Hoplites: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams*

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

***Abstract*— In this paper we address tasks for multirobot teams that require solving a distributed multi-agent planning problem in which the actions of robots are tightly coupled. The uncertainty inherent in these tasks also necessitates persistent tight coordination between teammates throughout execution. Existing approaches to coordination cannot adequately meet the technical demands of such tasks. In response, we have developed a market-based framework, Hoplites, that consists of two novel coordination mechanisms. Passive coordination quickly produces locally-developed solutions while active coordination produces complex team solutions via negotiation between teammates. Robots use the market to efficiently vet candidate solutions and to choose the coordination mechanism that best matches the current demands of the task. In experiments, Hoplites significantly outperforms even its nearest competitors, particularly in the most complex instances of a domain. We also present implementation results on a team of mobile robots.**

I. Introduction

Researchers frequently employ multirobot teams to complete tasks that cannot be accomplished by a single robot, such as carrying heavy objects or moving in formation. These tasks usually require a very high degree of coordination between team members throughout execution because the actions taken by one team member significantly constrain the actions available to others. For instance, when two robots carry a box, they are constrained to remain within a certain distance of each other so they do not drop it. Because of uncertainty in the environment, sensors, and actuation, each robot must frequently reevaluate its actions within the context of its teammate's simultaneous actions. We call this close coupling between robots *tight coordination*.

Although robots must tightly coordinate, the responses they have to each other's activities are typically simple, consistent, and require little planning because the constraints imposed by these tasks are fairly simple. When moving in formation, a robot knows a priori the position it ought to maintain relative to its teammates. So, when its teammate moves, the robot can easily compute its own heading and velocity. When carrying a box, robots can adopt leader-follower roles and use a similar technique.

*The authors would like to thank Marc Zinck and Boris Sofman for their contributions to the robotic experiments. 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.

In this paper, we address tasks that have significantly more complex constraints between robots and, consequently, require significantly more complex coordination between team members. Consider a team of robots tasked with exploring a hazardous environment. To minimize robot loss, we impose the constraint that no teammate may be stranded without communication access to some base station. A robot would have to plan an exploratory path that has at least one link with the base station at all times, perhaps with the assistance of its teammates. In cluttered environments, teammates may need to traverse complex paths simultaneously to provide communication service to all members of the group. Accomplishing this task within the imposed constraints can require extensive planning of future interactions between teammates.

The gallery monitoring and security sweep domains also require such complex coordination. In the former, robots monitor a complex environment to ensure that certain exhibit wings are continuously observed while others are periodically checked for intruders. The task may also require that robots handle on-line monitoring requests, for instance, if suspicious activity has been detected in an area. The robots must plan coordinated monitoring sequences that satisfy the constraints of the task. In the security sweep domain, robots sweep an area suspected to contain mobile adversaries. The robots must determine and execute in concert those paths through the environment that maximize the likelihood of detecting all adversaries.

Accomplishing these tasks requires a coordination mechanism that provides:

- **tight coordination:** We say that robot A coordinates with robot B if it considers the state of B when selecting its own actions. Further, we say that this coordination is *tight* if A considers B's state at a high frequency throughout execution.
- **planned coordination:** We say that a robot A plans its coordination with robot B if at some time t it anticipates the interactions it will have with B at a later time t'.

As we discuss in Section II, no existing system addresses these requirements sufficiently. We have developed a new market-based framework that facilitates the sophisticated coordination that is necessary between team members. We have named our framework "Hoplites" after the ancient Greek infantrymen who specialized in tightly-coordinated

maneuvers. Hoplites couples planning with two coordination mechanisms: one emphasizes speed while the other emphasizes solution quality. Robots use the market to vet candidate solutions and to employ the coordination mechanism that best fits the immediate needs of the task. We describe Hoplites in detail in Section III. We have applied Hoplites to the security sweep domain, which we describe in Section IV, and we extensively compare its performance to those of existing coordination frameworks in Section V. Our results demonstrate that Hoplites significantly outperforms its competitors. We further validate Hoplites on a team of mobile robots. We conclude in Section VII with a discussion of future work.

II. Related Work

For brevity, we focus our attention on research that deals specifically with planned and tight coordination in multirobot teams. A comprehensive review of related work can be found in our earlier technical report [1].

Emergent coordination strategies are most common for facilitating tight coordination in multirobot teams. They are based on the idea that coordinated group activity can successfully emerge from a collection of simple but carefully-tailored individual activities. Accordingly, coordination in robot teams is a byproduct of robots following a set of rules that map sensor data and robot states to particular actions or behaviors. Robots can choose actions very quickly and with minimal computation since the rules are simple. Tight coordination is possible because a robot can respond quickly and in a prescribed manner to the actions of its teammates. By the same token, coordination is limited to interactions that can be simply expressed, making such approaches insufficient in domains that require advanced planning of complex interactions between robots.

