Opportunistic Optimization for Market-Based Multirobot Control

M. Bernardine Dias and Anthony Stentz
Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, {mbdias, axs}@ri.cmu.edu

Abstract

Multirobot coordination, if made efficient and robust, promises high impact on automation. The challenge is to enable robots to work together in an intelligent manner to execute a global task. The market approach has had considerable success in the multirobot coordination domain. This paper investigates the effects of introducing opportunistic optimization with leaders to enhance market-based multirobot coordination. Leaders are able to optimize within subgroups of robots by collecting information about their tasks and status, and re-allocating the tasks within the subgroup in a more profitable manner. The presented work considers the effects of a leader optimizing a single subgroup, and some effects of multiple leaders optimizing overlapping subgroups. The implementations were tested on a variation of the distributed traveling salesman problem. Presented results show that global costs can be reduced, and hence task allocation can be improved, utilizing leaders.

1 Introduction

The growing demand for robotic solutions to increasingly complex and varied problems has dictated that a single robot is no longer the best solution for many application domains; instead, teams of robots must coordinate intelligently for successful task execution. Driven by these demands, many research efforts have focused on the challenge of multirobot coordination. Dias and Stentz [2] present a detailed description of multirobot application domains and their demands and show that robot teams are more effective than a single robot in many application domains. Simply increasing the number of robots assigned to a task does not necessarily solve a problem more efficiently; multiple robots must cooperate to achieve high efficiency. The difficulty arises in coordinating many robots to perform a complex, global task. Dynamic environments, malfunctioning robots, and multiple user requirements add to the complexity of the multirobot coordination problem. Dias and Stentz [2] explore some of these issues, and present some of the principal efforts in this field of research.

One approach is to design the team such that a single robot or central computer acts as a “leader” and is responsible for planning the actions of the entire group. The principal advantage of such centralized approaches is that they allow optimal planning. However, they suffer from several disadvantages including sluggish response to dynamic conditions, intractable solutions for large teams, communication difficulties, and the leader becoming a central point of failure.

Local and distributed approaches address these problems by distributing the planning responsibilities amongst all members of the team. Each robot operates independently, relying on its local sensor information. Many research efforts have modeled distributed systems inspired by biology, physics, and economics. The principal drawback of distributed approaches is that they often result in highly sub-optimal solutions because all plans are based solely on local information.

Recently, negotiation-based and economy/market-based multirobot coordination has gained popularity. This work in multirobot coordination draws from the software agents literature that began with Smith’s Contract Net Protocol [9], its extension by Sandholm and Lesser [7], and the general concepts of market-aware agents developed by Wellman and Wurman [12]. These concepts have since been extended to control a variety of multiagent (and more recently multirobot) systems. Golfarelli and Rizzi [5] proposed a swap-based negotiation protocol for multirobot coordination that restricted negotiations to task-swaps. Stentz and Dias [10] proposed a more capable market-based approach for multirobot coordination which aims to opportunistically introduce pockets of centralized optimal planning into a distributed system, thereby exploiting the desirable properties of both distributed and centralized approaches. Thayer et al. [11], Gerkey and Mataric [4], and Zlot et al. [13] have since produced market-based multirobot coordination results. Economic approaches are not without their disadvantages. Negotiation protocols, mapping of task domains to appropriate cost functions, and introducing relevant de-commitment penalty schemes can quickly complicate the design of a control architecture. Furthermore, some negotiation schemes can drastically increase communication requirements. Thus, all of these factors must be considered when designing a market-based architecture.

2 The Market Approach

Stentz and Dias [10] first introduced the concept of using a market approach to coordinate multiple robots to cooperatively complete a task, building on the contract net protocol by Smith [9], its extension by Sandholm and Lesser [7], and the general concepts of market-aware agents developed by Wellman and Wurman [12]. This work introduced the methodology of applying market mechanisms to intra-team robot coordination (i.e. in typically non-competitive environments) as opposed to competitive multirobot domains and competitive inter-agent interactions in domains such as E-commerce. Simulation results using this approach were produced by Dias and Stentz [3], and proven robot results were presented by Thayer et al. [11], and Zlot et al. [13]. A brief introduction to this approach is presented here.

