2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

A Generalized Framework for Solving Tightly-coupled Multirobot Planning Problems

Nidhi Kalra, Dave Ferguson, and Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania
{nkalra,dif,tony}@cmu.edu

Abstract— **In this paper, we present the generalized version of the Hoplites coordination framework designed to efficiently solve complex, tightly-coupled multirobot planning problems. Our extensions greatly increase the flexibility with which teammates can both plan and coordinate with each other; consequently, we can apply Hoplites to a wider range of domains and plan coordination between robots more efficiently. We apply our framework to the constrained exploration domain and compare Hoplites in simulation to competing distributed and centralized approaches. Our results demonstrate that Hoplites significantly outperforms both approaches in terms of the quality of solutions produced while remaining computationally competitive with much simpler approaches. We further demonstrate features such as scalability and validate our approach with field results from a team of large autonomous vehicles performing constrained exploration in an outdoor environment.**

I. INTRODUCTION

Many practical multirobot applications such as collaborative manipulation and moving in formation require robot to tightly coordinate during execution. In such tasks, the actions and state of one robot often constrain the actions available to its teammates; thus, robots must work closely together to complete the mission. We are interested in solving very complex types of these problems: those with tight-coordination requirements that cannot be met using hill-climbing techniques and that instead necessitate teammates to plan their interactions significantly in advance of execution.

Our motivating problem, constrained exploration (CE), is one such application [1]. In CE we task a team of robots with exploring a possibly hazardous environment. For safety, information collection and dissemination, and mission tasking, we require each robot to maintain communication contact with some communication antenna, either directly or indirectly via other teammates that act as relay nodes. Essentially, the team is constrained to maintain a connected ad-hoc network during exploration. Because of this constraint, each robot must continuously consider the positions and actions of its teammates when choosing its own. The problem is particularly challenging because robots must tightly-coordinate their future paths to generate valid solutions [2].

As we describe in Section III, current approaches to tight-coordination fail when applied to problems that also require long-term planning. To achieve these tasks, we have developed Hoplites, a market-based coordination framework that negotiates the limitations of both distributed and centralized approaches and couples the best features of both. Hoplites dynamically adapts to the changing demands of the task: it employs distributed planning when deliberation time is scarce or when problem scenarios are easy, and relies on more centralized planning when ample time is available or when faced with difficult scenarios that require extensive coordination. As we show in this paper, by selectively injecting pockets of complexity, Hoplites provides higher-quality solutions than both competing distributed and centralized approaches while remaining computationally competitive with simpler distributed approaches.

Our initial work on Hoplites led to a system able to solve a simplified version of CE known as 'perimeter sweeping' [2]. Our current work makes important extensions to Hoplites that enable it to solve a much wider range of problems. Previously, robots were limited to coordinating with a small, predefined set of teammates; now robots can coordinate with any of their teammates or groups of teammates, which allows the search of a much larger space and leads to better, more flexible solutions. Furthermore, now robots can demonstrably use any planning algorithm to achieve coordination and can also use different planners in different stages of coordination. We describe these improvements in Section IV. In particular, as we describe in Section V, we incorporate into Hoplites four very different general planning approaches that address different aspects of the tight-coordination challenge.

This paper also makes a number of experimental contributions. In Section VI, we compare Hoplites in simulation along a number of metrics to powerful competing approaches. Our results show that Hoplites is responsive to the level of mission difficulty, achieves flexible coordination among different team members, scales very well to large teams, and outperforms its competitors in terms of solution quality and is computationally competitive with simpler, faster approaches. Lastly, We verify our approach in Section VII with field results from a team of large autonomous outdoor vehicles performing CE. Further details of the work presented here and additional experiments can be found in Kalra's Ph.D. thesis [11].