Emergent approaches to tight coordination are prevalent for formation control in multirobot teams [2], [3] where robots maintain the relative headings and distances explicitly stated in the formation definition. They have been used for simple box pushing tasks [4], quality of service tasks [5], and loosely-constrained exploration tasks [6].

Planned coordination is most often provided by *intentional* approaches in which robots communicate and coordinate their efforts explicitly. By sharing resources and information, robots consider each other's contributions to the task and plan better team solutions. Almost always, however, planned coordination is limited to allocating and scheduling discrete, partially-ordered subtasks in which tight coordination is absent [7], [8] or restricted to unplanned, simple interactions [9]. These approaches cannot be readily used in domains such as security sweeping that do not permit discrete task decomposition.

As an exception, Gerkey et. al. [10] use a stochastic hill-climbing approach to plan actions for very small groups of robots. The centralized planning for small teams is analogous to the team plan subcomponent of Hoplites which we discuss in section III-C. Unlike Hoplites, their approach is not scalable to large teams. Firstly, it requires perfect information about teammates and the sharing of large data structures which is not realistic to the demands of many application domains. Secondly, since tight coordination in their approach can only occur through centrally developed plans, tightly coordinating more than a few robots becomes intractable.

Of all the prior work, we believe that Stroupe's MVERT [11] is the most promising coordination framework for accomplishing tasks like security sweeping and constrained exploration. MVERT is a behavior-based framework in which robots choose their next actions based on the expected next actions of their teammates. Thus, MVERT facilitates planned coordination for very small time-steps. Like most behavior-based approaches, MVERT is fast, fault tolerant, and can operate under uncertainty. It also has virtually no communication needs and can also be applied to non-decomposable tasks. However, MVERT's one-step lookahead causes robots to behave myopically in tasks that require long-term planning. We discuss MVERT further in Section V-B.

We have found no system in the literature of multirobot coordination that adequately provides both planned and tight coordination as required by the constrained exploration, security sweep, and gallery monitoring domains. We believe that Hoplites fills this gap.

III. Hoplites

The difficulty of the aforementioned domains varies between instances of the domain. For example, a cluttered environment is more challenging than an obstacle-free environment that permits full network connectivity and visibility. Also, by having redundancy in both visibility and connectivity, the same task is easier for a team of fifty robots than a team of five robots. The difficulty of a task can also vary within a single instance of the domain, for example, if some parts of the environment are more complex than others.

The philosophy behind Hoplites is that team members should adapt their coordination strategy to the changing demands of the task. In simple situations, the team can work faster if team members make decisions more locally and coordinate via a mechanism that is light on both communication and computation. However, harder scenarios may require more complex interactions between teammates which, in turn, require a more complex coordination mechanism. Hoplites is a market framework that consists of "passive" and "active" coordination mechanisms for easier and harder problem scenarios, respectively. When passively coordinating, robots quickly react to each other's actions and influence each other implicitly. When actively coordinating, robots try to influence each other's actions explicitly by buying and selling complex team plans over the market.

A. Market Frameworks

In market frameworks [12], robots model an economy in which individuals act out of self interest. Each task a robot completes generates for it some revenue, which reflects how the task affects the team's proximity to the goal. It

also requires some cost expenditure from the robot, which reflects the amount of team resources the robot consumed to complete the task, such as fuel or network bandwidth. Robots trade tasks through auctions and negotiations to win the tasks that generate the greatest profit, defined as revenue minus cost. This exchange of tasks simultaneously results in lower cost solutions for the team.

Market frameworks have all the benefits of distributed approaches including robustness, speed, and flexibility [13], [14], [15]. Resources permitting, they can also be more centralized to produce better solutions [16]: a robot can plan a more cost-effective task distribution for part of the team, bid on the tasks, and use the cost savings between the two distributions to purchase team members' participation. This distribution is profitable for that robot and also produces a better team solution.

B. Passive Coordination

In Hoplites, robots act in a self-interested manner; however, rather than selecting the most profitable tasks, they select the most profitable plans. A robot first generates a set $P_{candidate}$ of plans that are possible in the environment. It then estimates the profitability of each plan $p_i \in P_{candidate}$ in the context of its teammates' actions and the environment using functions $R : A \rightarrow \Re$ and $C : A \rightarrow \Re^+$ that map actions to revenue and cost, respectively. Once the robot has chosen its most profitable plan p_{max}, it broadcasts p_{max} to its teammates. Its teammates use this information to reevaluate the expected profitability of their current plans, update the plans, and broadcast these changes back. In this way robots' planning cycles iteratively incorporate the most recent information about teammates' intentions. As we demonstrate in Section V, passive coordination alone can be effective in many environments where the correct actions are obvious to all robots.

