tasks for a mid-range level of interaction.

III.A. Task Tree-based Market Mechanism

Coordinated task decomposition is achieved by participating in a task tree market [10]. Within the market, contracts are sold for executing trees of tasks which represent complex tasks at variable levels of abstraction. An example task tree is shown in Figure 3. At the root of the tree is an abstract task, and at each successive level of the tree the tasks are further refined into more primitive tasks. Constructing a task tree involves performing a task decomposition on an abstract task. The tree of Figure 3 is an AND/OR tree, meaning that subtasks are related to their parent tasks through one of the logical connectives AND or OR. If a parent node contains an AND operator, this implies that all of its children must be performed to satisfy it. An OR operator implies that at least one of the child tasks must be completed to satisfy the parent. In Figure 3, an area reconnaissance mission is decomposed into two NAIs, each of which is decomposed into two alternative plans that involve visiting multiple OPs.

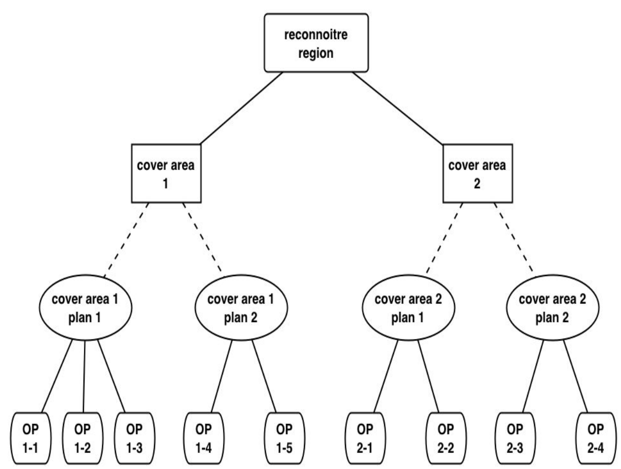


Figure 3: An example task tree for an area reconnaissance scenario with two areas. The solid edges represent AND operators, and the dashed lines represent OR operators. Note that for the cover area tasks, there are two alternative decompositions specified in the tree.

The task tree market enables a team of robots to buy and sell their roles in performing joint subtasks. A robot can offer a task tree on the market by holding an auction, and other robots can bid on parts of the tree. Robots with lower costs for specific subtrees can bid on those tasks. The auctioneer awards subcontracts for the subtrees that result in the greatest improvement of its individual cost, which translates to a decrease in global solution cost. A bid for a primitive task is simply the expected marginal cost of executing the task. A bid for an abstract task can represent one of two quantities. The first is the marginal cost of minimally satisfying the plan represented by that tree node. For an AND task, this amounts to the cost of performing all of the subtasks of that node, while for an OR task, the cost is the minimum cost of performing one of the child tasks. The second way a robot can bid for an abstract task is by computing its own decomposition of the task. If the resulting decomposition is of lower cost than the plan on offer, then the robot can use that cost as its bid. In this case, the robot is essentially offering a new plan for a task, rather than just agreeing to execute it as offered. If the robot wins the abstract task, it may use its new decomposition to execute it. A fast clearing algorithm attempts to find the minimum-cost task allocation while obeying the constraints of the tree structure. Auctions can proceed in rounds, so subtrees may be distributed among the robots in any arbitrary manner; but in each round the global solution cost decreases monotonically.

Perhaps the best way to illustrate the mechanics of the task tree market is through a simple example. Figure 4 displays a series of auctions for an abstract cover area task. At the bottom of each subfigure is a geometric representation of the task. The large square shows the area to be covered, and the labeled points are observation points from which the area can be viewed to achieve the required coverage. The edges between the tasks are labeled with the costs of navigating between the points. At the top of each subfigure is a task tree representing the current decomposition of the task, labeled with the task prices. The coverage task may be a subtask of some larger global mission. On the center-right of each image is the global cost of the current solution.

