in *Encyclopedia of Complexity and System Science*, Robert A. Meyers, Editor-in-Chief, Springer, 2009.

Path Planning and Motion Coordination in Multiple Mobile Robot Teams

Lynne E. Parker
Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, Tennessee, USA

Article Outline

Glossary

Autonomous robot
An *autonomous robot* is a robot that can perform tasks in unstructured environments with minimal human guidance.

Planned path
A *planned path* is a pre-determined, obstacle-free, trajectory that a robot can follow to reach its goal position from its starting position.

Complete path planner
A *complete* path planner is an algorithm that is guaranteed to find a path, if one exists.

Deadlocked path planning
A *deadlock* is a situation in path planning in which a solution cannot be found, even though one exists. Typically, this is caused by robots blocking each other's paths, and the planner being unable to find a solution in which robots move out of each other's way.

1 Definition of the Subject and Its Importance

Multi-robot path planning and motion coordination addresses the problem of how teams of autonomous mobile robots can share the same workspace while avoiding interference with each other, and/or while achieving group motion objectives. Nearly all applications of multiple autonomous mobile robots must address this issue of motion coordination, either explicitly or implicitly. Multi-robot path planning and teaming has been extensively studied since the 1980s. While many techniques have been developed to address this challenge, the general centralized multi-robot path planning problem is known to be intractable, meaning that optimal solutions cannot be found in polynomial time. Thus, alternative techniques that decouple aspects of the motion planning and coordination problem have been proposed that trade off optimality for efficiency. A wide variety of applications can benefit from teams of robots that can coordinate their motions effectively, including search and rescue, planetary exploration, mineral mining, transportation, agriculture, industrial maintenance, security and surveillance, and warehouse management.

2 Introduction

Many practical applications of autonomous robots require the use of multiple team members. Such teams have many potential benefits, including faster task completion time (through parallelism) and increased robustness (through redundancy). Further, teams of robots can increase the application domain of autonomous robots by providing solutions to tasks that are inherently distributed, either in time, space, or functionality. Since the 1980s, researchers have addressed many issues in multi-robot teams, such as control architectures, communication, task allocation, swarm robots, learning, and so forth [83].

A critical issue in these mobile robot teams is coordinating the motions of multiple robots interacting in the same workspace. Regardless of the mission of the robots, they must be able to effectively share the workspace to prevent interference between the team members. Solutions to the motion coordination problem are approached in a variety of ways, depending upon the underlying objectives of the robot team. In some cases, the paths of the robots are explicitly planned and coordinated in advance, as might be needed in a busy warehouse management application, for example. In other cases, planning is relaxed and emphasis is placed on mechanisms to avoid collision, applicable for tasks such as automated hospital meal deliveries. In yet other situations, the robots could have mechanisms with little pre-planning that focus on coordinating robot motions in real-time using reactive, behavior-based, or control-theoretic approaches, such as would be used in a convoying or formation-keeping application.

The *multi-robot path planning problem* is defined as follows: given a set of m robots in k-dimensional workspace, each with an initial starting configuration (e.g., position and orientation) and a desired goal configuration, determine the path each robot should take to reach its goal, while avoiding collisions with obstacles and other robots in the workspace. More formally (adapting the notation of [58, 59]), let $\mathcal{A}$ be a rigid robot in a static workspace $\mathcal{W} = \mathbb{R}^k$, where $k = 2$ or $k = 3$. The workspace is populated with obstacles. A *configuration* $\mathbf{q}$ is a complete specification of the location of every point on the robot geometry. The *configuration space* $\mathcal{C}$ represents the set of all the possible configurations of $\mathcal{A}$ with respect to $\mathcal{W}$. Let $\mathcal{O} \subset \mathcal{W}$ represent the region within the workspace populated by obstacles. Let the closed set $\mathcal{A}(\mathbf{q}) \subset \mathcal{W}$ denote the set of points occupied by the robot when it is in the configuration $\mathbf{q} \in \mathcal{C}$. Then, the *C-space obstacle region*, C_{obs}, is defined as:

$$\mathcal{C}_{obs} = \{\mathbf{q} \in \mathcal{C} | \mathcal{A}(\mathbf{q}) \cap \mathcal{O} \neq \emptyset\}.$$

The set of configurations that avoid collision (called the *free space*) is:

$$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}.$$

A *free path* between two obstacle-free configurations c_{init} and c_{goal} is a continuous map:

$$\tau[0,1] \rightarrow \mathcal{C}_{free}$$

such that $\tau(0) = c_{init}$ and $\tau(1) = c_{goal}$.

For a team of m robots, define a state space that considers the configurations of all the robots simultaneously:

$$X = \mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^m.$$

Note that the dimension of X is N, where $N = \sum_{i=1}^{m} \dim(\mathcal{C}^i)$. The C-space obstacle region must now be redefined as a combination of the configurations leading to a robot-obstacle collision, together with the configurations leading to robot-robot collision. The subset of X corresponding to robot $\mathcal{A}^i$ in collision with the obstacle region, $\mathcal{O}$, is

$$X^i_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{O} \neq \emptyset\}. \tag{1}$$

The subset of X corresponding to robot $\mathcal{A}^i$ in collision with robot $\mathcal{A}^j$ is

$$X^{ij}_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{A}^j(\mathbf{q}^i) \neq \emptyset\}. \tag{2}$$

The obstacle region in X is then defined as the combination of Equations 1 and 2, resulting in

$$X_{obs} = \left(\bigcup_{i=1}^{m} X^i_{obs} \right) \bigcup \left(\bigcup_{ij, i \neq j} X^i_{obs} \right). \tag{3}$$

With these definitions, the planning process for multi-robot systems treats X the same as $\mathcal{C}$, and X_{obs} the same as $\mathcal{C}_{obs}$, where c_{init} represents the starting configurations of all the robots, and c_{goal} represents the desired goal configurations of all the robots.