Consider a team of robots assembled to perform a particular set of tasks. Consider further, that each robot in the team is modeled as a self-interested agent, and the team of robots as an economy. The goal of the team is to complete the tasks successfully while minimizing overall costs. Each robot aims to maximize its individual profit (which often translates to minimizing individual cost where possible); however, since all revenue is derived from satisfying team objectives, the robots' self-interest equates to doing global good. Moreover, all robots can only increase their profit by eliminating unnecessary waste (i.e. excess cost). Hence, if the global cost is determined by the summation of individual robot costs, each deal made by a robot (note that robots will only make profitable deals) will result in global cost reduction. The competitive element of the robots bidding for different tasks enables the systems to decipher the competing local information of each robot, while the currency exchange provides grounding for the competing local costs in terms of the global value of the tasks being performed.

2.1 Revenues, Costs, the Role of Price and the Bidding Process

Appropriate functions are needed to map possible task outcomes onto revenue values and to map possible schemes for performing the task onto cost values. As a team, the goal is to execute some plan such that the overall profit (the excess of revenue over cost), is maximized. Furthermore, these functions must provide a means for distributing the revenue and assessing costs to individual robots.

Thus, robots receive revenue and incur costs for accomplishing a specific team-task, but the team's revenue function is not the only source of income. A robot can also receive revenue from another robot in exchange for goods or services. The *price* dictates the payment amount for the good or service. A common approach is to *bid* for a good or service in order to arrive at a mutually acceptable price.

2.2 Cooperation, Competition, Learning and Adaptation

Two robots are *cooperative* if they have complementary roles; that is, if both robots can make more profit by working together than by working individually. Conversely, two robots are *competitive* if they have the same role; that is, if the amount of profit that one can make is negatively affected by the presence of the other robot. The flexibility of the market-model allows the robots to cooperate and compete as necessary to accomplish a task.

Moreover, the robot economy is amenable to learning new behaviors and strategies as it executes its complex global task. An added strength of the market approach is its ability to deal opportunistically with dynamic environments.

2.3 Self Organization

Conspicuously absent from the market approach is a rigid, top-down hierarchy. Instead, the robots organize themselves in a way that is mutually beneficial. Since the aggregate profit amassed by the individuals is directly tied to the success of the task, this self-organization yields the best results.

Consider a group of ten robots. An eleventh robot, **A**, offers its services as their leader. It does not become their leader by coercion or decree, but by convincing the group that they will make more profit by following its advice than by acting individually or in subgroups. **A** does this by investigating "plans" for utilizing all ten robots. If **A** comes up with a truly good plan, it will maximize profit across the whole group. The prospective leader can use this large profit to bid for the services of the group members, and of course, retain a portion of the profit for itself. Note that all relevant robots will have to commit to the plan before the plan can be sold. The leader may be bidding not only against the individuals' plans, but also against group plans produced by other prospective leaders. Note that the leader acts both as a benevolent and a self-interested agent since it receives personal compensation for efforts benefiting the entire group.

But there is a limit to this organization. As the group becomes larger, the combinatorics become intractable and the process of gathering all of the relevant information to produce a good plan becomes increasingly difficult. A leader will realize this when it can no longer convince its subjects (via bidding for their services) to follow its plans.

3 Contribution

The work presented in this paper explores some effects of opportunistic optimization with leaders in market-based multirobot coordination. Furthermore, this work addressed one of the key limitations of our implementation of this approach thus far: the restriction of negotiations to single-party, single-task deals. In many cases, this restriction limits the global cost reduction, since the robots do not have the negotiation tools to reason their way out of shallow, local minima. The work presented here extends these tools to permit multi-party and multi-task deals with better global cost reduction potential.