In the example, the area coverage task is initially allocated to robot *R1*. The decomposition and the plan of *R1* is shown in Figure 4(a). The total cost of this plan is $40, which can be incurred by *R1* navigating to goal point *b* followed by goal point *a*. Suppose *R1* now holds a task tree auction for the coverage task, and there is another robot, *R2*, within communications range which decides to bid on the task tree. In Figure 4(b), *R2*'s travel costs and its valuation of *R1*'s plan are shown. *R2*'s costs are higher than *R1* for all three tasks in the tree, so with this bid, *R2* would not be awarded any tasks. However, as Figure 4(c) illustrates, *R2* computes a different decomposition (with observation points *c* and *d*) and then bids on the area task based on the new plan. In this case, *R2* can complete the coverage task for a lower cost ($25), and thus is awarded the task tree by *R1* (Figure 4(d)). By holding this auction, the global solution cost has dropped from $40 to $25. Figure 4(e.) shows that the solution can be improved even further if *R2* holds another auction round, and a third robot, *R3*, wins task *c* which reduces the global solution cost to $21.

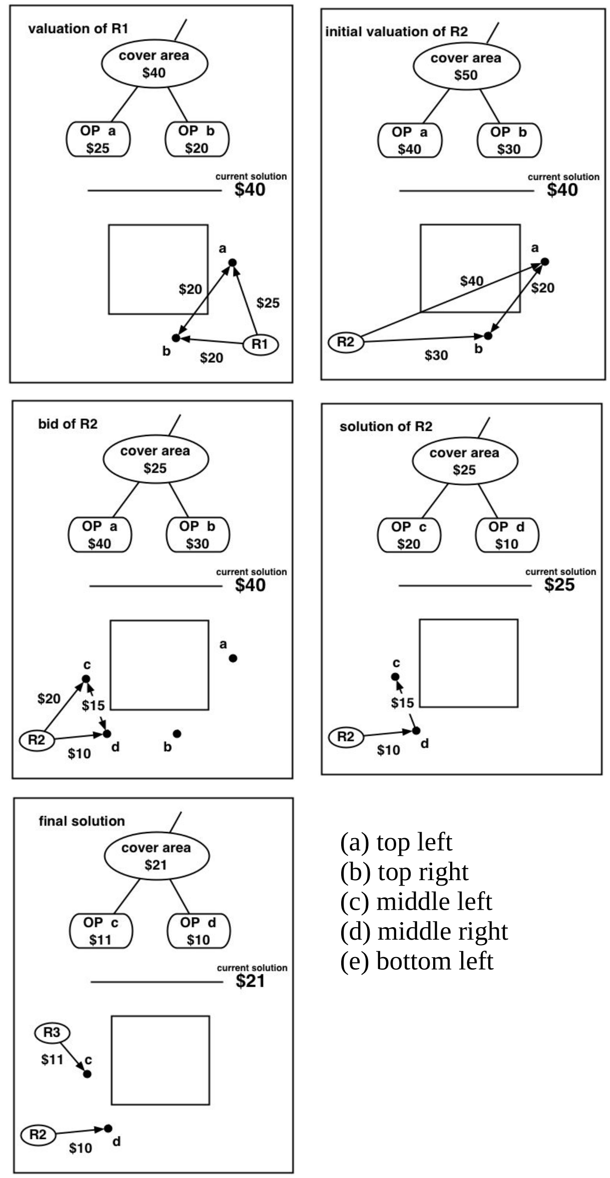


Figure 4: Simple example of a task tree auction for an area coverage task. (a) *R1* holds a task tree auction. The initial plan of *R1* is displayed along with *R1*'s reserve price for the task tree. (b) *R2*'s valuation of *R1*'s tree, without replanning. (c) *R2* comes up with a different decomposition for the cover area task, and updates its bid accordingly. (d) The auction is cleared. *R2* is awarded the abstract area coverage task. *R2*'s new plan is shown along with the associated task tree decomposition. The global solution cost has been reduced from $40 to $25. (e) *R2* holds another auction round, which results in task *c* being subcontracted out to *R3*. The global solution cost has been further reduced to $21.

III.B. Experimental Results

The task tree market has been implemented on a team of Pioneer II-DX robots [9]. Figure 5 shows a map created by a team of two robots performing an area reconnaissance task requiring coverage of three NAIs. All tasks are initially assigned to the robot on the left. An auction results in NAI 1 being subsequently subcontracted to the other robot, who then re-decomposes the task and achieves the required coverage.