C. Active Coordination

In difficult environments, a good solution may require that teammates simultaneously perform complex coordinated actions. These actions are often *high-risk*. That is, if executed by all teammates, the actions will be very profitable. However, with only partial participation, the actions can be unprofitable. In contrast, *low-risk* actions will reap some rewards for a robot independent of it teammates' actions. When passively coordinating, a robot cannot directly influence or guarantee its teammates' actions. Without a commitment strategy, passively coordinating robots pass up high-risk actions for low-risk ones that are less profitable and correspondingly suboptimal, trapping the team in local minima.

The active coordination mechanism can be used to escape such local minima and is related to the idea of "leaders" developed by Dias and Stentz [16]. When a robot's best individual plan p_{max} is only marginally profitable, it develops a team plan p_{team} that consists of actions its teammates could take that would make its own actions more profitable. It then requests price quotes $q_1 ... q_n$ from its teammates that reflect the financial compensation they require before they will abandon their existing plans and adopt the proposed team plan. If the marginal benefit of the team plan ($Profit(p_{team}) - Profit(p_{max})$) is greater than the cost of the team plan ($q_1 + ... + q_n$), then the robot bids for its teammates' participation. In this way the benefits of the proposed plan are explicitly weighed against the costs. If the teammates accept, they are bound by contract to complete their portions. Once finished, the team members revert to a passive strategy and continue on their own. Profitable team plans minimize team cost by redistributing motivations and resources over larger portions of the team and guide the team towards more effective solutions.

D. Framework Details

Several aspects of Hoplites deserve further explanation. We briefly discuss a few key points here and have provided a comprehensive discussion in our technical report [1].

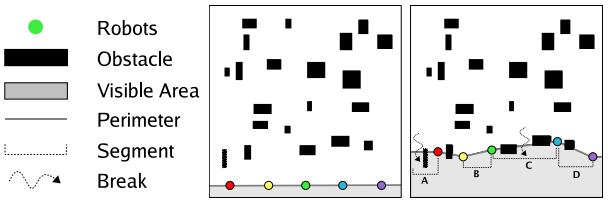
1) Choosing Coordination Strategies: We want a robot to expend time generating a team plan only when one is likely to result in a better, more profitable solution. Unlike resource consumption, which is typically independent of other robots' actions, constraint violations must often be avoided through active coordination. Consequently, a robot passively coordinates until it expects that its best plan will be less profitable specifically due to constraint violations; then, it actively coordinates with its teammates.

2) Planning: In order to develop a team plan, a planner must tractably search joint action spaces. The Hoplites framework does not specify a particular algorithm; the right choice depends upon the domain, the number of robots, and the importance of optimality. To plan in high-dimensional spaces, we recommend and use algorithms that sacrifice optimality for speed such as Probabilistic Roadmaps (PRMs) [17] and Rapidly Exploring Random Trees (RRTs) [18].

3) Commitments Between Teammates: We want a robot to be committed enough to fulfill the agreements made with other robots, while still having the flexibility to abandon its commitments if new information indicates that they will be less beneficial to the task than anticipated. We require that robots are bound to a team plan except when they can negotiate breach-of-contract terms. These terms are reparations a robot must pay its teammates before it will be released from its commitments; they can include opportunity and replacement costs incurred by the teammates. In this way, the value of an existing commitment is explicitly weighed against the value of other opportunities; the most profitable option is adopted.

IV. The Security Sweep Domain

We have chosen to demonstrate Hoplites in a security sweep domain in which robots with a limited sensor range sweep a cluttered environment for mobile adversaries. The aim is to sweep the area as quickly as possible, moving from a start location to a goal location, while detecting any adversaries that could be hiding in the environment.



(a) (b)

Fig. 1. An illustration of the security sweep task. *(a)* A sample room to be swept. *(b)* Secure (B and D) and broken (A and C) perimeter segments.

This task has obvious applications in the fields of security, surveillance, and reconnaissance.

The security sweep domain is related to the domain of pursuit evasion, in which one or more *pursuers* must find all the *evaders* that are moving in an environment [19]. Nearly all of the work on pursuit evasion is aimed at developing geometric algorithms for different instances of the domain and does not address dynamic coordination in multirobot teams. We provide a review of this work in our technical report [1].

Figure 1*(a)* is a sample room of dimensions $X_{max} \times Y_{max}$ containing twenty obstacles. The robots (marked as circles) begin in a line at the bottom of the room ($y = 0$) and have no prior knowledge of the number of obstacles or their locations. Their task is to sweep the area with their sensors as they move to the top of the room ($y = Y_{max}$). The line of sight between the robots forms the forward perimeter of the sweep. The area behind this line (shaded in light gray) has already been swept while the area ahead of it is unswept. Robots must move in a way that minimizes the blind spots along this line that would allow a foe to enter undetected into the swept area. We refer to these blind spots as *breaks* in the perimeter. Robots must coordinate to avoid breaking the perimeter as they traverse highly cluttered environments.