4 Optimizing with Leaders

An important contribution of this work is 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. Centralized and distributed approaches are two extremes along a continuum. The introduction of leaders allows the market-based approach to slide along this continuum in the direction of improved profitability in an opportunistic manner. In this work we implement a preliminary version of the leader capability by means of a combinatorial exchange, as proposed in [2].

4.1 Clustering for Multi-Task Processing

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 clustering algorithm is necessary to determine the clusters of tasks to bid on. The possibilities for such clustering algorithms are numerous [6].

The clustering algorithm used in this work is chosen to ensure a span in size (from single-task clusters to a wholly inclusive cluster) and task membership (i.e. ensure that every task is included in at least one cluster). These properties are important because a robot cannot necessarily predict the interaction of the clusters it offers with the tasks of other bidders, and hence, needs to give the allocator ample flexibility in offloading tasks. The chosen clustering algorithm operates as follows:

1. Create a list of edges spanning all tasks on offer (N), where each edge joins two tasks and the cost of the edge represents the distance in cost space between the two tasks. A low edge value implies, but does not guarantee, that two tasks can be performed more cost-effectively together than apart.
2. Sort the edge list from lowest to highest cost.
3. Form the first group of clusters by creating a single-task cluster for each task on offer.
4. For cluster sizes ranging from 2 to N, recursively form new clusters by adding the next best available edge (an edge is unavailable if it is either already included in a previous cluster or if the edge connects two tasks which are not included in any of the previous clusters) to a cluster in the previous cluster-list. (Note, when new clusters are formed, all previous clusters are preserved). Thus, recursively form a forest of minimum spanning trees (MSTs) [1] ranging in size from 1 to N.

This algorithm can be applied in general to determine which tasks are best dealt with in clusters, without computing every possible cluster. Suitable variations of this algorithm (or others) can be chosen to enable multi-task negotiations in different task domains. The presented work is verified on a multi-depot distributed traveling salesman problem (TSP), and hence, the MSTs are decomposed into tours as follows. If a newly added edge breaks the continuity of the tour, the MST is adjusted by removing one of the edges connecting to the newly added edge and adding the necessary edge to preserve the continuity of the tour with the least addition to the cost of the tour. Note that this change still preserves the bounds of the MST, which guarantees that the cost of the tour does not exceed twice the optimal cost. This holds true for metric cost spaces where the triangle inequality is preserved.

Allowing robots to include the offloading of an owned cluster when bidding to accept a new cluster of tasks further enhances the bidding capability of the robots.

4.2 Combinatorial Exchange for Multi-Party Optimizations

A combinatorial exchange (a market where bidders can jointly buy and sell a combination of goods and services within a single bid) is chosen to enable 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 including Sandholm and Suri [8] have presented valuable insight on how to efficiently implement and clear combinatorial exchanges for E-commerce applications. However, many of these tools are relatively complex and are not used in this work for simplicity. Instead, the basic recommendation of searching a binary bid tree is applied. The chosen implementation for clearing the combinatorial exchange in this work is a depth first search on a binary tree where each node of the tree represents a bid and the binary aspect of the tree represents accepting or rejecting that bid. The tree is pruned to disallow accepting multiple bids from any single bidder, and to disallow exchanging of any single task more than once. Note that the pruning does not affect the solution except by improving the runtime.

The preliminary version of the leader role in the market approach is implemented as follows. A leader queries surrounding robots to discover what tasks they have to offer and their current states, and re-allocates tasks within the group using the combinatorial exchange mechanism. Note that this is just one way in which the leader can reduce the cost within the group (and thereby the global cost). Other schemes could involve the leader using different mechanisms to re-distribute tasks and even generating new tasks to coordinate the group more efficiently. Moreover, some tasks (for example, cooperative automated construction and cooperative maneuvering of large objects) may require tight coordination where a leader has to closely monitor the progress of individual team members and accordingly direct the efforts of other members of the team.