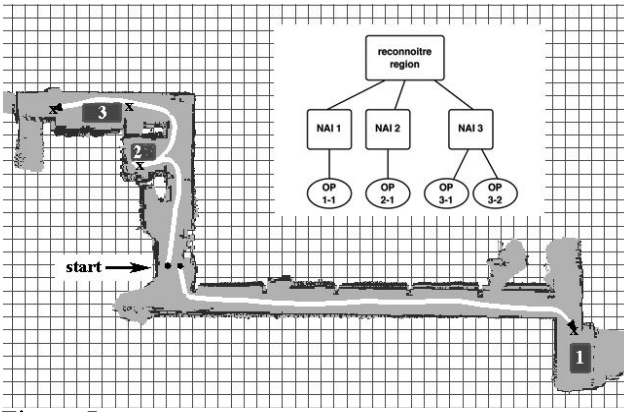


Figure 5: Map created during an area reconnaissance mission performed by two robots in an indoor hallway environment. Three areas of interest (numbered, shaded rectangles) are covered by two robots (dark triangles) by visiting observations points (marked with x's). Paths taken by the robots are shown in white. The grid cells are spaced 1m apart. Also shown in the upper right is the task tree representing the mission. The top two levels are given as input to the robots, who then generate the leaf tasks automatically using task decomposition.

IV. COORDINATED TASK EXECUTION

Some tasks require even tighter coordination, particularly during the execution stage. Consider the problem of a group of robots jointly lifting and moving heavy objects. The robots must lift an object securely and move together to prevent it from falling, while also avoiding obstacles and planning paths. In this and similar tasks, the effect of an action taken by one robot greatly depends upon the actions of other team members. To successfully complete the task, robots must coordinate at every step of task decomposition, planning, and execution, resulting in a very fine granularity of interaction [11].

IV.A. Execution Through Repeated Planning

When planning, a robot searches out to some planning horizon T for the sequence of actions that generates the most profit. Here, T is a fixed number of time steps into the future. Once it chooses a plan, it executes the plan up to some execution horizon t, where $t \leq T$. It then repeats the process, incorporating any new information gained since the last planning step.

Larger values of T mean that a robot generates plans farther into the future. Thus, because the profit expected from a plan depends on other robots' actions (which may be unknown), the uncertainty associated with the profit estimate increases. At the same time, longer plans are less likely than shorter plans to trap the robot in local minima. Larger values of t mean that a robot is more committed to its plans, reducing the uncertainty it creates in other robots' profit estimates. However, the robot also updates its plans less frequently and may be sluggish in responding to important changes in the environment. In this section, we examine the trade-off between longer and shorter horizons to find the planning strategy that works best for tight coordination.

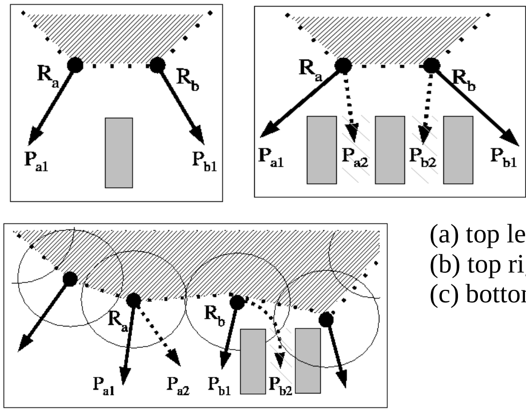
Often robots perform reasonably well by coordinating implicitly or reactively: each robot chooses the plan that appears most profitable based on some assumptions about other robots' actions. For example, a robot may assume that every plan generated to horizon T will be completely executed. Alternately, it may assume that only some portion of every plan will be executed. A robot then updates its own plan to accommodate changes in other robots' plans and actions, but it will not actively influence them.

In this scheme robots cannot guarantee each others' actions. Consequently, they may pass on high-profit plans that require commitments from other robots for less-profitable ones that are more robust to unexpected behaviors. They can overcome this difficulty by explicitly coordinating. A single robot R can solve for coordinated plans to horizon T to be carried out in lock-step by itself and other robots. If such a coordinated plan is more profitable than the best reactive plan by some margin $M1$, it may suggest this plan to other robots. If it is also more profitable for all other robots, they will mutually agree to execute this plan to horizon t. Alternately, suppose it creates a loss of $M2$ for some subset of the robots. If $M1 > M2$, then R can offer to compensate for the loss with some portion of $M1$, creating a profitable situation for all robots. Again, the robots will agree to execute this plan to horizon t.