Typically, optimization criteria guide the choice of a particular solution from an infinite number of possible solutions. Example criteria include minimized total path lengths, minimized time to reach goals, and minimized energy used to reach goals. Additional constraints can introduce more complexity in the planning process, such as navigational restrictions on the robots (e.g., maximum slope restrictions, inability to traverse rocky terrain, etc.), or the need for multiple robots to move in tandem with each other (e.g., a formation of robots moving over uneven terrain). Since the general optimal motion planning problem for multiple moving objects is computationally intractable (specifically, PSPACE-hard [47]), most approaches relax the requirement for global optimality, and instead seek to locally optimize portions of the path planning problem.

Planning approaches can be categorized, or taxonomized, in various ways. One taxonomy evaluates approaches in terms of completeness (i.e., whether they are guaranteed to find a solution if one exists), complexity (i.e., the computational requirements of the search process), and optimality (i.e., the quality of the resulting solution). Often, techniques that are complete and optimal are too computationally intensive to use in practice. Alternatively, techniques that achieve computational tractability typically trade off optimality and/or completeness.

Another taxonomy of multi-robot path planning techniques makes distinctions based on the amount of information used during the planning process. Approaches that use global information and plan directly in X are called *coupled, centralized* approaches. These approaches treat the robot team as a composite robot system, to which classical single-robot path planning algorithms are applied. For example, the A* algorithm [45] can generate complete and optimal solutions to the multi-robot path planning problem under a centralized and coupled approach. However, this type of planning approach requires computation time that is exponential in the dimension of the multi-robot configuration space. Thus, these approaches can only be used in real-time for the smallest of problem sizes. Section 3 describes these coupled, centralized techniques.

To deal with the high-dimensionality of X, alternative approaches *decouple* the path planning problem into independent components that can find good solutions quickly, although at the cost of losing optimality and completeness. These decoupled techniques can either be centralized or decentralized. Common examples of decoupled approaches include those that separate path planning and velocity planning. Typical approaches to decoupled planning will plan individual paths for a robot or set of robots, followed by a second step to resolve any potential conflicts between the paths. Section 4 describes some common techniques for decoupled multi-robot path planning.

A broader problem in multi-robot teams is that of *motion coordination*. Motion coordination encompasses multi-robot path planning, but also includes other problems such as flocking, formation-keeping, multi-robot target tracking, and other similar objectives. These tasks do not necessarily require advance planning of specific paths for each robot, but do require the coordination of trajectories as the robots move, to avoid collisions with each other, or to reach other group-level objectives, such as maintaining a desired inter-robot distance. Section 5 describes some of these techniques. This chapter is concluded with Section 6, which offers remarks on the future directions and impact of multi-robot path planning and motion coordination.

3 Coupled, Centralized Approaches

In *coupled, centralized* approaches to multi-robot path planning, the robot team is considered to be a composite robot system, to which a classical single-robot path planning algorithm is applied. Motion planning algorithms for single mobile robot systems have been intensively studied for years (see [58, 97, 40, 48]). Examples of classical single-robot path planning algorithms include sampling-based planning, potential-field techniques, and combinatorial methods. Sampling-based planners [54] avoid the explicit construction of $\mathcal{C}_{obs}$ by sampling different configurations to generate curves that represent collision-free paths in $\mathcal{C}_{free}$. Potential field techniques (e.g., [9, 11, 114]) construct real-valued functions that pull the robot toward the goal, and repulse the robot away from obstacles, via a combination of force vector fields. Combinatorial methods construct roadmaps through the configuration space using techniques such as cell decomposition (e.g., [75, 100]).

In an environment that contains a set of stationary obstacles, single robot path planning methods such as graph searching based on a geometric configuration of the environment are guaranteed to return optimal paths (in the sense of a performance measure such as shortest distance) in polynomial time if one exists. However, motion planning in a dynamic environment with moving obstacles is inherently harder. Even for a simple case in two dimensions, the problem is PSPACE-hard and is not solvable in polynomial time [35, 47]. Motion planning in dynamic environments was originally addressed by adding the time dimension to the robot's configuration space. The approach in [29] discretizes the configuration-time space to a sequence of slices of the configuration space at successive time intervals, representing the motions of moving obstacles using the set of slices embodying space-time. In [79], moving obstacles are represented as sheared cylinders, and a methodology was proposed to provide optimal tangent paths to the goal for a dynamic robot environment.

Extending the problem still further, to multiple robot path planning, requires even more computational resources. An example centralized approach for generating complete multi-robot path solutions is the work of Parsons and Canny [85], which takes a global cell decomposition approach, incorporating obstacles and other robots in a unified configuration space representation. This algorithm first computes a decomposition of the free space into cells; it then searches through the resulting adjacency graph for a path. However, not surprisingly, the algorithm is exponential in the number of robots. Other centralized algorithms that represent the path planning problem as a cross product of the configuration spaces of the individual robots include [96, 11].

Because of the high dimension of the multi-robot configuration space, centralized approaches that treat the multi-robot team as a single composite robot tend to be impractical computationally if the full search space is used. Instead, techniques that reduce the size of the search space have been shown to be practical for small-sized problems. One way to reduce the search space is to weakly constrain the allowable paths that robots can follow by limiting the motion of the robots to lie on *roadmaps* in the environment. Intuitively, roadmaps are akin to automotive highways, where robots move from their starting position to a roadmap, move along the roadmap to the proximity of the goal, and then move off the roadmap to the specific goal location. More formally, a roadmap is defined as follows [24]:

Definition (Roadmap): A union of one-dimensional curves is a **roadmap** RM if for all q_{start} and q_{goal} in $\mathcal{C}_{free}$ that can be connected by a path, the following properties hold:

1. **Accessibility:** there exists a path from $q_{start} \in \mathcal{C}_{free}$ to some $q'_{start} \in RM$,
2. **Departability:** there exists a path from $q'_{goal} \in RM$ to $q_{goal} \in \mathcal{C}_{free}$, and

3. **Connectivity:** there exists a path in RM between q'_{start} and q'_{goal}. □

Typically, a roadmap RM is represented as a graph $G = (V, E)$, in which the nodes V represent collision-free configurations, and the edges E represent feasible paths. (A *feasible* path is one that can be executed by robot $\mathcal{A}^i$, based on its physical motion constraints.) Various algorithms have been created that make use of the roadmap concept for motion planning, both for single robots and for multi-robot teams (e.g., [93, 87, 109]). The following subsections present two such approaches for multi-robot teams. The first, in work by Švestka and Overmars, is a probabilistically complete approach, meaning that the problem is solvable in finite time. Their approach creates a coordinated path for a composite robot by making use of the concept of *super-graphs*. The second, in work by Peasgood, et al., [87], is a multi-phase approach that uses a graph and spanning tree representation to create paths through the environment. This approach is shown to have linear-time complexity, and is thus scalable to much larger robot teams.