4.3 Competing Local Groups

When leaders are allowed to opportunistically optimize sub-groups, occasions could arise where two leaders are in competition for the services of the robots that overlap between the two groups. If a robot bids on tasks from both leaders, it could win both bids and be unable to perform them or find it unprofitable to do so. There are several ways to address this “synchronization” issue. For example, broken deals with a penalty can be allowed, or bids can be stamped with an expiration time during which they are valid and offers can be dealt with on a first-come-first-serve or last-come-first-serve basis.

In the work presented here, the groups are allowed to negotiate in round robin fashion, thus forcing serial synchronization.

5 Experimentation

The proposed multi-task and multi-party enhancements are developed and tested in a simulated distributed sensing task. A group of robots, located at different starting positions in a known simulated world, are assigned the task of visiting a set of pre-selected observation points. This problem is a variation of the multi-depot distributed traveling salesman problem, where the observation points are the cities to visit. Note that many multirobot application domains require an effective solution to the distributed traveling salesman problem. The costs are the lengths of the straight-line paths between locations, interpreted as money. Let $\mathbf{c_{ij}}$ be the cost for the $\mathbf{j}^{th}$ robot to visit the $\mathbf{i}^{th}$ city from the ($\mathbf{i\text{-}1}^{th}$) city in its tour (where the $\mathbf{0}^{th}$ city is the starting location).

The robot cost function for the $\mathbf{j}^{th}$ robot is computed as follows:

$$\mathbf{rcost}(\mathbf{j}) = \sum_{\mathbf{i}=1}^{\mathbf{n_j}} \mathbf{c_{ij}}$$

where $\mathbf{n_j}$ is the number of cities in the tour for robot **j**.

The team cost function is:

$$\mathbf{tcost} = \sum_{\mathbf{j}=1}^{\mathbf{m}} \mathbf{rcost}(\mathbf{j})$$

where **m** is the number of robots.

The team revenue and robot revenue functions are determined by the negotiated prices. All robots (bidders) adopt the same simplistic strategy of bidding a fixed 10% markup above the cost of completing the task. According to this strategy, if an announced task costs **c** to execute, a robot computes its bid as **1.1 c**. Thus, the robots bid for each city based on their estimated costs to visit that city. Similarly, if a robot offers up a task that will cost it **c** to execute, in an attempt to buy the services of another robot to complete that task, the maximum price it offers for this service is set as **0.9 c**.

Tasks and robot positions are randomly generated within a 100x100 world, and initial task allocations are made by randomly distributing the tasks among the robots. Heterogeneous robot capabilities are considered by restricting some robots' capabilities such that they can only process single-task (ST) deals, while other robots can process multi-task (MT) deals. Robots capable of playing leader roles are allowed the additional capability of performing multi-party (MP) optimizations via either a single-goods exchange or a combinatorial exchange, depending on their capability. Sections 5.1 through 5.4 describe in detail the scenarios of robots negotiating in the absence of a leader (TPST and TPMT) and the optimization scenarios with leaders (MPST and MPMT). Section 5.5 describes the scenario where robots have limited communication range and hence can only trade within subgroups.

5.1 Two-Party, Single-Task (TPST) Negotiations

In this case, once the initial random task assignments are made, each of the robots, *in turn*, offers all its assigned tasks to all the other robots, *in turn*. Thus, interactions are limited to two parties at any given time as illustrated in **Figure 1**.