We restrict the adversaries to the open space; this is comparable to a team of robots sweeping for vehicles among a cluster of buildings. We also assume that obstacles are rectangular and oriented along the x and y axes. This makes it easier for robots to detect and model obstacles. Complex obstacles such as arbitrary polygons make the mechanics of the sweep more complex but do not increase the complexity of the coordination required.

Figure 1*(b)* illustrates the definition of a perimeter break. Segments A and C are broken because they intersect obstacles that would shield an adversary from view, enabling it to enter the secured area undetected. Segments B and D are not broken because all of the free space along the segments is observable by at least one robot.

A. Revenue and Cost

A robot receives revenue proportional to its progress across the environment. With n robots sweeping an environment of dimensions $X_{max} \times Y_{max}$, each robot is responsible for sweeping an imaginary corridor along the y-axis of width X_{max}/n. Consequently, each step in the $+y$ direction generates revenue proportional to X_{max}/n, giving preference to direct routes from $y = 0$ to $y = Y_{max}$. A robot incurs costs proportional to the amount of time it takes and the distance it travels to complete the task. This encourages the robot to complete the task in a timely manner and discourages it from taking winding paths. A robot is only penalized for breaks in the two segments adjacent to it; it incurs a penalty for each step that breaks a previously-secure segment or fails to repair a previously-broken segment. A relatively high penalty encourages robots to expend both time and energy to take paths that keep the perimeter intact. A lower penalty encourages robots to traverse the environment quickly, permitting breaks when they are difficult to avoid. We would prefer the former in a security mission and the latter in a stealthy reconnaissance mission.

B. Illustration of Hoplites

Hoplites significantly improves the team's performance in the security sweep domain by facilitating active tight coordination between several robots. Consider a team of four robots performing a very thorough sweep. In the simple environment in Figure 2*(a)*, each robot initially selects a straight path (P_{*1}) as its best action, marked by solid arrows. When a robot receives its teammates' intended trajectories, it still finds that its current forward path is most profitable. In one step, the team converges to the optimal plan using only passive coordination.

However, with the addition of a second obstacle (Figure 2*(b)*), this approach produces a suboptimal solution. The local environments of B, C, and D have not changed, so they still initially choose the same straight path as before. However, A must now choose between following the dotted path P_{A2} or continuing with path P_{A1}. Both options have the same penalties for A in terms of perimeter breaks. If A chooses P_{A1}, B will not face a penalty and will continue following P_{B1}. If A chooses P_{A2}, B will have to make a choice between P_{B1} and P_{B2}. Now, B will be forced to choose P_{B2} to keep the perimeter intact between itself and A and avoid a large penalty. Indeed, the optimal solution is for A and B to follow P_{A2} and P_{B2}, respectively.

Unfortunately, using a passive coordination strategy, A chooses P_{A1} over P_{A2}. Although P_{A1} and P_{A2} have the same number of breaks in the perimeter, P_{A2} has larger time and distance costs, so it is overall more costly than P_{A1}. Even if A determines that following P_{A2} might cause B to follow P_{B2}, it cannot guarantee B's actions. So, A chooses the low-risk path P_{A1} over the high-risk path P_{A2}, resulting in a suboptimal solution for the team.

The active coordination mechanism produces the optimal solution. During planning, A finds that it will incur a minimum cost of C_a due to penalties from perimeter breaks. It searches for a team plan P_{AB} that would decrease this cost ($P_{AB} = \{P_{A2}, P_{B2}\}$). A then requests a price quote Q_b from B for P_{AB}. This quote reflects the additional cost C_b to B for taking P_{B2} instead of its default path P_{B1}. We expect that, since the penalty for perimeter breaks is higher than the costs of consuming resources, C_b will be less than

C_a and A will have a profit margin $M = C_a - Q_b = C_a - C_b$. A can then offer Q_b to B to offset its cost and pocket M as its own additional profit. A could further offer part of M to B if B had competing alternatives to P_{AB}. Now, B will find that it can do at least as well as it would have with P_{B1}, so it will accept the offer. In the pursuit of larger profits via the market, the team arrives at the optimal solution.

Hoplites also enables the active coordination of several robots in a chain. In Figure 2*(c)*, we add a third obstacle to the environment. Although this changes B and C's local environments, it does not create any breaks in their default paths P_{*1}. Consequently, they will initially choose P_{*1} as in the previous environments. Also as before, only A faces a break in its perimeter. However, unlike in Figure 2*(b)*, B's price quote Q_b for P_{AB} will be much higher than before because P_{B2} will cause breaks between B and C and cause C_b to be much higher. Indeed, these breaks will be more costly than the breaks between A and the wall since the obstacle between B and C is larger than the obstacle between A and the wall. So, C_b will be larger than C_a and A will be unable to bid for B's participation.