Before presenting these two approaches, it is worth noting that many other roadmapping approaches to multi-robot path planning have been proposed. For example, the work of Ryan [93] reduces the search space by decomposing the original map into subgraphs, planning paths between subgraphs, and then coordinating motions within the subgraphs. This approach has been shown to be effective for up to 10 robots. In [26], Clark, et al., introduce the concept of *dynamic networks*, which are formed between robots that are within communication range. Within this framework, only robots within the same network use a centralized planner, which is based upon probabilistic road maps [54]; otherwise, robots plan their paths using decoupled planners based on optimizing priorities (see Section 4). In [95], efficiencies in the probabilistic road map are achieved by delaying collision checking along the roadmaps until necessary. The speed-up achieved by this collision-checking (on the order of a factor of 4 to 40) allows this technique to be used more practically for small-sized multi-robot teams. The authors incorporate this improved planning process into three multi-robot path planning variants: a centralized version, a decoupled planner with global coordination, and a decoupled planner with pair-wise coordination.

3.0.1 Super-graph method (Švestka and Overmars)

In [109], Švestka and Overmars present an approach for creating a composite roadmap, which represents a network of feasible motions for the composite robot. This composite roadmap is created as follows. First, a roadmap for each individual robot is constructed using the standard roadmap generation algorithm, Probabilistic Path Planner (PPP) [54]. Then, n such roadmaps are combined into a roadmap for the composite robot, which can be used to generate coordinated paths.

Specifically, the *coordinated path* for the composite robot $(\mathcal{A}^1, \ldots, \mathcal{A}^n)$ is an n-tuple of paths feasible for all robots $\mathcal{A}^i$ that, when executed simultaneously, introduce no mutual collisions between the individual robots. Formally, let $\mathcal{C}^{[0,1]}$ represent the configuration space from time $t = 0$ to time $t = 1$, where the robot is at its starting position at time 0, and is at its goal location at time 1. Let $s_1, \ldots, s_n$ and $g_1, \ldots, g_n$ be given starting and goal configurations for the n robots, where $\forall i \in \{1, \ldots, n\} : s_i \in \mathcal{C}_{free} \wedge g_i \in \mathcal{C}_{free}$. Let P represent a free path if P is in $\mathcal{C}_{free}$ for all times t (i.e., $\forall t \in [0, 1] : P(t) \in \mathcal{C}_{free}$). Let $A \cap B \neq \emptyset$ (i.e., A and B intersect) be represented by $A \otimes B$. Then if $P_1, \ldots, P_n \in \mathcal{C}^{[0,1]}$ are feasible paths, such that for all $i, j \in \{1, \ldots, n\}$

- $P_i(0) = s_i \wedge P_i(1) = g_i$
- $i \neq j \Rightarrow \forall t \in [0, 1] : \neg\mathcal{A}(P_i(t)) \otimes \mathcal{A}(P_j(t))$

then $(P_1, \ldots P_n)$ is a coordinated path for $(\mathcal{A}^1, \ldots \mathcal{A}^n)$ solving the problem $((s_1, \ldots, s_n), (g_1, \ldots, g_n))$.

Švestka and Overmars present an approach for constructing such a coordinated path for a composite robot [109]. The basic idea is to seek paths along the roadmap, G, that allow the robots to move from their starting to their goal configurations, while disallowing simultaneous motions or motions along paths that are blocked by other robots. This type of path is called a *G-discretized coordinated path*. They introduce the concept of *super-graphs*, which represent roadmaps for the composite robots created by combining n simple robot roadmaps. Two variants of super-graphs are proposed – *flat super-graphs* and *multi-level super-graphs*.

In the flat supergraph, a node represents a feasible placement of the n simple robots at the nodes of G, and an edge represents a motion of exactly one simple robot along a non-blocked path of G. A disadvantage of the flat supergraph is that its size is exponential in the number of robots.