II. CONSTRAINED EXPLORATION

In our motivating problem of Constrained Exploration (CE), a team of robots is tasked with exploring a hazardous

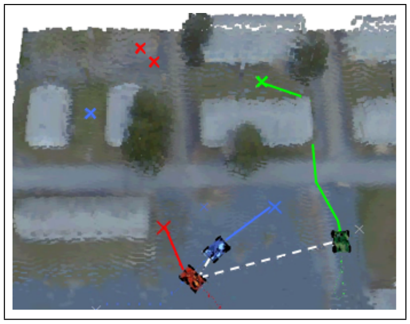


Fig. 1. An example of Constrained Exploration from our simulator with three robots exploring an outdoor environment with trees and buildings. Notice that the right-most robot will have to coordinate closely with its teammates to reach its goal behind the building.

environment while maintaining communication constraints between the team members. CE is a special case of the Multi-Depot Traveling Salesman Problem (in which a set of salesmen must collectively solve the traveling salesman problem) where the cost of traveling to a city depends on the length of the path to the city *and* on the communication connectivity available along the path. CE is a useful domain for developing multirobot approaches because, by varying the connectivity requirements, we can map it to a number of other complex multirobot domains including pursuit evasion, perimeter sweeping, gallery monitoring, and other vehicle routing problems [1]. As shown in Figure 1, for this paper we use an instance of CE in which the team must maintain an ad-hoc connected communication network. Direct communication between any two teammates is possible if and only if there is a clear line of sight between them.

III. Related Work

As we presented in our original Hoplites paper [2], in general, distributed approaches to tight-coordination use local hill-climbing techniques such as reactive or behavior based algorithms to achieve fast, responsive coordination between teammates. However, because these approaches are limited to simple, prescriptive rules, they cannot provide the long-term, adaptive planning required by the problems in our problem space. For example, Nguyen et al. [3] use teammate-following behaviors to solve simple CE problems with a single teleoperated robot and a team of dedicated relay nodes. In Wagner and Arkin's approach [4], robots explore around obstacles using predefined behavioral plans chosen according to the team size and the obstacle shape. This enumerated approach is also not adaptable to arbitrary and complex instances of CE.

Centralized approaches, on the other hand, work well for small teams or simple instances since they can plan the coordination of the entire team at once. Schouwenaars et al. [5] use centralized mixed integer linear programming; however, to ensure tractability, they impose a fixed ordering on the team structure which limits solution quality, fails in complex instances of CE, and is inapplicable to general problems where flexibility is required. In contrast, Ferguson et. al. [6] demonstrate that sampling-based methods can flexibly solve instances for small teams, and Kalra et. al. [7] show how planners such as A* can be coupled with roadmaps to enable the efficient search of a reduced planning space. Like all centralized approaches, however, these do not scale to very large teams or very complex problems, and they also suffer from single points of failure. Nevertheless, as we discuss in Section IV they are useful components to distributed systems and are used as planners in Hoplites.

IV. The Generalized Hoplites Framework

A. Intuition

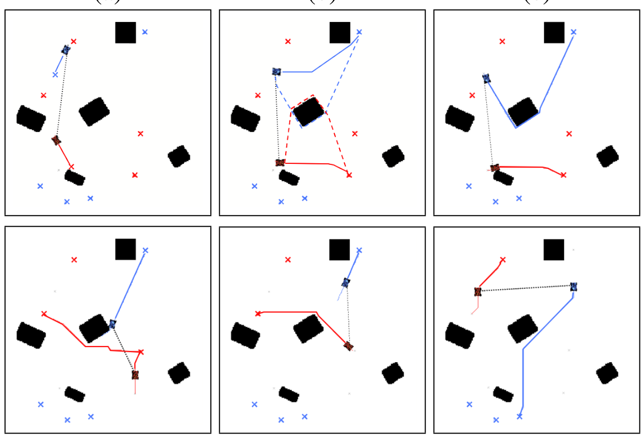
The difficulty of CE and similar problems varies greatly between different instances of the problem and even within a single instance of the problem. For example, in a sparsely-populated environment, robots can navigate freely much of the time because their line of sight with teammates will not be obstructed. Thus, simple coordination should still produce a good solution. In a complex environment, however, a robot will need to coordinate closely with one or many of its teammates to ensure its connectivity. This coordination may require more computation and more communication to produce a good solution.

The motivation behind Hoplites is that the complexity of the coordination should be responsive to the changing demands of the task. Hoplites achieves flexible coordination by enabling robots to decouple their planning and execution whenever possible using a *passive* coordination method and to dynamically form pockets of centralized planning when necessary using an *active* coordination method. This allows solutions to be built in an efficient, bottom-up manner rather than in a more expensive, top-down manner.