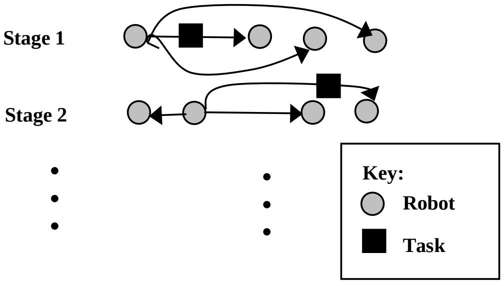

Figure 1: *TPST Illustration*

Each bidder then submits a bid for each task. In order to estimate the additional cost of inserting a task into its queue, the bidder uses the cluster generation algorithm described above to generate an MST with its current queue of tasks plus the offered task, and computes the cost difference between the resulting and original queues. The offerer accepts the most profitable bid it receives. The cost of the offerer's resulting queue is computed by removing from its queue the task that was transferred through the winning bid, clustering the remaining tasks using the clustering algorithm, and computing the cost of the resulting queue. Hence, in the TPST scenario, only single-task (ST) deals are considered, and pairs of robots continue to negotiate amongst themselves in round-robin fashion until no new, mutually profitable deals are possible. Therefore, negotiations cease once the system settles into a local minimum for the global cost function.

5.2 Two-Party, Multi-Task (TPMT) Negotiations

In this case, the previous case is repeated with clusters of tasks being the atomic unit of the negotiations as shown in **Figure 2**.

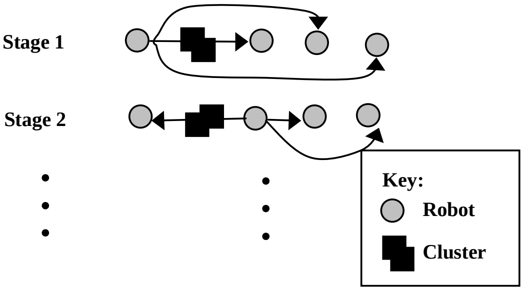

Figure 2: *TPMT Illustration*

That is, the initial assignments are followed by each of the robots, *in turn*, offering all of its assigned tasks to all the other robots, *in turn*. The robots then bid for clusters of these tasks. Once again, costs are computed by using the clustering algorithm to cluster all tasks under consideration and compute the cost of the resulting queues, and negotiations are always between two robots.

5.3 Leader Performing Multi-Party Single-Task (MPST) Optimizations

A leader, whose capability is restricted to dealing in single-task deals, is introduced in this case. The leader queries all the robots, and gathers all the tasks of all the robots along with each robot's state information. The leader then sets up an exchange by formulating single-task bids for the robots in the sub-group based on the gathered information. The exchange used in the MPST scenario is a single-task exchange (i.e. a single bid can contain buying of a single task and selling of another single task). The exchange is then cleared to maximize the leader's profit. These interactions are illustrated in **Figure 3**.

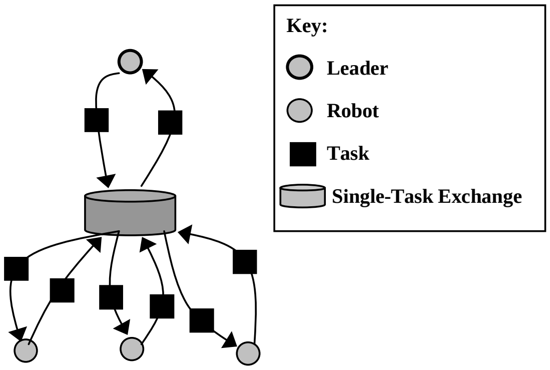

Figure 3: *MPST Illustration*

This process is repeated until the exchange cannot produce any further profit, and the corresponding task re-allocation is proposed to the sub-group of robots. If the leader's plan reduces the global cost, the resulting excess profit can be distributed among the entire subgroup (including the leader) such that the robots in the subgroup accept the leader's task re-allocation.