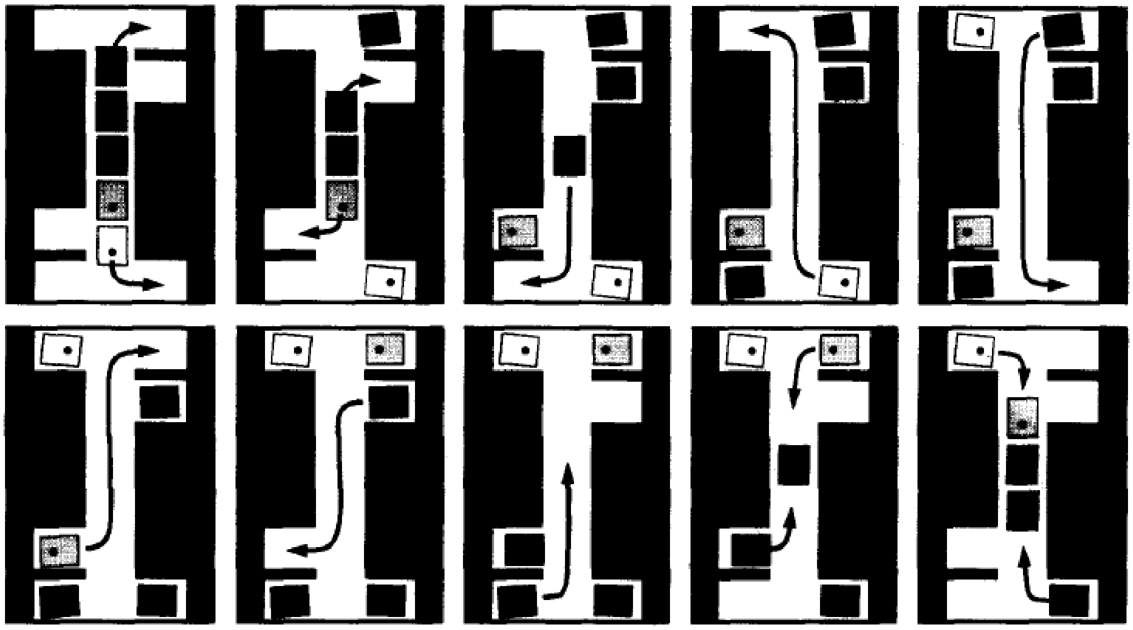


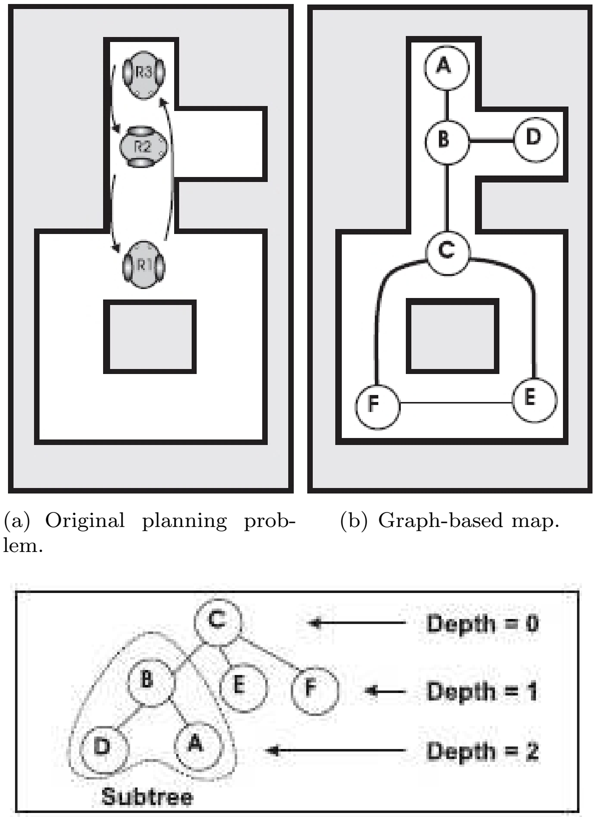
Figure 1: An illustration of a coordinated path generated by the super-graph approach of Švestka and Overmars, for 5 nonholonomic car-like robots (from [109]).

The second type of super-graph – the multi-level super-graph – reduces the size of the super-graph data structure by combining multiple nodes into a single node of the graph. This approach makes use of the concept of *subgraphs*. Whereas the nodes in a flat super-graph represent robots being located at particular nodes of G, the nodes in a multi-level super-graph represent robots being located in a subgraph of G. The restriction placed on node combinations is that the resultant subgraphs should not interfere with each other, meaning that the nodes in one subgraph cannot block paths in another subgraph. Experimental results have shown that the multi-level super-graphs are typically much smaller than the equivalent flat super-graphs.

Švestka and Overmars applied this approach to teams of up to 5 nonholonomic, car-like robots in simulation. An example of these results is shown in Figure 1, illustrating the feasibility of this approach for small-sized multi-robot teams. Nevertheless, this type of approach is appropriate only for relatively small numbers of robots. For much larger sizes of robot teams, decoupled approaches are necessary (see Section 4).

3.0.2 Spanning tree method (Peasgood, et al.)

Peasgood, et al., [87] present another roadmap-based planner for multi-robot teams. This approach is a multi-phase planner that uses a graph and spanning tree representation to create and maintain obstacle-free paths through the environment. Initially, a graph is created, in which the nodes are the robots' initial and goal positions, and the edges represent the connectivity of the node positions. An example is illustrated in Figure 2 (part a), in which the starting positions of the three robots (R1, R2, and R3) are (C, B, A), while the goal positions are (A, C, B). Figure 2 (part b) shows the graph-based map for this example. Then, a spanning tree of this graph is created, which is a connected subset of the original graph that includes all the nodes without cycles; Figure 2(part c) shows the example spanning tree. The root of this spanning tree is chosen to be the node that is closest to the geographic center of the map. Then, in the first phase of the approach, a plan is generated that moves the robots to the leaves of the spanning tree along collision-free paths, as shown in Figure 3 (part a). In the second phase, the robots are moved into positions where they can reach their goals without creating obstructions for other robots. This is accomplished by processing the robots in order according to the depth of their goals in the spanning tree. This is shown in Figure 3 (parts b and c). The third phase moves robots to the remaining unfilled goal locations, as shown in Figure 3 (part d). These three phases result in a sequence of motions that allow only one robot to move at a time. The



(a) Original planning problem.

(b) Graph-based map.

(c) Spanning tree for the graph representation.

Figure 2: An example multi-robot path planning problem using the spanning tree method of Peasgood, et al., along with the corresponding graph and spanning tree (from [87]).

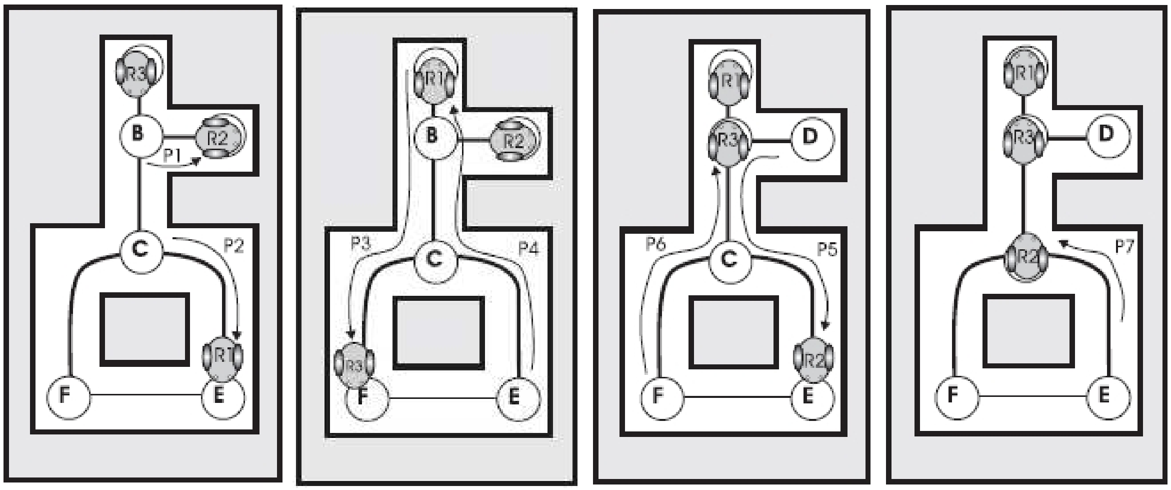
final phase of the process seeks to improve the quality of the concurrent plan by allowing robots to move simultaneously when doing so does not introduce any collisions.