B. Framework Components

Hoplites is market-based, meaning that robots act as self-interested agents in a virtual economy in which they receive revenue for contributing to the team mission and incur cost by consuming team resources such as energy. Robots buy and sell mission components and team resources to maximize their own profit, defined as revenue minus cost; this redistribution simultaneously produces efficient team solutions [8]. Market-based approaches are widely applicable and have been shown to be robust, flexible, responsive, and fast.

By incorporating two novel features, Hoplites extends market-based approaches for the first time to problems involving tight coordination. First, when a robot violates team constraints (e.g. line-of-sight connectivity for CE) it incurs a penalty which reduces the profit of its actions. Because these constraints depend on teammates' simultaneous actions, this penalty couples robots' actions and encourages them to closely coordinate. Secondly, Hoplites enables robots to buy and sell tightly-coupled action-level plans over the market; with this *active* coordination, robots can influence each other's actions and directly coordinate to find cost-effective solutions even in complex scenarios. This is one of the primary contribution of the Hoplites framework.



(a) (b) (c) (d) (e) (f)

Fig. 2. Snapshots from a simulation involving two robots and a series of sequential goals to be visited by each robot. (a) The direct path to the robots' first cities causes no line-of-sight violations, and they use passive coordination. (b) The path to the second cities creates line-of-sight violations. (c) The robots actively coordinate: the top robot has accepted a contract by the bottom robot to take a path under the center obstacle (this path is generated by a randomized planner). (d) After reaching its second goal, the bottom robot's shortest path to its next goals will cause line-of-sight violations. (e) The bottom robot solves the problem by moving above the center obstacle (this path is generated by A*) (f) The top robot's final path will cause the line-of-sight to be broken. It accepts a contract to wait at its current position and allow its teammate to complete its mission successfully.

We illustrate these features in the context of the CE example in Figure 2 in which two robots must each visit a set of goals in an environment with five obstacles while always maintaining line-of-sight communication. Robots use passive coordination as a first-attempt strategy in which they continuously select the most profitable plan given their teammates' simultaneous actions and broadcast this plan back to the team. In Figure 2(a), robots independently plan direct paths to their first goals; these paths are the most profitable since they consume minimum team resources (they are the shortest) and they do not violate team constraints (the environment is clear). Thus, passive coordination allows them to solve this intermediate problem with independent planning and minimal computation and coordination. In Section VI, we demonstrate that passive coordination is very effective, particularly in simple environments.

However, as shown in Figure 2(b), independently-planned direct paths to the second set of goals violate the line-of-sight constraints. Two valid solutions are possible: either robot can travel to the far side of the center obstacle (dashed) while the other takes a direct path (solid). However, passive coordination may not find either solution: with an optimistic planner, each robot may assume the other will take the long route so neither will; with a pessimistic planner, each may assume the other will *not* take the long route and so both will; or, they may oscillate between these options [2]. Even if passive coordination finds a valid solution, it may not be the least-cost solution. Finally, in more complex scenarios, both robots may need to change their paths simultaneously which cannot be achieved using passive coordination. In sum, the team is likely to fall into a local minima.

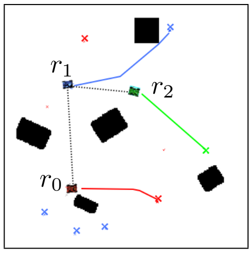
Active coordination is designed to escape such local minima. When a robot anticipates a constraint violation based on the published paths of its teammates, it tries to develop a joint, tightly-coupled solution with its teammates and induces them to commit to this solution by offsetting their marginal costs for participation. Firstly, Hoplites can make use of centralized planners mentioned in Section III to find good joint plans. Secondly, by comparing several solutions and costs, it can find the most cost-effective solution for itself (and thus retain the most wealth); simultaneously, this leads to a cost-effective team-wide solution. It also enables robots to guarantee each others actions and avoid oscillations. In Figure 2(c), the bottom robot compares both solutions and finds that the best (shorter) solution is for the top robot to travel around the obstacle, and it pays its teammate for its assistance in executing this solution.