However, B can generate the team plan $P_{BC} = \{P_{B2}, P_{C2}\}$ and request a quote Q_c from C that reflects the added cost C_c of taking P_{C2} instead of P_{C1}. We expect C_c to be small since P_{C2} only constitutes a small change in C's plan and does not create any perimeter breaks for C in the context of P_{BC}. Thus, upon receiving Q_c, B can evaluate its new cost C'_b that is the cost of taking P_{B2} instead of P_{B1} when C simultaneously takes P_{C2}. It can then send to A a new price quote $Q'_b = C'_b + Q_c$. Again, we expect that $C'_b + C_c$ will be less than C_a, so A will be have a profit margin $M = C_a - Q'_b = C_a - (C'_b + C_c)$. A will then offer Q'_b to B which, in turn, will offer Q_c to C. Both B and C find that they can do at least as well as they would have with their original paths, so they will accept the respective offers. Here, the system is able to find the more complex optimal solution via the market.

As shown in this example, Hoplites makes it possible to efficiently coordinate several robots. Firstly, planning occurs through a series of linked plans (e.g. P_{AB} and P_{BC}), so a single agent does not have to plan for a large number of robots. Secondly, the utility of a plan (e.g. Q_b and Q_c) is evaluated locally, so robots can use more accurate local information to generate more accurate cost estimates. Thirdly, by combining a series of price quotes (e.g. Q'_b), robots can concisely transmit the costs and benefits of a collection of plans. This means that robots can investigate different solutions to the same problem; the most cost effective one is naturally adopted in the market. Thus, a robot (e.g. A) can benefit from and even cause and pay for the coordination between its teammates (e.g. B and C) without ever being aware that the coordination occurs. In this way, the needs of one part of the team can be transmitted to and met by other parts of the team transparently and efficiently.

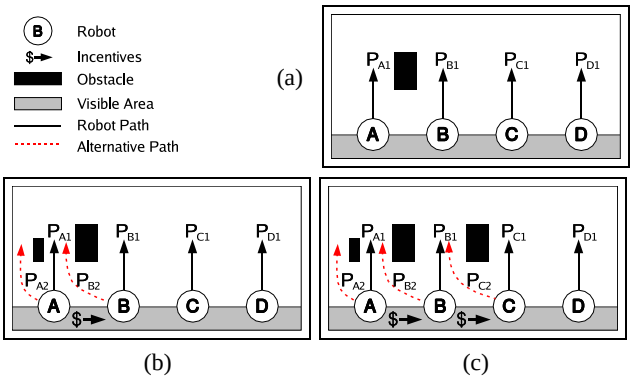


Fig. 2. Simple examples motivating active coordination. *(a)* An environment where passive coordination suffices. *(b)* An environment requiring active coordination. *(c)* An environment requiring active coordination between several teammates.

V. Simulation Experiments and Results

We performed experiments in simulation comparing our implementation of Hoplites on the security sweep domain to three other frameworks. Firstly, the original MVERT framework [11] developed by Stroupe served as our baseline for comparison. Secondly, we compared Hoplites to two further-improved versions of MVERT, which we call P-MVERT and PC-MVERT. We also implemented Hoplites on a real robot team (Section VI).

A. Frameworks for Comparison

1) MVERT: MVERT [11] is an emergent coordination framework that was originally used for foraging and tracking tasks for robot teams. In MVERT, a robot first estimates the next action of every other team member. Then, it chooses for itself the action that is most valuable given the expected contributions of the team in the next time step. Each robot repeats this procedure of selecting its next best action.

We believe that MVERT is the most appropriate approach to use for comparison for two reasons. Firstly it can be applied to tasks that have strong constraints between robots. As Stroupe notes, "The potential applications of MVERT include any tasks that can be represented by some computable mathematical function" [11]. Secondly, in contrast to other emergent approaches, it allows robots to plan their coordination by anticipating each other's future actions. Stroupe demonstrates that MVERT outperforms other approaches on several domains specifically because of this feature. Consequently, we believe that MVERT is "best in class" and thus is appropriate for comparison to Hoplites.

2) P-MVERT: MVERT has the obvious disadvantage that robots act myopically and cannot recognize a perimeter break until it is unavoidable. We have improved MVERT to allow extended planning to a horizon $h > 1$, resulting in what we have dubbed "P-MVERT" for Planning MVERT. A robot generates a set of candidate plans for itself and its teammates. It chooses for itself the plan that has the highest expected profit given a uniform distribution over the set of teammates' options. We do not permit a

more informed distribution to ensure that the performance difference between P-MVERT and MVERT is strictly a product of extended planning.