Peasgood, et al., show that this algorithmic approach results in time complexity that is linear in the number of robots. To further improve the resulting path lengths, the authors propose a hybrid planning approach, which uses the regular multi-phase planner, but then also calls a decoupled planner (such as [15]), to attempt to find shorter path solutions. For smaller-sized robot teams (less than 20), the decoupled planner can often find better solutions. However, for larger-sized teams, the multi-phase approach is more time-efficient (increasingly so as the team size grows larger).

4 Decoupled Approaches

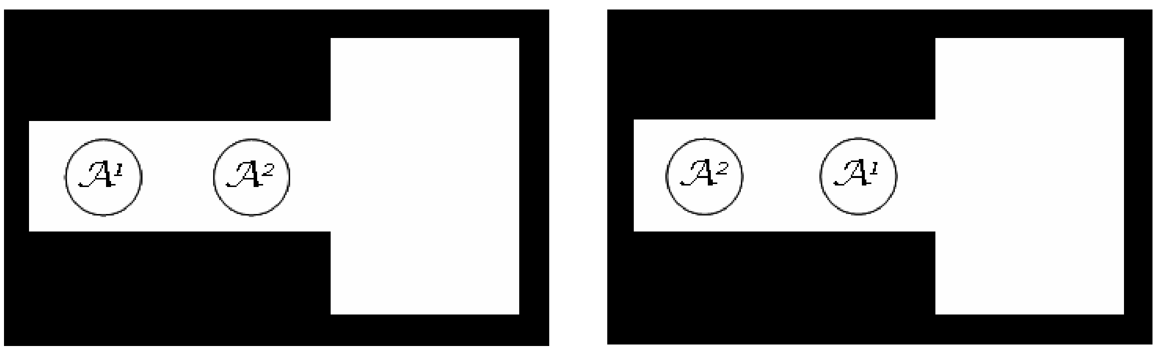
Decoupled approaches to multi-robot path planning typically trade off solution quality for efficiency by solving some aspects of the problem independently. There are many alternative ways of decomposing the planning problem. Most commonly, approaches plan individual paths for robots, followed by methods for handling collision avoidance. While decoupled approaches are typically more efficient than centralized approaches, they lose completeness. For instance, Figure 4 shows an example of a situation that is difficult for decoupled approaches to solve. In this situation, robots must exchange positions in a narrow corridor. While a centralized approach would find a solution in which the robots first move into the open space at the end of the corridor to exchange places, a decoupled approach will have difficulties discovering this solution.

Decoupled approaches are typically divided into two broad categories [58, 59]: prioritized planning and path coordination. *Prioritized planning* considers the motions of the robots one at a time, in priority order, calculating path information for the i^{th} robot by treating the previous $i-1$ robots as moving obstacles.



(a) Phase 1. (b) Phase 2a. (c) Phase 2b. (d) Phase 3.

Figure 3: The multi-phase solution of the multi-robot path planning problem in Figure 2, using the spanning tree method of Peasgood, et al. (from [87]).



Initial configuration Goal configuration

Figure 4: An example multi-robot path planning problem that is difficult for decoupled approaches to solve. Here, the robots must exchange positions by first moving into the open space at the end of the corridor. While a centralized approach can find this solution, most decoupled approaches would fail (recreated from [58]).

Path coordination, on the other hand, first plans independent paths for the robots separately, then seeks to plan their velocities so as to avoid collisions along those paths. The following subsections describe these approaches in more detail.

4.1 Prioritized Planning

The *prioritized planning* approach to multi-robot path planning was first proposed by Erdmann and Lozano-Peréz [29]. In this approach, priorities are assigned to each robot. These priorities could be assigned randomly, or they could be determined from motion constraints, in which more-constrained robots are given higher priority. A path is planned for the first robot using any single-robot path planning approach. The path for each successive robot, $\mathcal{A}^i$, then takes into account the plans for the previous robots $\mathcal{A}^1, \dots, \mathcal{A}^{i-1}$, treating these higher-priority robots as moving obstacles.

More specifically, in the prioritized planning approach of [29], the configuration space is extended to account for time, since the time-varying motions of previously-planned robots must be taken into account.

Configuration space-time is represented as a list of configuration space slices at particular times – specifically, those times corresponding to when a moving object changes its velocity. Motions between slices can then be interpolated via straight-line translations between these configuration space slices. The configuration space-time can be constructed in $O(m)$ time, where $m = nr$, for n edges in the environment and r time slices.

Paths through configuration space-time are computed using a visibility graph algorithm, which searches along a visibility graph consisting of the vertices of the configuration space obstacles (plus vertices for the start and goal positions), and the line-of-sight edges between the vertices. Planners using this method have time complexity $O(rn^3)$, although [29] also suggests a faster implementation. The prioritized planning approach has been demonstrated in several application domains, including the translation of multiple planar robots, as well as the motion of two-link planar articulated robot arms.