C. Extensions

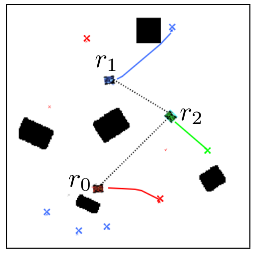
Since our original Hoplites publication [2], we have extended the framework in a number of significant ways to produce a general approach that can solve a wide range of tasks involving tight coordination.

Firstly, in the original perimeter sweeping domain, we imposed a fixed ordering on constraints, so that each robot had to maintain contact with the same two adjacent neighbors. As in work by Schouwenaars et al. [5], this reduced the complexity of the problem but also made it difficult to effectively solve domains such as constrained exploration, where robots may need to coordinate with any teammate at any time. To illustrate, in Figure 3(a), we add a third robot to the same environment in Figure 2. In our previous implementation (and in Schouwenaars et. al.), the chain of communication links (and thus the team structure) would be fixed to $r_0 - r_1 - r_2$, and r_0 and r_2 would never coordinate to maintain team connectivity. This would force r_0 and r_1 to coordinate as in Figure 2(b) to move around the center obstacle. As shown in Figure 3(b), however, a flexible team structure would allow r_2 to act as a link between r_0 and r_1 and result in a better solution. We have incorporated this flexibility into Hoplites: robots consider the paths of any teammates for which they have recent information and can coordinate with them passively or actively at any time.

Secondly, robots previously used simple discrete planners for passive coordination (since it is a simple coordination method) and complex sampling-based planners for active coordination (since it is a complex coordination method). However, this unnecessarily couples the planning algorithms to the coordination algorithm, reduces flexibility, and is in contrast to Hoplites' philosophy of adaptable coordination. We have extended Hoplites to provide robots with a planning toolbox that is separate from the coordination mechanism. Thus, they can choose planners according to the difficulty of the coordination scenario rather than the complexity of the mechanism. Moreover, when a simple planner fails to find



(a)



(b)

Fig. 3. Snapshots of three robots solving CE using a flexible team structure.

a solution, they can use a more complex planner and thus attack problems with increasingly sophisticated algorithms.

The simulation snapshots in Figure 2 demonstrate how four different planning algorithms can be used in the same experiment to provide different solutions. In Figure 2(a), the robots use A* to find direct paths; this is the simplest algorithm and provides very efficient solutions in simple scenarios. In Figure 2(c), r_1 uses the most complex planner, an RRT, to find a feasible path for actively coordinating with r_0 when simpler planners cannot find a solution. In Figure 2(e), r_0 uses a complex domain-specific discrete planner to find an independent path when active coordination with r_1 fails. Finally, in Figure 2(f) r_1 actively coordinates with r_0 by using a reduced-dimensionality planner to synchronize its path to r_0's without changing its route. We describe these planning algorithms in the following section. As illustrated in this example, a large planning toolbox increases the chances of finding a solution, while also encouraging the use of less expensive planners whenever possible and the use of complex planners when necessary.

Both of these extensions generalize Hoplites to a much wider range of applications, including those which may require coordination between different teammates and subgroups of teammates at different times and which may require a variety planning algorithms in all phases of coordination.

V. Planning for Tight Coordination

To solve tight coordination problems, it is necessary to have effective planning algorithms for generating tightly-coupled plans. We have incorporated four general planning approaches that fall into two categories of techniques. An example of each can be seen in the single experiment highlighted by Figure 2.

A. Planning with Relaxed Constraints

Our first two approaches use relaxed constraints: they ignore constraints between robots, solve the remaining problem optimally, and then check the solution against the full problem with constraints. Planning with relaxed constraints is useful because the planning can often be accomplished with very fast and simple algorithms. The first of these approaches uses A* to plan a direct path from a robot's initial position to its goal position. The second uses domain-specific information to plan around the far side of an obstacle that obstructs the line of sight between two robots (Figure 2(e)).

