Meet Point Planning for Multirobot Coordination

Siddhartha S. Srinivasa
Intel Research Pittsburgh
4720 Forbes Avenue
Pittsburgh, PA 15213
siddhartha.srinivasa@intel.com

Dave Ferguson
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dif@cmu.edu

***Abstract*— We present an approach for determining optimal meet points for a team of agents performing coordinated tasks in an environment. Such points represent the best locations at which the team should meet in order to complete their current task. We develop analytic solutions for finding these meet points and describe how these techniques can be applied to several different multi-agent coordination scenarios.**

I. Introduction

A wide range of tasks exist that are better solved using a team of agents rather than a single agent. Indeed, some of these tasks *require* a team for completion. In robotics, example applications that have benefited from multirobot solutions include exploration [1], [2], [3], tracking [4], [5], security sweeping [6], formation control [7], [8], and box pushing [9], [10].

In all of these applications, individual robots must work together and coordinate their actions in order to accomplish a team objective. Often, this coordination may take the form of two or more agents combining to accomplish some part of the task. For instance, consider two robots at different positions that wish to team up to push a box from some initial location to a desired goal location. Perhaps each robot can push the box on its own, but together they can push it much more effectively, at a faster speed. In order for the robots to get the box to its goal as quickly as possible, they need to decide where they are going to meet to push the box, whether one is going to start pushing first and have the other join it at a later point, and so on. Solving for the rendezvous points, or *meet points*, is an important part of coordinating the team.

In this paper, we develop an approach for constructing such meet points. Depending on the application, these points may correspond to places where two or more robots get together to exchange map information, to form a convoy for venturing into hazardous terrain, or to combine to move a box or cumbersome object. In each of these example cases, the best rendezvous points may differ significantly. We therefore introduce a general approach that can be used to generate the set of *all* possible optimal meet points, and show how we can optimize over this set to determine the true optimal point for a particular application.

This paper is structured as follows. In Section II we formally introduce the meet point planning problem for two agents and provide an approach for computing time-optimal meet points for two agents traveling at different speeds. We extend this result in Section III to provide the set of time-optimal meet points for two agents starting out at different times, and describe how this result can be used to compute optimal meet points for a range of different coordination scenarios. In particular, we apply this to our box-pushing example and provide an approach that computes an optimal meet point for two robots coordinating to move a box from some initial location to a desired location. We conclude in Section IV with discussion and future work.

II. Two agent time-optimal meet points

Consider two agents located at positions p_1 and p_2 in some environment and capable of moving with speeds k_1 and k_2, respectively. We wish to have the two robots meet each other in minimum possible time in order to exchange some information or combine to solve some task. We call the resulting point in the environment at which they rendezvous a *meet point*. If, by having each agent travel to this meet point at full speed, the agents are able to rendezvous in minimum possible time, the meet point is time-optimal.

When planning time-optimal paths, it is common in mobile robotics to use some discrete graph representation of the environment. One popular representation is a visibility graph [11]. A visibility graph encodes all of the possible time-optimal paths in a two-dimensional uniform-cost environment. The graph consists of a set of nodes, chosen to be the set of all obstacle corners combined with the initial and goal locations of any agents in the environment, and a set of edges connecting all nodes that are visible from one another. An example visibility graph for an environment containing two agents and five obstacles is shown in Figure 1(left).

Each edge in a visibility graph also has a weight or cost associated with it, corresponding to its length. Computing a time-optimal path between two nodes on a visibility graph consists of finding a sequence of edges that can be used to transition between the nodes and whose combined cost is the least out of all such sequences. Several efficient graph-based search algorithms exist for finding such paths, such as Dijkstra's search [12] or A* [13], [14].

It turns out that, in order to solve for a time-optimal meet point for two agents, we need only to solve for a time-optimal path from one agent to the other, then find the point along this path at which the agents would meet if they both moved towards each other at full speed.