Other researchers who have studied prioritized path planning for multiple mobile robots include [32, 121, 16, 20]. Both Ferrari, et al. [32] and Warren [121] used a fixed priority scheme for the decoupled planner. In the work of Buckley [20], a heuristic is applied to assign higher priorities to robots that can move in a straight line to their target location. Chun, et al. [25] use this priority scheme to coordinate independently-generated schedules online, as the conflicts arise. The work of Azarm and Schmidt [6] considers all possible priority assignments, although the resulting approach is computationally complex. A more tractable method for finding and optimizing priority schemes for decoupled priority-based planners is presented by Bennewitz, et al., in [16]. The proposed approach performs a centralized, randomized search with hill-climbing (i.e., the A* search algorithm [73]) to search the space of prioritization schemes to find priority schemes that minimize the overall path length. The resulting priority scheme can then be applied in decoupled priority-based planners, such as Erdmann's method described above [29].

The advantage of prioritized planning approaches is that they reduce the problem from a single planning problem in a very high-dimensional space to a sequence of planning problems in much lower dimensional space. The disadvantage, as with all decoupled approaches, is that these approaches are not complete.

4.2 Path coordination

Path coordination techniques decouple the planning problem into path planning and velocity planning (e.g., [52]. The ideas are based on scheduling techniques for dealing with limited resources, inspired by the approaches developed for concurrent access to a database by multiple users [124]. In the current context, the shared resource is space. The decomposition of path and velocity planning provides a solution through the complexity barrier caused by the additional time dimension, and also provides solutions that are relevant when robots move along fixed paths.

In the path coordination approach, the path planning step first generates individual robot paths independently, using common single-robot path planners. The second step plans a velocity profile that each robot should follow along its path so as to avoid collisions with other robots. This approach is typically called *fixed-path* coordination, since the paths planned in the first step are not altered in the second step. Instead, only the velocities taken by the robots along the paths are varied.

In more detail (using the notation of [59]), assume that the path generated for each individual robot in the first step constrains robot $\mathcal{A}^i$ to follow a path $\tau_i : [0,1] \rightarrow \mathcal{C}^i_{free}$. Then, an m-dimensional *coordination diagram* $X = [0,1]^m$ for m robots is defined that is used to schedule the motions along their paths so that they do not collide [74]. In this diagram, the i^{th} coordinate represents the domain, $S_i = [0,1]$, of the path of robot $\mathcal{A}^i$. At state $(0,\ldots,0) \in X$, every robot is in its initial starting configuration. At state $(1,\ldots,1) \in X$, every robot is at its goal configuration. Within the coordination diagram, obstacles form obstacle regions X_{obs} that must be avoided. Any continuous, obstacle-free path, $h : [0,1] \rightarrow X$, for which $h(0) = (0,\ldots,0)$ and $h(1) = (1,\ldots,1)$, is a valid path that moves the robots from their starting positions to their goals. The objective, therefore, is to find $h : [0,1] \rightarrow X_{free}$, in which $X_{free} = X \setminus X_{obs}$. An example coordination diagram showing a valid path for the robots is illustrated in Figure 5.

Several authors have looked at variations of the path coordination approach. In [62], Lee and Lee use a similar idea to plan the motions of two robots. Griswold and Eem [41] take uncertainty of the moving obstacles into account while using the same principle for path planning. Pan and Luo [77] use the concept of *traversability vectors* to analyze the spatial relationship between the robot and moving obstacles, and develop a search algorithm to coordinate the robot motion. Rude [92] proposes a space-time representation

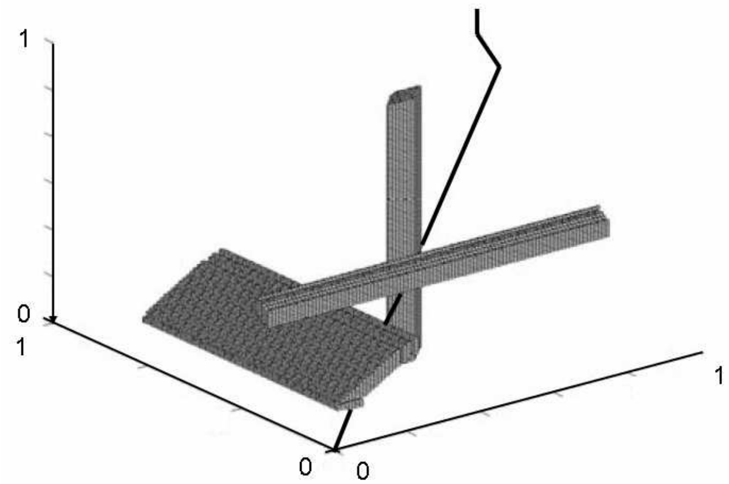


Figure 5: An example coordination diagram for three robots. Each axis represents the domain of an individual robot's path. The cylindrical objects are obstacles, and the path from (0, 0, 0) to (1, 1, 1) represents the coordinated velocity plans for moving all three robots to their goals without collisions (adapted from [43]).

for collision avoidance in pre-planned individual robot paths. In [43], Guo and Parker present a decentralized path coordination approach that also incorporates optimization issues into the planning, including a global performance measurement to minimize the weighted sum of the most expensive time to reach the goals and all idle time, as well as individual optimization goals for navigation over rough terrain. In [61], LaValle and Hutchinson consider multiple robots with independent goals and performance measures, and proposes algorithms optimizing a scalarizing function that is a weighted-average of individual performance functions. Other approaches to optimal motion planning have been proposed in [17, 18, 23, 61, 101, 88], sometimes in the context of robotic manipulator motion planning. In [22], one robot is randomly chosen to stop, and time delays are inserted to resolve potential collisions. Path coordination schedules, which are another form of velocity planning, are studied in [17, 102, 74]. A priority-based method using collision maps is presented in [78]. Extensions of the path coordination approach to coordination on roadmaps have been proposed by [61, 39].

While all of these decoupled approaches typically allow good solutions to the multi-robot path planning problem, they can lead to deadlocks, in which solutions cannot be found, even though they exist. In these cases, it may be possible to make use of a centralized planner for small portions of the original problem, in order to solve the immediate deadlock problem.