3) PC-MVERT: In PC-MVERT (Planning and Communicating MVERT), robots communicate their plans. So, robots use their teammates' intended actions to evaluate and select their own plans. PC-MVERT is identical to the passive coordination mechanism in Hoplites.

B. Simulation Details

We tested our approach in a graphical simulation of five robots tasked with clearing an environment. We generated twenty environments of size 200×200 units, each with twenty obstacles. The obstacles were randomly placed and had randomly chosen dimensions ranging from 5×5 units to 15×15 units. An example environment is shown in Figure 1*(a)*. We ran each approach on each environment fifty times.

The robots had no prior knowledge of the number or configuration of the obstacles in the environment. They were equipped with simulated 360° laser scanners with a range of 200 units to provide sensory information as they moved. Furthermore, each robot functioned as an independent software agent. Robots communicated via UDP and were therefore subject to the speed and fidelity of this protocol. Lastly, in simulation, we prescribed that a robot moved at a rate of one unit per second; this roughly scaled one map unit to one meter.

To make the domain challenging, we wanted robots to perform a sweep in which security is of the utmost importance. Specifically, we encouraged robots to travel up to fifty additional units to avoid breaking the perimeter. We penalized a robot $\$500$ for each step that either broke a previously-secure perimeter segment or failed to secure a previously-broken perimeter segment. In contrast, we only charged the robot $\$5$ per unit of time required to traverse the environment and $\$5$ per unit traveled. Each robot also received $\$40$ (a fifth of the room's width) for each unit it moved towards the far end of the environment. So, if a robot moved from location (x_0, y_0) at time t to (x_n, y_n) at time t' via a path $\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$, its profit p was computed by:

$$
\begin{array}{ll}
p = 40(y_n - y_1) & \text{Revenue} \\
\quad -500 \sum_{i=0}^{n-1} Broken(x_i, y_i, x_{i+1}, y_{i+1}) & \text{Penalty} \\
\quad -5 \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} & \text{Distance} \\
\quad -5(t' - t) & \text{Time}
\end{array}
$$

where $Broken(x_i, y_i, x_{i+1}, y_{i+1})$ is a function that returns 1 if the perimeter broke during the transition and 0 otherwise.

C. Results and Discussion

Table I summarizes the performances of MVERT, P-MVERT, PC-MVERT, and Hoplites. We measure performance in terms of the total number of perimeter breaks, the planning time spent by a single robot, and the distance traveled by a single robot. The first and second columns list each approach and the number of successful runs for each approach, respectively. The primary cause of run failure is that occasionally, a few robot processes will fail to attach to a network port, thereby causing failure in the initialization sequence. We did not rerun trials because of time constraints (each run takes approximately five minutes). We have excluded these runs from the results.

$\overline{B}$ is the mean of the total number of perimeter breaks incurred by the team, $\text{SEM}_{\overline{B}}$ is the standard error of this mean, and $\text{Rel}_{\overline{B}}$ is the number of perimeter breaks incurred by the current approach relative to the previously listed approach. This measure reflects the team's ability to perform a secure sweep. $\overline{T}$ is the mean of the computation time in seconds spent by a single robot, $\text{SEM}_{\overline{T}}$ is the standard error of this mean, and $\text{REL}_{\overline{T}}$ is the time taken by the current approach relative to the previously listed approach. It includes the time spent planning individual and team paths, updating map information, communicating state information, and negotiating with teammates. $\overline{D}$ is the mean of the distance in units traveled by a robot, $\text{SEM}_{\overline{D}}$ is the standard error of this mean, and $\text{Rel}_{\overline{D}}$ is the distance traveled by a robot in the current approach relative to the previously listed approach.

The improvement in performance between MVERT and P-MVERT reinforces the idea that this domain requires planning. The sequence of good local solutions created by MVERT's one-step lookahead does not add up to a good global solution. Naturally, this planning results in an increase in computation time. It also results in an increase in the distance traveled since robots will take longer paths around particular obstacles to avoid breaking the perimeter.

In PC-MVERT and P-MVERT, robots use the same algorithm to generate the set of candidate paths through the environment. In P-MVERT, robots can only guess which of these paths will be most profitable. However, in PC-MVERT, robots know precisely what their teammates will be doing. Because this information results in a significant improvement in the performance of the team, we can infer that knowing teammates' actions enables robots to choose correctly from those paths. This corroborates our claim that planned tight coordination is essential in this domain. Also, an increase in communication results in a decrease in computation time because a robot no longer needs to generate and then evaluate an entire set of plans for its teammates; instead, it can rely on the plans they broadcast. Finally, there is effectively no change in the distance traveled by the robots since both approaches generate candidate paths in the same way.