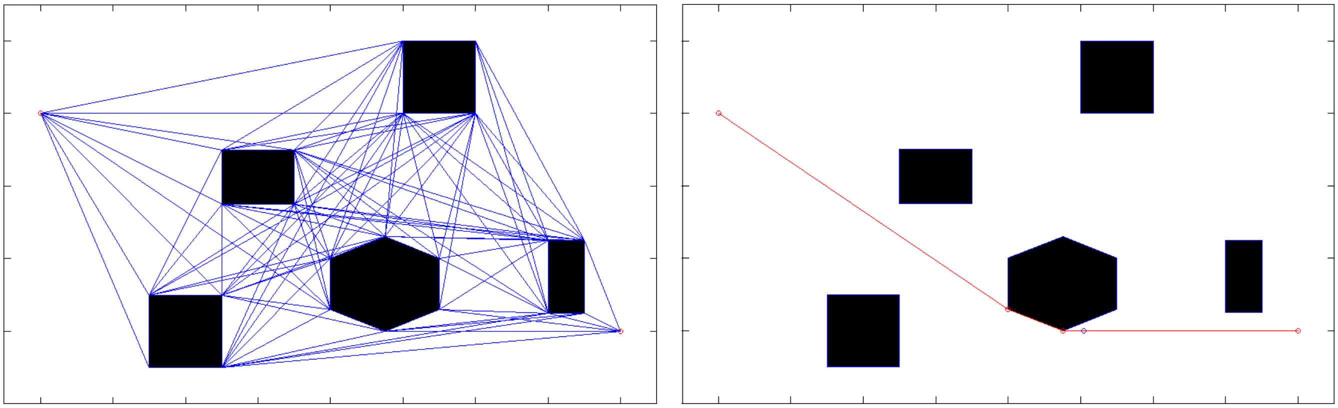

Fig. 1. Computing time-optimal meet points for two agents. (left) The visibility graph constructed for an environment containing five obstacles (shown in black) and two agents (the two red circles on the far left and right of the environment). This graph encodes all possible time-optimal paths from each agent to any of the obstacle vertices in the environment, as well as from one agent to the other. (right) The time-optimal path between the agent locations and the time-optimal meet point (shown as a blue circle) at which the agents should rendezvous if the left agent travels at twice the speed of the right agent.

Theorem 1: Given two agents located at positions p_1 and p_2 in an undirected, distance-weighted graph, and capable of moving with speeds k_1 and k_2, respectively, a time-optimal meet point for the agents will reside on any time-optimal path from p_1 to p_2. Further, this meet point will reside a distance of d_1 from p_1 along such a path, where

$$d_1 = d_P \frac{k_1}{k_1 + k_2} \tag{1}$$

with d_P is the length of a time-optimal path P from p_1 to p_2.

Proof: We first show the latter holds, namely that if the meet point is fixed to reside on a particular path P, then its position along this path is fixed by the above equation. Trivially, the time-optimal meet point along a fixed path results from each agent moving towards each other along the path at top speed. The position at which they will meet is thus determined by their relative speeds:

$$\frac{d_1}{k_1} = \frac{(d_P - d_1)}{k_2} \tag{2}$$

resulting in

$$d_1 = d_P \frac{k_1}{k_1 + k_2} \tag{3}$$

Now, since the lengths of all time-optimal paths are equal, if a time-optimal meet point resides on one time-optimal path, time-optimal meet points must exist on all the time-optimal paths. So assume no time-optimal meet point resides on a time-optimal path P from p_1 to p_2. Then the meet point must reside on some non-optimal path, which we denote by Q[1]. We know from above that the position of the meet point along this path Q is

$$d_1 = d_Q \frac{k_1}{k_1 + k_2} \tag{4}$$

[1]The meet point must reside on *some* path connecting p_1 and p_2 since each agent travels from its respective location to the meet point and so the union of the two agent's paths to the meet point is a path from p_1 to p_2.

If we were to fix the meet point to reside along path P, then its position would be

$$d_{1*} = d_P \frac{k_1}{k_1 + k_2} \tag{5}$$

Since P is a time-optimal path and Q is not, $d_P < d_Q$. But this means that $d_1 > d_{1*}$, so the time taken to reach our time-optimal meet point is greater than the time taken to reach the meet point on path P. Contradiction. Hence, a time-optimal meet point resides on any time-optimal path from p_1 to p_2. ■

This result is encouraging, because it means that we can compute time-optimal meet points very efficiently. It is also very general: the result applies to any undirected graph, not just visibility graphs, and in fact also holds in any space in which a time-optimal path from p_1 to p_2 for the first agent is also a time-optimal path (in reverse) from p_2 to p_1 for the second agent. Further, if there is more than one time-optimal path for agent 1 and some of these paths do not correspond to time-optimal paths for agent 2, then as long as one of the paths that does correspond to a time-optimal path for agent 2 is chosen, the same result holds.