5 Motion Coordination

Closely related to the topic of multi-robot path planning is the issue of multi-robot *motion coordination*. Unlike multi-robot *path* planning or *path coordination* approaches, which plan and/or coordinate the complete paths of all of the robots in advance, techniques for *motion coordination* focus on decentralized, online approaches that allow robots to avoid and/or resolve conflict as the situation arises during path execution, such as through the use of *traffic control* rules. In traffic control applications, individual robots still have independent starting and goal positions, and must move so as to avoid conflict with each other. Even broader concepts of motion coordination seek to have the robots move according to some constraints on the team as a whole, such as can be seen in *formation keeping*, *flocking/swarming*, *target search/tracking*, *dispersion/aggregation*, and related topics. In these problems, the motions of individual robots are no longer independent of each other; instead, the group must move in synchrony according to pre-defined motion constraints for the entire team. The following subsections discuss some of the key research in these areas of motion coordination.

5.1 Traffic control

Traffic control approaches to multi-robot motion coordination typically predefine traffic or control rules that robots must obey as they move through the workspace. Individual robots often move along paths to their goals that they pre-plan in advance, based only on the individual robot goals. Then, as regions involving shared resources are reached (such as the space in an intersection), robots follow the traffic or control rules to coordinate their motions with other robots who also need access to the shared resources.

An early example of traffic control is the work of Grossman [42], which addresses the motion of large numbers of Automatic Guided Vehicles (AGVs) in a factory. Grossman defines three types of control possibilities: 1) restrict the roads so that there is a unique route between all starting and goal positions; 2) allow AGVs to select their own routes autonomously; and 3) control all AGVs' paths using centralized traffic control. Grossman shows that allowing AGVs to select their routes autonomously (option 2) is preferred over the highly suboptimal restriction of roads (option 1). Of course, as previously noted, the centralized approach (option 3) has high combinatorial complexity.

The problem of the autonomous coordination of paths (option 2) is formulated as follows. A set of r AGVs are allowed to follow unconstrained paths in two dimensions, on a *grid-iron* network of roadways, with n parallel roads along each axis. Each section of roadway between intersections is called an *arc*; in this formulation, there are $2n(n-1)$ arcs in the network. Each intersection of roadways is called a *node*, representing the locations of machine tools to be serviced by the robots. It is assumed that $1 \leq r \leq n^2 - 1$, and that all vehicles move at the same speed, v. Each AGV has the task of moving from a source location (i.e., starting position) to a sink location (i.e., a goal location). Defining S to be the average number of time steps per task for each AGV, the average throughput of all the AGVs together is $W = \frac{vr}{S}$. This throughput must exactly match the throughput of all the n^2 machine tools, leading to a requirement that the AGV speed must satisfy: $v = \frac{Sn^2}{r}$. The price of r AGVs is considered negligible in comparison to the price of the machine tools. Thus, the problem is formulated as the problem of optimizing the traffic control and the value of r so as to minimize v in an $n \times n$ grid-iron floor plan. The constraints on the traffic in this environment are as follows:

- At the end of each step, at most one AGV may be at each node.
- During each step, no two AGVs may pass on the same arc.
- All AGVs have equal priority.

Different policies are investigated, including a greedy policy and a benevolent policy. Simulation results show that the benevolent policy performs the best, with a performance close to the derived lower bound. This traffic policy requires the AGVs to follow these rules:

1. From the AGVs own (i, j) location, determine in which quadrant q the goal node (i', j') lies:
   - Quadrant 1 has $i' > i$ and $j' \geq j$.
   - Quadrant 2 has $j' > j$ and $i' \leq i$.
   - Quadrant 3 has $i' < i$ and $j' \leq j$.
   - Quadrant 4 has $j' < j$ and $i' \geq i$.
2. Depending on the value of q, try to move to an adjacent node:
   - If q is 1 then $(i + 1, j)$.
   - If q is 2 then $(i, j + 1)$.
   - If q is 3 then $(i - 1, j)$.
   - If q is 4 then $(i, j - 1)$.
3. If that node is blocked, add 1 to q and try Step 2 again.
4. If that node is blocked, add 1 to q and try Step 2 again.
5. If that node is blocked, add 1 to q and try Step 2 again.

6. If all adjacent nodes are blocked, then wait at the current node.

This policy leads to an overall counterclockwise flow of traffic through the workspace. Based on analysis and simulation results, the authors conjecture that this policy is the optimal policy for AGVs without memory or task trading.

There are many variants on the traffic control and conflict resolution theme [53, 5, 117, 125, 118, 65]. For example, in [53], Kato, et al., categorize the traffic rules into three types: 1) traffic rules to be applied to the current positions of the robot (examples include *passage zone*, *stop*, *slow*); 2) traffic rules to be applied to current positions and conditions (examples include *overtaking*, *avoiding obstacles*, *crossing intersections*); and, 3) traffic rules to ensure safety in case of accidents or failures. These rules are illustrated for robot teams operating in indoor hallway-types of settings.

In [5], Asama, et al., propose two basic rules for avoiding collisions:

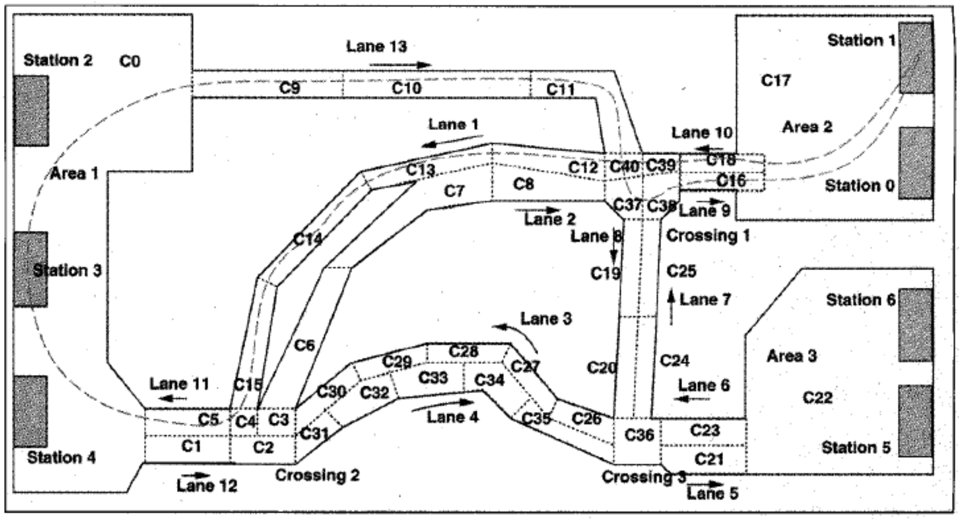
- "If the colliding robot is nearby to the front and approaching, then avoid from the left", and
- "If the colliding robot is nearby to the front and leaving, then stop for a while".