It is the performance improvement between PC-MVERT and Hoplites that most interests us. It substantiates our claim that passive coordination alone may trap robots in local minima, but, by developing team plans and actively influencing each other's actions, the team can avoid many of these pitfalls. The results also demonstrate that, through the market, Hoplites provides a powerful mechanism that allows robots to intentionally work together to find better solutions. There is an increase in computation time since developing team plans involves planning through high-

TABLE I

A COMPARISON OF THE PERFORMANCES OF THE FOUR APPROACHES USING THREE METRICS: THE NUMBER OF PERIMETER BREAKS CAUSED (B), THE PLANNING TIME IN SECONDS SPENT BY A SINGLE ROBOT (T), AND THE DISTANCE IN MAP UNITS TRAVELED BY A SINGLE ROBOT (D).

Algorithm	Runs	$\overline{B}$	SEM$_{\overline{B}}$	Rel$_{\overline{B}}$	$\overline{T}$	SEM$_{\overline{T}}$	Rel$_{\overline{T}}$	$\overline{D}$	SEM$_{\overline{D}}$	Rel$_{\overline{D}}$
MVERT	963	41.9	0.43	–	12.1	0.29	–	195.8	0.5	–
P-MVERT	973	22.4	0.51	0.54	29.2	1.72	2.41	200.4	0.53	1.02
PC-MVERT	957	8.6	0.20	0.39	20.1	0.36	0.69	198.8	0.54	0.99
Hoplites	966	5.5	0.15	0.63	28.9	0.90	1.44	208.6	1.44	1.05

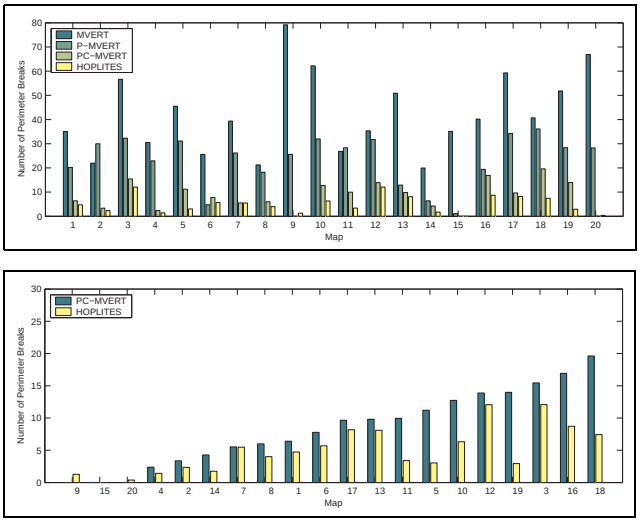


Fig. 3. The performances on individual environments. *(top)* The number of perimeter breaks caused by MVERT, P-MVERT, PC-MVERT, and Hoplites on individual environments ordered by number. *(bottom)* The number of perimeter breaks caused by PC-MVERT and Hoplites on individual environments, ordered by increasing complexity. Error bars have been left out for clarity.

dimensional joint action spaces. There is also an increase in distance traveled since the team plans that prevent perimeter breaks are usually long and more complex than the plans generated by P-MVERT and PC-MVERT.

The results in Table I combine the performance of the approaches over all the environments. In contrast, Figure 3*(top)* presents the relative performances of each of the approaches on individual environments. It is clear from the data that both planning and communication improve the performance of MVERT. More interestingly, we observe that the effect of enabling active coordination differs between environments.

We would like to examine specifically how the effects of passive and active coordination change as we make environments more and more challenging. However, it can be difficult to determine by inspection alone which environments are more complex than others. We use the performance of PC-MVERT as a measure of complexity. Figure 3*(bottom)* compares the performance of PC-MVERT and Hoplites. Environments are ordered by decreasing performance of PC-MVERT which correlates with increasing environmental complexity.

First, note that Hoplites outperforms PC-MVERT in nearly every environment. More importantly, the margin of improvement is significantly larger in more difficult environments. Specifically, suppose we divide the environments into two sets of ten, split into an "easy" set and a "hard" set based on PC-MVERT's performance. Then, the number of perimeter breaks incurred on the easy set by Hoplites relative to PC-MVERT is 0.76 as compared to 0.54 on the harder set. This difference exists in some part because there is simply more room for improvement when PC-MVERT performs poorly than when it performs well. However, it is more importantly due to the design of Hoplites. Active coordination between robots can require extensive computation and create very complex interactions between robots. This complexity is most justified and most beneficial in correspondingly complex situations.

VI. IMPLEMENTATION ON ROBOTS

We implemented Hoplites on a team of Pioneer II-DX robots equipped with SICK LMS 200 laser range finders that provide 180^o fields of view. Robots planned using only their local maps. Each robot was locally controlled and communicated with its teammates via 802.11b wireless Ethernet. We borrowed much of the supporting low-level controls and sensing from the TraderBots implementation by Dias et. al. [16].