Figure 1(right) shows a time-optimal meet point for the example scenario in Figure 1(left), along with the time-optimal path taken by each agent to this point. In this example, the left agent travels at twice the speed of the right agent, and so the distance of the meet point along the path from the left agent is twice its distance along the path from the right agent.

III. Computing Sets of Meet Points

The approach developed above allows us to compute a time-optimal meet point for two agents starting at different locations in an environment and traveling at different speeds. However, it may be the case that the two agents are not just trying to minimize the time required to rendezvous, but the time required to perform some coordinated task, such as pushing a box to a goal location. In such a case, the time-optimal meet point for the two agents may not be the best place for them to congregate to get the box to the goal most efficiently.

For instance, consider two agents, $R1$ and $R2$, where $R1$ is pushing a box to some goal location G and wishes to solicit the help of $R2$. Rather than having $R1$ and $R2$ move directly towards each other to rendezvous, it may be faster to have $R1$ start pushing the box towards G and have $R2$ meet up with it along the way to help.

Because we are now trying to minimize over the time to complete a task, rather than just the time for the agents to meet, the time-optimal meet point calculation in the previous section may not be enough. Instead, we would like to generate the set of all possible meet points for the two agents so that we can optimize over this set to find the best meet point for the current task at hand. It turns out that, for a given meeting time t, there are at most two locations at which the agents can meet (assuming the agents take the fastest paths possible to their destinations) and we can construct the full set of possible meet points by varying this meeting time.

It may also be the case that our agents start moving towards their meet point at different times. In our current box-pushing example, each agent's path to the meet point will begin with moving along some edges on the visibility graph, then at some point will depart from a node on the visibility graph to move directly towards the meet point. We can analytically compute the final meet point if we know the final visibility nodes passed through by the agents. But because the agents travel at different speeds and these nodes may be different distances from the initial positions of the agents, the agents may reach, and thus leave, these nodes at different times. As a result, our set of possible meet points depends on the difference in time, δt, between when agents $R1$ and $R2$ reach their final visibility nodes.

Claim 1: There are no pauses in the time-optimal trajectory — all robots are either moving at the maximum possible speed, or have reached the goal.

Proof: It is clear that if there were just one robot moving to a goal, the time-optimal trajectory will have no pauses. The robot moves as quickly as it can along the shortest path. In the presence of other robots, the only time the robot might consider to pause is if it needs to wait to meet another robot. However, it could as well have spent that wait time moving towards the other robot and achieved a quicker meeting. Hence there are no pauses in moving to a meet point. Since the multi-robot trajectory comprises of either moving to a meet point or to a goal, there are no pauses in the entire trajectory. ■

Theorem 2: Given two robots, $R1$ and $R2$ located at $(0,0)$ and $(d,0)$, respectively, and capable of moving with speeds k_1 and k_2, respectively, and given that $R2$ reaches its location a time δt after $R1$, the meet point (x,y) at time t is given by:

$$x(t) = \frac{1}{2d}\left(d^2 - k_2^2(t-\delta t)^2 + k_1^2 t^2\right) \quad (6)$$

$$y(t) = \pm\sqrt{k_1^2 t^2 - x^2} \quad (7)$$

subject to:

$$t \in [t_l, t_u] \quad (8)$$

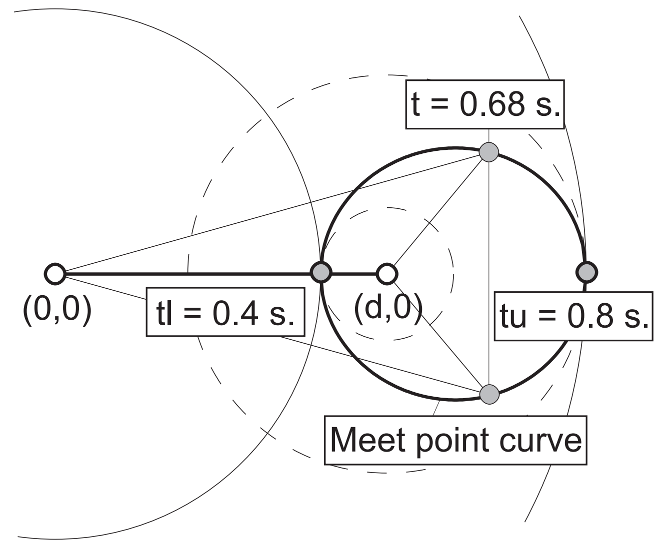

Fig. 2. The dark circle shows the meet point curve for $k_1 = 2$, $k_2 = 1$, $d = 1$, and $\delta t = 0.2s.$. Also shown are the two candidate meet points at $t = 0.68s.$ and the meet time interval $[0.4s., 0.6s.]$. The lighter solid and dashed circles represent the locus of points reachable in a given time, for the individual robots.