5.4 Leader Performing Multi-Party, Multi-Task (MPMT) Optimizations

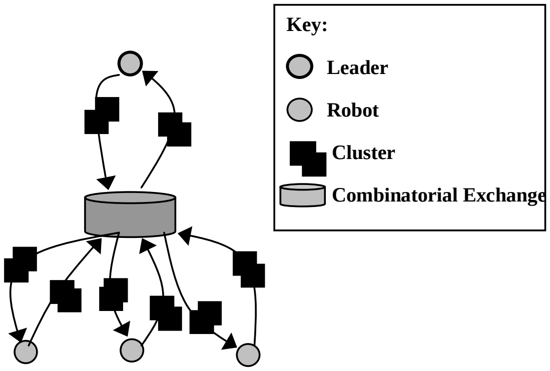

Figure 4: *MPMT Illustration*

Here, the previous case was repeated with the added capability of the leader to process MT bids as shown in **Figure 4**. That is, the leader sets up and clears a combinatorial exchange to determine the re-allocation of tasks. In a combinatorial exchange, clusters of tasks can be bought and sold within a single bid.

5.5 Multiple Competing Local Groups

This set of experiments involved 8 robots divided into 3 groups of 4 robots each (with the middle group overlapping the other two groups) and 10 tasks. Trading and optimization with leaders are restricted to within the subgroups. The robots are evenly spread throughout a 2000x2000 world and the cities (tasks) are randomly generated. Scenarios with and without leaders, and with ST-capable and MT-capable robots are considered.

6 Results and Discussion

The results for the experiments described above are shown below. **Figure 5** and **Figure 6** show the final tours of each robot for a 2-robot, 10-city TSP and a 4-robot, 10-city TSP respectively. In both figures, the robots are shown as circles and the cities are shown as squares.

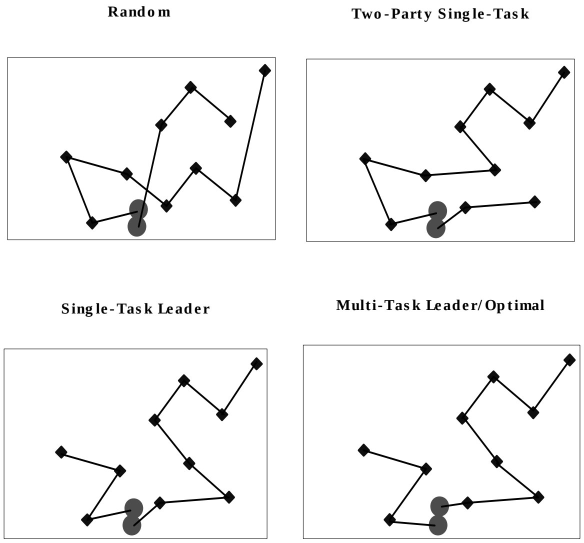

Figure 5: *Solutions to a 2-robot, 10-task TSP with and without leader-optimization*

	Cost	Itns	Improved	Opt. Error
Random	351	-	0.0 %	65.6 %
No Leader				
2 ST	256	2	25.9 %	21.4 %
2 MT	231	1	33.0 %	9.0 %
ST Leader	245	2	29.0 %	16.2 %
MT Leader	227	1	34.4 %	7.0 %
Optimal	212	-	38.6 %	0.0 %

Table 1: *Performance averaged over 100 randomly generated 2-robot, 10-task TSPs*

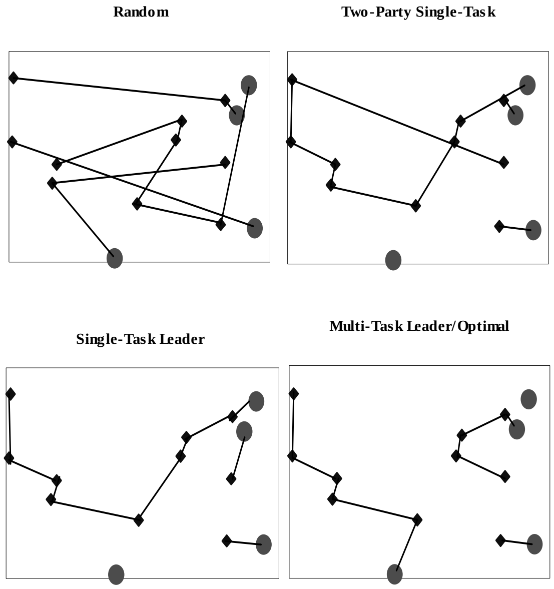

Figure 6: *Solutions to a 4-robot, 10-task TSP with and without leader-optimization*