B. Coupled planning

In difficult problem scenarios, we must incorporate the constraints directly into the planning process in order to find effective solutions. This typically means centrally planning the actions of a subset of the team which quickly becomes intractable because it has complexity exponential in the number of teammates. Our third algorithm overcomes this problem by reducing the search space with prioritized planning: rather than planning in all three dimensions x, y, and t simultaneously for each robot, it plans over x and y first and then coordinates the robots in t. The drawback to removing part of the search space is that we are not guaranteed to find the best solution or indeed any solution at all. Our fourth algorithm plans in the full configuration space of the problem. Tractability is achieved by using sampling-based methods (specifically, an RRT [9]) that do not need to generate and plan over a discrete representation and that can explore even high-dimensional spaces very quickly. The drawback is that they cannot effectively trade off different cost factors and so are unable to provide guarantees of solution quality.

VI. Simulation Experiments

We have run a large set of experiments in simulation to compare Hoplites to competing approaches along a number of dimensions.

A. Problem Setup

We tested our approach in a graphical simulation using two to ten simulated robots tasked with line-of-sight CE. For each team size we randomly generated forty environments of size 200×200 cells which each contain five occluding obstacles that range in size from 5×5 cells to 30×30 cells. Figure 2(a) shows a typical environment.

Each robot must visit four preallocated cities while maintaining communication constraints with its teammates; once it has visited its cities, it can drop out of this constraint. For these experiments, we give robots perfect information about the environment; nevertheless, they must respond to their teammates' actions as they are planned. Robots move at a speed of 5 cells per second. In the distributed approaches, each robot ran as a separate software agent and communicated with its teammates via UDP. Each approach uses a planning horizon of ten seconds, corresponding to roughly 50 steps of path lookahead. We ran our experiments on a 1.5 GHz PowerPC G4 with 512 MB of memory.

B. Metrics

We evaluate solutions along a number of dimensions. First we measure solution cost by the function C which combines the energy consumed by the team in visiting the sites and by the team's ability to satisfy connectivity constraints:

$$C = T + \sum_{r_i \in \mathrm{R}} [dist(r_i) + 100 \cdot disconn(r_i)] \qquad (1)$$

where T is the mission time, $disconn(i)$ is the number of seconds for which robot r_i is directly disconnected from the

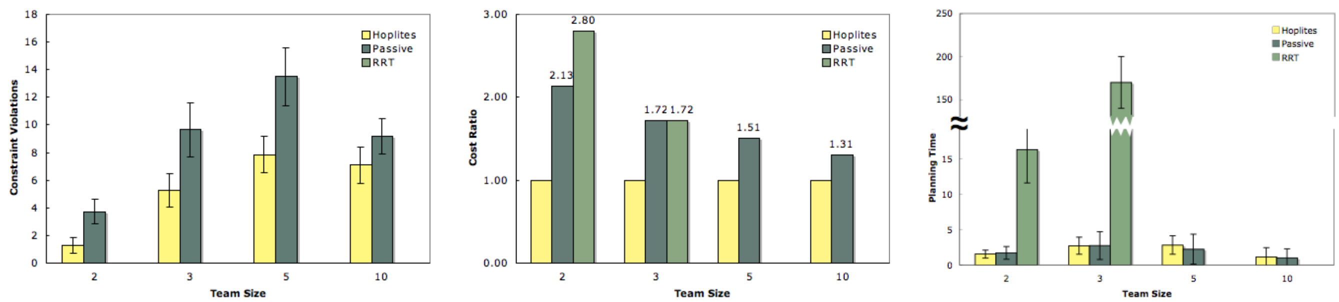


Fig. 4. The mean total disconnection time in seconds (left), the overall solution cost normalized to Hoplites's performance (center), and the mean planning time in seconds (right). Results in each graph are categorized by team size and error bars reflect the standard error of the mean.

rest of the team, and $dist(r_i)$ is the total distance traveled by r_i. The time and distance components reflect energy usage. The disconnection component is scaled by 100 to indicate that maintaining connectivity is significantly more important that completing the mission quickly. We also evaluate the cost of producing solutions in terms of the amount of time spent planning.