where:

$$t \in \emptyset$$
$$\text{if } k_1 \geq k_2 \text{ and } (d - k_1\delta t) < 0$$

otherwise:

$$t_u = \begin{cases} \dfrac{(d - k_2\delta t)}{(k_1 - k_2)} & \text{if } k_1 \geq k_2 \\ \dfrac{(d + k_2\delta t)}{(k_2 - k_1)} & \text{if } k_1 < k_2 \end{cases}$$

$$t_l = \frac{(d + k_2\delta t)}{(k_1 + k_2)}$$

Proof: The proof is in two parts. In the first part, we compute the optimal meet point (Eqn.6 and Eqn.7) assuming that the two robots can meet at time t. In the second part, we compute the constraints on t (Eqn.8).

Since the two robots reach the meet point at the same time,

$$\frac{\sqrt{x^2+y^2}}{k_1} = \delta t + \frac{\sqrt{(x-d)^2+y^2}}{k_2} = t \quad (9)$$

Squaring Eqn.9, we get

$$\frac{x^2+y^2}{k_1^2} = t^2 \quad (10)$$

$$\frac{(x-d)^2+y^2}{k_2^2} = (t-\delta t)^2 \quad (11)$$

Substituting the value for y^2 from Eqn.10 into Eqn.11, we obtain

$$x(t) = \frac{1}{2d}\left(d^2 - \frac{(t-\delta t)^2}{k_2^2} + \frac{t^2}{k_1^2}\right)$$

$$y(t) = \pm\sqrt{k_1^2 t^2 - x^2}$$

A sample meet point curve is illustrated in Fig.2. Note that there are two solutions for the meet point for a given time t, due to symmetry.

To compute Eqn.8, the following visualization is useful. For each robot, the locus of points that the robot can reach in a given time is a circle centered around the start point. As time increases, the circles get bigger. These are shown as the lighter solid and dashed circles in Fig.2. The intersection of the circles for the two robots defines a meet point at that time t.

The circles of $R2$ start to grow δt after those of $R1$. If by the time $R2$ starts, $R1$ has already passed it, the only way $R2$ can catch up is if it is faster than $R1$. If it is slower, the two robots can never meet. This can be written as

$$t \in \emptyset$$
$$\text{if } k_1 \geq k_2 \text{ and } \frac{d}{k_1} < \delta t$$

Once we have ensured that the two robots can indeed meet, the earliest time they can meet is if they head straight towards each other. This corresponds to the time when the two circles first touch each other. This can be easily computed as

$$t_l = \frac{(d + k_2 \delta t)}{(k_1 + k_2)}$$

The upper bound on t occurs due to the fact that the two circles expand at different rates. Eventually, the faster expanding circle will engulf the slower one. At the critical point, the two circles will touch at exactly one point and the point will lie on the line joining the two start points. There are two cases for t_u because if the faster robot starts with the time delay δt, it will take more time to engulf the slower robot than if it did not start with the time delay. Note that if $k_1 = k_2$, then $t_u = \infty$. ∎

Corollary 1: If the two robots $R1$ and $R2$ are at (x_1, y_1) and (x_2, y_2) respectively, the meet point $(\bar{x}(t), \bar{y}(t))$ at time t is given by

$$\begin{pmatrix} \bar{x}(t) \\ \bar{y}(t) \end{pmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{12}$$

where

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

and (x, y) are computed from Eqn.6 and Eqn.7 with

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Proof: This follows from the invariance of the Euclidean metric under rigid transformation. ∎

The above result provides the set of all meet points for two agents starting at two different positions at two different times. Figure 3 illustrates the set of meet points for the scenario shown in Figure 1. Note that it is possible that some of the potential meet points may reside within obstacles in the environment. Such points can easily be detected and should be discarded.