The first illustration in each figure shows the tours after the initial random allocation of tasks. The second illustration shows the resulting tours after the robots have completed TPST deals and reached a local minimum in global cost. The third and fourth illustrations show the results of the MPST and MPMT scenarios. In the illustrated cases, the optimal allocation is reached in the MPMT scenario.

	Cost	Itns	Improved	Opt. Error
Random	411	-	0.0 %	124.6 %
No Leader				
4 ST	230	5	42.7 %	27.7 %
2ST+2MT	222	5	44.6 %	23.3 %
1ST+3MT	209	4	47.8 %	16.2 %
4MT	197	4	50.9 %	9.7 %
ST Leader	218	3	45.8 %	21.1 %
MT Leader	193	2	51.8 %	7.5 %
Optimal	183	-	-	0.0 %

Table 2: *Results averaged over 100 randomly generated 4-robot (heterogeneous), 10-task TSPs*

Table 1, **Table 2**, and **Table 3** report the performance averaged over 100 randomly generated task distributions for the 2-robot-10-task case, the 4-robot-10-task case, and the 4-robot-20-task case respectively. As evident from these results, on average, an MT-capable leader can improve the profit of the group significantly. An ST-capable leader can only improve the profit of the group on average for groups of robots where there are at most 50% MT-capable robots.

	Cost	Iterations	Improved
Random	725	-	0.0%
No Leader			
4 ST	400	10	44.1%
2ST+2MT	388	9	45.7%
1ST+3MT	359	7	49.8%
4MT	336	5	53.0%
ST Leader	373	6	47.7%
MT Leader	322	3	54.9%

Table 3: *Performance averaged over 100 randomly generated 4-robot (heterogeneous), 20-task TSPs*

Figure 7 and **Table 4** illustrate preliminary results for the competing subgroup scenario. The subgroups of robots are circled in **Figure 7**, which depicts the results of a single run. **Table 4** reports the performance averaged over 100 randomly generated task distributions. Again, the results show that on average the local optimization with leaders improves the global profit.

	Cost	Iterations	Improved
Random	9091	-	0.0%
No Leader			
4 ST	4598	8	48.9%
2ST+2MT	4379	9	51.2%
ST Leader	4312	6	52.1%
MT Leader	3687	6	58.9%

Table 4*: Performance averaged over 100 randomly generated 8-robot (heterogeneous), 10-task TSPs with 3 overlapping groups of 4 robots each*

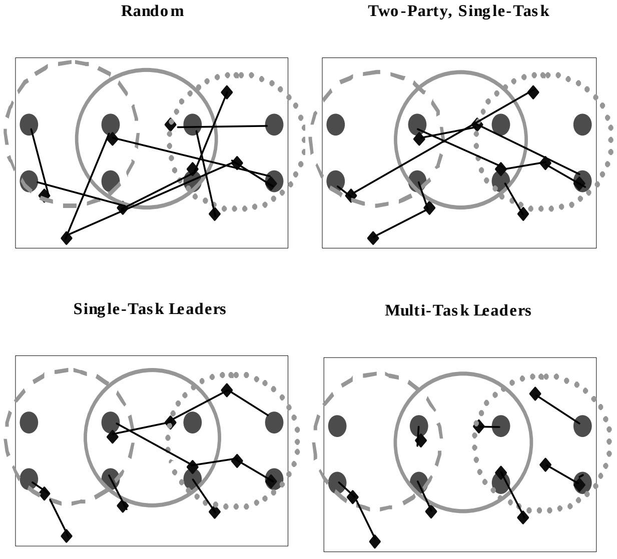

Figure 7: *Solution for TSP with 3 overlapping subgroups of 4 robots each and 10 tasks*