C. Frameworks

In each approach, the team begins as a chain of robots in which robot r_n must maintain line of sight with r_{n-1}, where r_n is known as the downlink of the pair and r_{n-1} is known as the uplink. The team is connected if all robots are able to communicate with their uplink. However, unlike other approaches to CE, we make this ordering flexible: any robot can change its uplink provided that doing so does not disconnect the team. In a centralized approach, this is easily detected because all information is known. In the distributed approaches, robots track the route by which packets travel from them to the first robot r_0 on the team, with which they all must be able to communicate. If a robot r_j is not used in the route of packets from some teammate r_i to r_0 , then r_j can switch to using r_i as an uplink and the team will still remain connected. In this way, any robot can coordinate with any other robot to satisfy the team constraints.

1) Hoplites: The Hoplites approach uses all four planners described in section V. A robot begins by trying to maintain contact with its uplink using passive coordination. If this fails (i.e. the robot needs its teammates' active assistance to stay in contact), it searches for an alternative uplink to avoid active coordination, which may be expensive. If that, too, fails, then it attempts actively coordination with its uplink to resolve constraints: the robot shares its current path with its uplink, request assistance to its next goal, and requests a price for that solution. The robot also computes a reserve price that reflects the maximum the robot is willing to pay its teammate for a solution; the reserve price is the cost of its best alternative solution. If the uplink can provide a solution at a competitive cost, it is committed to that solution and cannot accept other offers of coordination if they conflict with this one. Our results show that even this simple implementation of active coordination is effective (many more complex ones are possible).

2) Passive coordination: We have already shown that Hoplites significantly outperforms even the leading approach to problems in our domain [2]. Our aim now is to demonstrate the unique advantages of active coordination (one of the primary contributions of Hoplites) for problems that require flexible coordination. To this end, we compare Hoplites to passive coordination which was originally based on the MVERT coordination framework [10]). Passive coordination has every resource available to Hoplites, including the ability to use multiple planners and long term planning.

3) RRTs: We are also interested in evaluating the impact of *selectively* injecting complexity into the approach. In sum, these comparisons would allow us to answer the question, What is the impact of adaptive coordination? To this end, we have compared Hoplites to a centralized approach that essentially does active coordination among all of the robots all of the time. We use a multirobot version of the RRT algorithm that is tailored to the problem of CE.

D. Results and Discussion

Our results are shown in the three plots in Figure 4, categorized by approach and by team size. The left figure shows the average number of seconds for which the team was disconnected and demonstrates that by directly coordinating robots, Hoplites satisfies the team constraints better than passive coordination. Not surprisingly, Hoplites cannot solve violation constraints as well as the RRT approach (for teams of two and three robots, there are no constraint violations) because the latter plans for every teammate at once. However, as shown by the right-most figure which measures planning time, RRTs cannot plan in real time, and we were unable to solve instances for teams of five or more robots using RRTs within the three minute time limit we imposed per run.

Nevertheless, to compare overall solutions, we must evaluate ability to balance competing needs as defined by Equation 1. The ratio of costs is shown in the center graph. That Hoplites significantly outperforms passive coordination clearly verifies that active coordination improves solutions by enabling robots to vet a greater variety of solutions to the constraint satisfaction problem over larger portions of the team. Moreover, Hoplites outperforms RRTs for small team sizes even though it incurs more constraint violations. The drawback to RRTs which we observe here is that, to achieve tractability, they are not optimized on other factors

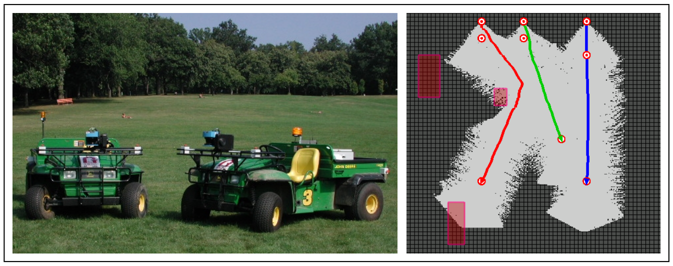


Fig. 5. (left) Two of the three E-Gators used for field testing. (right) The results of a sample CE mission. Targets represent the vehicles' initial and goal positions, shaded rectangles represent communication obstacles, dark ares indicate unknown terrain, and white areas represent traversable terrain detected by the robots' lasers. Notice that the leftmost vehicle takes a path to the right of the center obstacle as a result of a contract made with the center vehicle.