We can use the above result to find all meet points for a collection of N agents by calculating the loci as above for each pair of agents, then intersecting these loci. This ensures that the agents all reach the same place. For sufficiency, the agents also need to reach the points at the same time. These points will be rare, even for the simple case of three agents — potential meet points not only need to lie on the intersection of three circles but also need to have the same time parametrization.

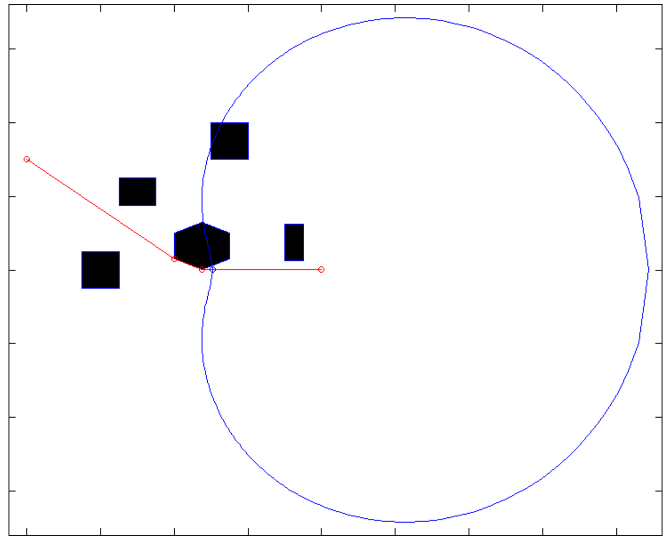

Fig. 3. Computing the set of all time-optimal meet points for two agents. Here, we have used the result from Theorem 2 to construct the set of possible meet points for the two agents initially traveling along the path shown in Figure 1. This set of meet points is computed from the last visibility nodes passed through by each agent; in this case, these points are the bottom corner of the six-sided obstacle for robot $R1$ and the initial location for $R2$.

Once equipped with this set of meet points, we can then optimize over the set to compute optimal solutions for particular applications. For example, take our box-pushing task and assume that one of the agents starts out pushing the box (so that the speed of that agent reflects how fast it can push the box on its own) and that the two agents together can push the box at speed k_3. The total time taken to reach the goal, τ, and the time taken to reach the meet point, t, are related by:

$$\tau = t + \frac{d_g(t)}{k_3}$$

where

$$d_g(t) = \sqrt{(x_g - x(t))^2 + (y_g - y(t))^2}$$

and $x(t)$ and $y(t)$ are obtained from Eqn.6 and Eqn.7, respectively.

At the extrema of τ

$$\frac{d\tau}{dt} = 1 + \frac{1}{k_3}\frac{d}{dt}(d_g(t)) = 0 \tag{13}$$

It can be shown that Eqn.13 will have only two roots, facilitating a fast numerical solution. A sample solution is shown in Fig.4.

Note that the solution obtained above might not be time optimal since it might be faster for the two agents to just head to the goal independently and not meet at all. These times are, however, easy to compute and compare.

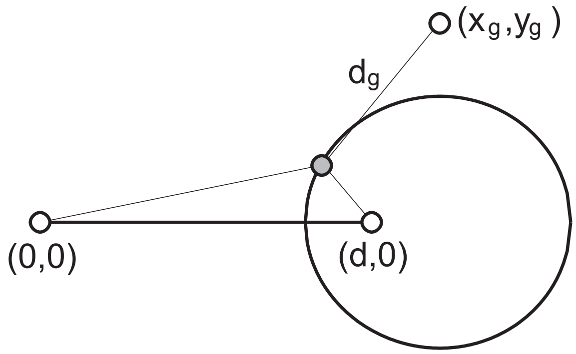

Fig. 4. The optimal meet point and the path to the goal for the example problem mentioned in the text, with $k_3 = 3$ and the goal at $(0.6, 1.2)$.

The set of all meet points can also be used for several other applications. For example, it can be used to compute time-optimal paths for agents wishing to meet in a particular region by simply intersecting the curve with the region, leaving us with a set of meet points that reside within the desired region. From this set, a single meet point can then be selected that enables the team to meet in minimum possible time. The set of meet points can also be used for convoy tasks, where we would like groups of agents to form before venturing into certain areas of the environment. In this scenario, it makes sense to select a meet point from our set that is located outside the area of interest and that minimizes the overall time to accomplish the convoy operation.