The presented work only addresses scenarios where leaders run exchanges to optimize task allocation within a group of robots. Some leaders are also capable of clustering tasks and hence can conduct combinatorial exchanges. It is also possible to have combinatorial exchanges and leaders as distinct entities within the economy. For example, there could be a leader that simply clusters tasks and sells these *cluster plans* to a combinatorial exchange. Note that the leader is not selling the *actual* cluster of tasks—just a plan for which tasks to cluster. The exchange could then buy all of the component tasks, sell off the resultant cluster, and pay a fee to the leader. The presented results indicate that the benefit from the ability to cluster tasks and participate in multi-task negotiations exceeds the benefit from the ability to perform multi-party negotiations. Leaders could also use other approaches to generate plans for a subgroup of robots. Finally, a leader could simply act as a means of enabling trade between subgroups of robots who are otherwise unable to communicate, thus enriching the possible trades.

7 Conclusions and Future Work

Presented results show that leaders can considerably reduce global costs in market-based multirobot coordination. Initial experiments for optimizing within robot sub-groups with leaders also proved promising. Future work includes implementing these capabilities on a robot team and further extensions of the market approach. Proposed enhancements include more detailed analysis of optimizing with leaders, dealing with time constraints, and experimentation with different task domains. The goal of this work is to produce an efficient and robust market-based multirobot coordination architecture.

8 Acknowledgements

This research was sponsored in part by NASA, under contract "Heterogeneous Multi-Rover Coordination for Planetary Exploration" (contract number NCC2-1243). This research was also sponsored in part by DARPA, under contract “Cognitive Colonies” (contract number N66001-99-1-8921, monitored by SPAWAR). 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, either expressed or implied, of the U.S. Government. The authors thank the members of the Cognitive Colonies group: Scott Thayer, Bruce Digney, Martial Hebert, Robert Zlot, Aaron Morris, Charles Smart, and Bart Nabbe and wish to acknowledge the contributions of the FIRE project team: Reid Simmons, Trey Smith, Dani Goldberg, Steve Smith, Jeff Schneider, Drew Bagnell, Vincent Cicirello, David Apfelbaum, and Stuart Anderson.

9 References

[1] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. eds. 1990. *Introduction To Algorithms*. Cambridge, MA: The MIT Press/McGraw-Hill Book Company.

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

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

[4] Gerkey, B. P. and Matarić, M. J. Submitted 2001. *Sold! Market methods for multi-robot control*. IEEE Transactions on Robotics and Automation Special Issue on Multi-Robot Systems.

[5] Golfarelli, M., Maio, and D., Rizzi, S. 1997. *A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm*. Technical Report, 005-97, CSITE (Research Centre For Informatics And Telecommunication Systems), University of Bologna.

[6] Papadimitriou, C. H., and Steiglitz, K. eds. 1998. *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY: Dover Publications, Inc.

[7] Sandholm, T., and Lesser, V., *Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework*, Proceedings of the first International Conference on Multiagent Systems (ICMAS-95), 1995.

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

[9] Smith, R. 1980. *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*. IEEE Transactions on Computers Vol. C-29, No. 12.

[10] Stentz, A. and Dias, M. B. 1999. *A Free Market Architecture for Coordinating Multiple Robots*. Technical Report, CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University.

[11] Thayer, S. M., Dias, M. B., Digney, B. L., Stentz, A., Nabbe, B., and Hebert, M. 2000. *Distributed robotic mapping of extreme environments*. Proceedings of SPIE: Vol. 4195: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII.

[12] Wellman, M. P., and Wurman, P. R., *Market-Aware Agents for a Multiagent World*, Robotics and Autonomous Systems, Volume 24, 1998.

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