(i.e. time and distance), so constraint-satisfying solutions can be extremely expensive overall. Hoplites uses RRTs but, because it selectively injects this complexity, it pays the price of these algorithms only when absolutely necessary.

Moreover, Hoplites does this in real time and by consuming planning resources that are essentially the same as passive coordination. This is because Hoplites consumes more planning time but less frequently because it finds solutions using active coordination. Meanwhile, passive coordination consumes less planning time but has to replan more frequently as robots must often try (in vain) to meet constraints without the active assistance of their teammates. Finally, Hoplites scales well to large teams even as it provides better solutions than its competing approaches because it operates in a highly-efficient bottom-up fashion that gives robots every opportunity to find good solutions in a distributed way.

VII. Experiments on Robotic Platforms

We have also implemented Hoplites on a team of three John Deere E-Gator robotic platforms used for constrained exploration in outdoor environments. These vehicles are equipped with IMUs and GPS for position estimation, laser range finders for terrain observation, and wireless Ethernet for communication. Two of these platforms can be seen on the left in Figure 5. In this example, we gave each robot two to three goal cities in a 60×80 meter environment; we provided a prior map of virtual communication obstacles but no information about navigation obstacles (e.g. rocks). The robots generated initial paths for the vehicles and updated these paths as new information was received. The right part of Figure 5 shows the results of this traverse. Here, a communication obstacle between the left and center robots required that they carefully coordinate to avoid losing line-of-sight contact. These two robots negotiated between two competing joint plans, determined that the best solution was for the leftmost vehicle to alter its course, formed a contract, and successfully executed the plan.

VIII. Conclusion and Future Work

We have presented the general Hoplites coordination framework for solving complex, tightly-coordinated multirobot planning problems. Although planning for tightly-coupled multirobot systems is a difficult problem, Hoplites solves this problem efficiently by using distributed decision-making whenever possible and centralized planning as required. Our work in this paper significantly extends the initial development of the Hoplites framework to allow for more flexible coordination between teammates and the exploitation of a range of different planning algorithms.

Our simulation experiments show that the resulting technique significantly outperforms competing distributed and centralized approaches as it is able to dynamically adjust the quality of the solutions produced and the amount of computation used in response to the difficulty of the problem. In sum, Hoplites is better able to trade off the team's resources with the team's constraints and mission goals to efficiently provide the most cost-effective solutions. Lastly, further details of the work presented here and additional experiments can be found in Kalra's Ph.D. thesis [11].

IX. Acknowledgments

This work was sponsored by the U.S. Army Research Laboratory, under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012). 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. Dave Ferguson is supported in part by a National Science Foundation Graduate Research Fellowship.

References

[1] N. Kalra, D. Ferguson, and A. Stentz, "Constrained Exploration for Studies in Multirobot Coordination," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2006.

[2] ——, "Hoplites: A market-based framework for complex tight coordination in multi-robot teams," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2005.

[3] H. G. Nguyen, N. Pezeshkian, A. Gupta, and N. Farrington, "Maintaining communication link for a robot operating in a hazardous environment," in *Proceedings on the American Nuclear Society's Conference on Robotics for Remote and Hazardous Environments*, March 2004.

[4] A. Wagner and R. Arkin, "Multi-robot communication-sensitive reconnaissance," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, April 2004.

[5] T. Schouwenaars, A. Stubbs, J. Paduano, and E. Feron, "Multivehicle path planning for nonline-of-sight communication," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 269–290, 2006.

[6] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2006.

[7] N. Kalra, D. Ferguson, and A. T. Stentz, "Incremental reconstruction of generalized voronoi diagrams on grids," in *Proceedings of Intelligent Autonomous Systems*, March 2006.

[8] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *IEEE Special Issue on Multirobot Systems*, vol. 94, no. 7, pp. 1257–1270, July 2006.

[9] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[10] A. Stroupe, "Collaborative execution of exploration and tracking using move value estimation for robot teams (MVERT)," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 2003.

[11] N. Kalra, "A market-based framework for tightly-coupled planned coordination in multirobot teams," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2006.