IV.B. Experimental Results

Consider a group of robots providing an advancing security perimeter by sweeping out an area with their sensors as they move. The robots are instructed not to create any blind spots along the perimeter that would allow a foe to enter the secured area. The task is easy in an open area but more difficult when there are obstacles present that occlude the line of sight. Robots must tightly coordinate with their neighbors to breach obstacles without breaking the perimeter.

For this problem, a robot receives compensation proportional to the change in coverage its actions create. To accurately assign credit, this change is evaluated at every time step and under the assumption that other robots remain stationary over the interval. Moreover, two robots are equally penalized for every time interval that the perimeter between them is broken. The most profitable paths maximize the area gained and minimize the number of perimeter breaks.



(a) top left
(b) top right
(c) bottom

Figure 6: Successful coordination strategies in different environments. (a) Simple environment where implicit coordination is sufficient. (b) Complex environment where explicit coordination naturally results in greater profits for both robots. (c) Complex environment where explicit coordination enables one robot to buy the participation of another.

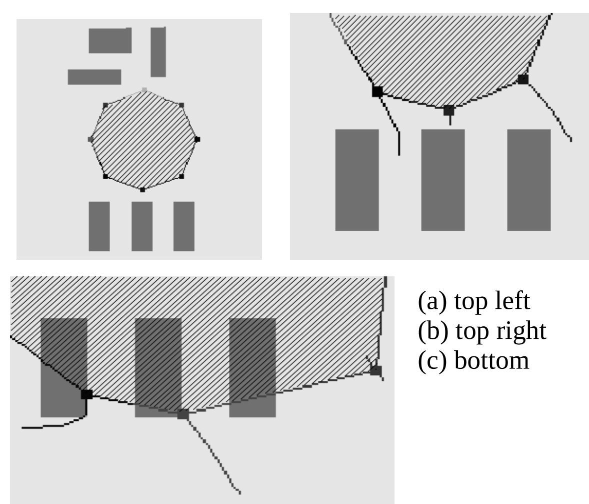
Figure 6 illustrates the differences between explicit and implicit coordination. In each figure robots *Ra* and *Rb* are depicted as black circles and the obstacles are solid gray rectangles. The perimeter between the robots and their adjacent neighbors is outlined in dotted black and the area they have secured is shaded in black. To advance the perimeter, the robots need to move down and engulf the obstacles. Robots assume that others are not reliably committed to their plans. So, they consider all the possible actions others could take when searching for the optimal path.

In Figure 6(a), robots *Ra* and *Rb* find that paths *Pa1* and *Pb1*, respectively, have the greatest expected profit. These paths are optimal because they enable the robots to breach the obstacle without breaking the perimeter while also maximizing the area secured with each step. In simple environments such as this, implicit coordination is sufficient.

Figure 6(b) presents a more complex environment. Paths *Pa1* and *Pb1* maximize the area secured for *Ra* and *Rb*, respectively. However, this arrangement enables a foe to enter the secured area undetected through the channels shaded in light gray. The perimeter remains intact only if the robots follow paths *Pa2* and *Pb2*. Consider the scenario from *Rb*'s perspective. If *Ra* follows path *Pa2*, *Rb*'s best course of action is to follow *Pb2* (it incurs a small area loss but avoids the larger cost of breaking the perimeter). However, if *Ra* follows *Pa1* and *Rb* follows *Pb2*, they will still incur a penalty for breaking the perimeter, but *Rb* will also incur the additional area loss. Consequently, without knowing *Ra*'s actions for certain, *Rb* determines that *Pb1* has the greatest expected profit and follows it. *Ra* does the same computation and follows *Pa1*, leading to a suboptimal solution. However, if either robot searches for a coordinated plan, it will discover that following *Pa2* and *Pb2* will naturally result in larger profits for both. They can then select and execute this joint plan.

In Figure 6(c), the robots are barely within their sensor range (noted by the intersecting circles) and approach the maximum area they can secure. The solid arrows show the path with the greatest expected profit for each robot. *Pb1* is the best option for *Rb* because, although it creates a break in the perimeter, *Rb*'s movement is too restricted by *Ra* to follow *Pb2*. However, if *Rb* searches for a coordinated plan, it will discover that it can follow *Pb2* if *Ra* follows *Pa2* and thus avoids the high cost of breaking the perimeter. Although *Pa2* is somewhat less profitable for *Ra*, *Rb* can offset the loss with its profits, making the arrangement more desirable for both robots.