IV. Discussion

We have presented an analytical solution to the problem of computing the meet point locus for multi-robot coordination tasks. We have also shown how this result can be used to compute the time-optimal trajectory to reach a goal point by means of a one-dimensional gradient descent algorithm. Finally, we have discussed how our algorithm can be integrated with a visibility graph planner in the case where the departing visibility nodes for the agents are known, giving us time-optimal trajectories parametrized by meet time t, in the presence of obstacles.

In the future, we would like to extend our algorithm by relaxing the requirement of knowing the departing visibility nodes. A simple, albeit naïve, method for doing this would be to calculate time-optimal paths from each agent to every node in the visibility graph and store the corresponding times associated with these paths. Then, the set of all pairs of visibility nodes (n_i, n_j) can be constructed and we can use the stored time values to calculate the difference in arrival time, δt, between the first agent arriving at node n_i and the second agent arriving at node n_j. We can then use these node pairs and differences in arrival times as inputs to our time-optimal meet point calculation. The meet point with smallest overall time t is our global time-optimal meet point.

However, in large environments with several obstacles this na ive approach could be rather computationally expensive. Significant improvements in efficiency could be gained if we were able to remove from contention some of the pairs of visibility nodes. To do this, we could employ a heuristic search, where we compute efficient lower and upper bounds on the time taken to complete the task using various pairs of visibility nodes, and we use these bounds to focus our computation on the most promising pairs of nodes. For instance, if the two agents start out close to each other and the goal location in a large environment, then it makes sense to investigate meet points that are close to the agents to begin with, and it may be possible to quickly rule out any solutions that involve distant nodes as being grossly suboptimal. Such an approach follows the same general ideas as A* search, and appears very promising.

Also, for each pair of visibility nodes, given the speeds k_i, we can precompute the range of δt for which the robots will meet before reaching the goal as opposed to just moving to the goal without bothering to meet. From Theorem 2, we can compute a lower bound $\delta t = \frac{d}{k_1}$ if $k_1 \geq k_2$. The upper bound on dt can be computed from the fastest or the slowest (depending on the definition of completion as one robot reaching the goal or both robots reaching the goal) of the time taken for either robot to reach the goal. For n robots, choosing the slowest and the fastest k_i gives us a fast conservative estimate of the bounds on δt. These bounds can then also be used as a heuristic during the search.

We are also interested in applying these results to multirobot coordination involving large teams. Some of the most effective coordination frameworks for dealing with complex multirobot tasks involve market techniques, where individual agents act in self-interested fashions, but occasionally form sub-teams to maximize their effectiveness as a group [15]. Such frameworks have the nice property of being able to employ various planning algorithms to compute individual or sub-team solutions. We are currently investigating how the techniques presented in this paper can be incorporated into an impressive recent extension of market frameworks [16] to provide efficient, high-quality solutions to complex tasks involving large teams.

Acknowledgements

The authors would like to thank Rob Zlot for many fruitful discussions on multirobot frameworks for coordination. Dave Ferguson is supported in part by a National Science Foundation Graduate Research Fellowship.

References

[1] 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.

[2] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2000.

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

[4] 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.

[5] J. Kim, R. Vidal, D. Shim, O. Shakernia, and S. Sastry, "A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles," in *Proceedings of the IEEE Conference on Decision and Control*, December 2001.

[6] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[7] D. J. Naffin and G. S. Sukhatme, "Negotiated formations," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, March 2004.

[8] P. Pirjanian, C. Leger, E. Mum, B. Kennedy, M. Garrett, H. Aghazarian, S. Farritor, and P. Schenker, "Distributed control for a modular, reconfigurable cliff robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[9] 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.

[10] 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.

[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, "Visibility graphs: Finding the shortest route," in *Computational Geometry: Algorithms and Applications, Second Edition*. Springer-Verlag, 2000, pp. 307–317.

[12] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[13] P. Hart, N. Nilsson, and B. Rafael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968.

[14] N. Nilsson, *Principles of Artificial Intelligence*. Tioga Publishing Company, 1980.

[15] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based Multirobot Coordination: A Survey and Analysis," *IEEE Special Issue on Multirobot Coordination*, 2006.

[16] R. Zlot and A. Stentz, "Market-based multirobot coordination for complex tasks," *International Journal of Robotics Research, Special Issue on the 5th International Conference on Field and Service Robotics*, vol. 25, no. 1, 2006.