Figure 4 shows a sequence of images from a sweep performed by three team members in a 7×10 meter environment that contained two large obstacles. This environment was consistent with the example environment shown in Figure 2*(b)*. As we described in Section IV-B, the left-most robot actively coordinated with the center robot to complete the sweep without breaking the perimeter. The team completed this task in under two minutes.

VII. CONCLUSIONS

In this paper we present Hoplites, a sophisticated market-based coordination framework that explicitly addresses planned tight coordination in multirobot teams. Hoplites consists of two novel coordination mechanisms; passive coordination quickly produces locally-developed solutions while active coordination produces complex team solutions via negotiation between teammates. Hoplites is the first market-based approach to operate in domains requiring tight coordination. Robots use the market to efficiently vet candidate solutions and also use market indicators to choose the coordination mechanism that best matches the current demands of the task. Our simulation experiments demonstrate that Hoplites is best in class, significantly

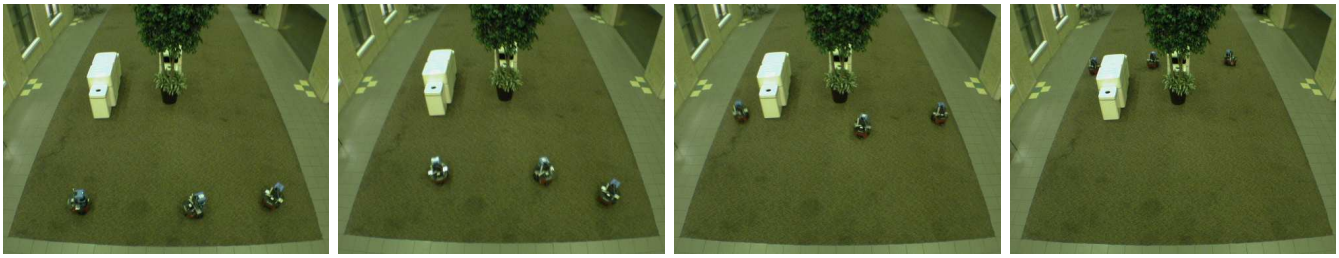


Fig. 4. Snapshots (in chronological order from left to right) of a team of three Pioneer II-DX robots completing a security sweep task. Active coordination begins between the left and center robots in the third frame, where the center robot chooses to move to the left of the center obstacle. This experiment was performed in the Newell-Simon Atrium at Carnegie Mellon University.

outperforming even improved versions of its competitors. We also validated Hoplites on a team of mobile robots performing a security sweep task.

In the immediate future we plan to develop more advanced negotiation and replanning protocols to better adapt commitments between teammates to changes in both the environment and the needs of the task. We will also apply Hoplites to other domains such as constrained exploration and the gallery monitoring problem. This will further validate our approach and enable us to strengthen different aspects of Hoplites.

REFERENCES

[1] N. Kalra, A. Stentz, and D. Ferguson, "Hoplites: A market framework for complex tight coordination in multi-agent teams," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-41, August 2004.

[2] L. E. Parker, "Designing control laws for cooperative agent teams," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 1993.

[3] T. Balch and M. Hybinette, "Social potentials for scalable multirobot formations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2000.

[4] R. G. Brown and J. Jennings, "A pusher/steerer model for strongly cooperative mobile robot manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, August 1995.

[5] H. G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J. M. Spector, "Autonomous communication relays for tactical robots," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, June 2003.

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

[7] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-UAV context," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2004.

[8] D. MacKenzie, "Collaborative tasking of tightly constrained multi-robot missions," in *Multi-Robot Systems*. Kluwer, 2003.

[9] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, August 2000.

[10] B. Gerkey, S. Thrun, , and G. Gordon, "Parallel stochastic hill-climbing with small teams," in *Multi-Robot Systems*. Kluwer, 2005.

[11] 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.

[12] A. Stentz and M. B. Dias, "A free market architecture for coordinating multiple robots," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-99-42, December 1999.

[13] B. P. Gerkey and M. J. Matarić, "Sold!: Auction methods for multi-robot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, Oct. 2002.

[14] R. M. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Market-driven multi-robot exploration," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-02-02, January 2002.

[15] R. M. Zlot and A. Stentz, "Market-based multirobot coordination using task abstraction," in *Proceedings of the International Conference on Advanced Robotics (FSR)*, July 2003.

[16] M. B. Dias, R. Zlot, M. Zinck, J. P. Gonzalez, and A. Stentz, "A versatile implementation of the traderbots approach for multirobot coordination," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, March 2004.

[17] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[18] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University, Tech. Rep. 98-11, October 1998.

[19] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM Journal on Computing*, vol. 21, no. 5, pp. 863–888, 1992.