We have implemented the task in simulation on a group of eight robots coordinating implicitly but not explicitly. The robots assume that their neighbors are fully committed to the plans they declare. Figure 7(a) illustrates the placement of the robots (squares in black), the obstacles in the environment (gray), and the area they have secured (shaded in black). The best local move for each robot is to move radially out from the center of the group. Figure 7(b) focuses on the lower three robots. Here, the optimal global move is for two adjacent robots to move between the closest pair of obstacles, thereby preventing perimeter breaks. The lines extending from the robots mark the paths they plan to take. Note that the robot on the left has chosen a plan to move between two obstacles. Although clearly not locally optimal, this plan is globally optimal. Reacting to that plan, the center robot plans the correct neighboring path. They extend these plans and, as seen in Figure 7(c), successfully breach the cluster of objects.



(a) top left
(b) top right
(c) bottom

Figure 7: Results from a simulated perimeter sweeping task involving implicit coordination. (a) Eight robots begin in an environment with six obstacles. (b) The lower three robots must find paths between the obstacles. (c) The robots successfully cover the area without breaking the perimeter.

V. CONCLUSIONS

This paper illustrates how to use market mechanisms to coordinate a multi-robot team at the task decomposition, assignment, and execution phases for map building, area reconnaissance, and perimeter sweeping applications. In the future, we will extend these mechanisms to optimize more complex applications requiring synchronization between many team members in both the assignment and execution stages.

Acknowledgments

This work was sponsored in part by the U.S. Army Research Laboratory, under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012) and by NASA, under contract "Heterogeneous Multi-Rover Coordination for Planetary Exploration" (contract number NCC2-1243). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements of the U.S. Government.

References

[1] Brumitt, B. L., Stentz, A., "Dynamic Mission Planning for Multiple Mobile Robots", Proceedings of the IEEE International Conference on Robotics and Automation, No. 3, 1996.

[2] Simmons, R., Apfelbaum, D., Fox, D., Goldman, R. P., Haigh, K. Z., Musliner, D. J., Pelican, M., and Thrun, S., "Coordinated Deployment of Multiple, Heterogeneous Robots", Proceedings of the Conference on Intelligent Robots and Systems (IROS), 2000.

[3] Arkin, R. C., Balch, T., "AuRA: Principles and Practice in Review", Journal of Experimental & Theoretical Artificial Intelligence, Vol. 9, No. 2/3, 1997.

[4] Parker, L. E., "Designing Control Laws for Cooperative Agent Teams", Proceedings of IEEE International Conference on Robotics and Automation, 1994.

[5] Dias, M. B., and Stentz, A., "A Free Market Architecture for Distributed Control of a Multi-Robot System", Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS), 2000.

[6] Dias, M. B., and Stentz, A., "A Market Approach to Multirobot Coordination", Technical Report, CMU-RI-TR-01-26, Robotics Institute, Carnegie Mellon University, 2001.

[7] Sandholm, T., and Suri, S., "Improved Algorithms for Optimal Winner Determination in Combinatorial Auctions and Generalizations", National Conference on Artificial Intelligence (AAAI), 2000.

[8] Zlot, R., Stentz, A., Dias, M. B., and Thayer, S., "Multi-Robot Exploration Controlled By A Market Economy", Proceedings of the IEEE International Conference on Robotics and Automation, 2002.

[9] Dias, M. B., Zlot, R., Zinck, M., Gonzalez, J. P., and Stentz, A., "A Versatile Implementation of the TraderBots Approach for Multirobot Coordination", Proceedings of the 2004 conference on Intelligent Autonomous Systems, 2004.

[10] Zlot, R., Stentz, A., "Market-based Multirobot Control Using Task Abstraction", Proceedings of the 2003 International Conference on Field and Service Robotics (FSR), 2003.

[11] Kalra, N., Stentz, A., "A Market Approach to Tightly-Coupled Multi-Robot Coordination: First Results", Proceedings of the ARL Collaborative Technologies Alliance Symposium, 2003.