These rules are combined with a communication-based negotiation process that resolves conflicts by setting priorities based on the task requirements, the environmental situation, and robot performances. In the work of Yuta and Premvuti [125], robots move along pre-planned paths in network of roadways, which can involve conflicts at intersections. These deadlock situations at intersections are resolved through a "shunting" process, in which one robot, acting as a leader, devises a solution for moving robots through the intersection, and then broadcasts the instructions to the other robots for how to resolve the conflict. Another approach to conflict resolution is to use techniques from distributed computing, as illustrated in the work of Wang [117, 118], in which robots use a mutual exclusion protocol to compete for the right to move along certain pathways or to resolve conflicts at intersections.

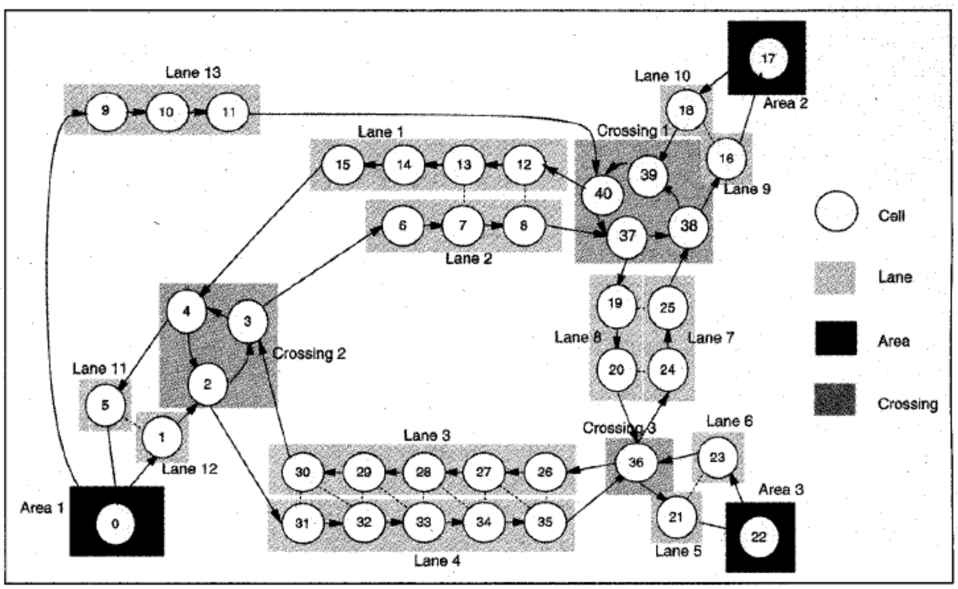
In [67], Lumelsky, et al., present a decentralized approach for motion planning that has robots plan and execute their paths "on the fly" in real time, resolving conflicts as they arise. The authors make an analogy to human cocktail parties, in which people do not plan optimal paths in advance, nor consult with others about their intended destinations; instead, they move toward their destinations while avoiding collisions as they go. Their approach is based on maze-searching techniques, and makes use of perpendicular bisectors and Voronoi diagrams [90] to allow robots to avoid collisions.

Another approach that is closely related to the decoupled path coordination research described earlier is the work of Alami, et al. [2, 1]. This online *plan-merging* paradigm does not require advance planning of all robot paths in advance. Instead, robots move as needed, coordinating their paths as new goal destinations arise. In this decentralized planning approach, robots also treat segments of their paths as shared resources. However, when a robot elaborates a new plan for itself, it must validate that plan within the current multi-robot context. This is done by collecting the plans from all the other robot team members via communication, and "merging" its own plan into the existing robot plans. This merging operation is done without affecting the plans of other robots, thus allowing them to continue on with their current executions. In this approach, the environment is represented as a topological graph of areas, routes, and crossings. Routes are composed of lanes with direction, thus setting up a type of traffic pattern through the environment. The motion planning approach makes use of a graph searching technique, planning dependency graphs, and synchronization points to coordinate the motions of the robots. Figure 6 illustrates the geometrical and topological planning space for this approach in a prototypical application.

More recent work in conflict resolution for multi-robot teams is the work of Pallottino, et al., [76], which considers a more realistic kinematic model of the robot dynamics, recognizing that most robots cannot stop instantly in order to avoid collisions. This model focuses on large numbers of robots (e.g., 70) operating closely in shared, open spaces. As with other techniques discussed to this point, this approach also assumes robots have independent starting positions and goal destinations. This approach is particularly relevant for applications of aerial vehicles flying at constant altitude. This work makes use of the concept of *reserved region*, which is an area for which a robot claims exclusive ownership. The control policy is defined for a set of discrete modes of operation, including a *hold* state in which a robot is stopped, a *straight* state in which the robot is moving forward without turning, and two *roll* states – one for mild turns and a second for tight turns. Control theoretic definitions of the motions of the robots in each state are given, and the policy is shown to be *safe*, meaning that it guarantees collision avoidance. Under certain conditions, the approach



(a) Geometrical representation.



(b) Topological representation.

Figure 6: Representations used for a prototypical application of the plan-merging paradigm of Alami, et al. (from [1]).

is also shown to have the property of *liveness*, meaning that all the robots are guaranteed to reach their destinations in finite time.

5.2 Reactive approaches

Reactive-style methods for coordination are useful in many applications, since they are fast, and can operate well in real-time. One common reactive method makes use of *potential fields* [55]. In the potential field approach, the robot moves through space as if it is being acted upon by a set of forces. Attractive forces pull the robot toward a goal destination, while repulsive forces push the robot away from obstacles and/or other robots. At each point in the configuration space, the robot moves along the vector representing the combined forces acting on that point in the configuration space. These concepts have been applied to various

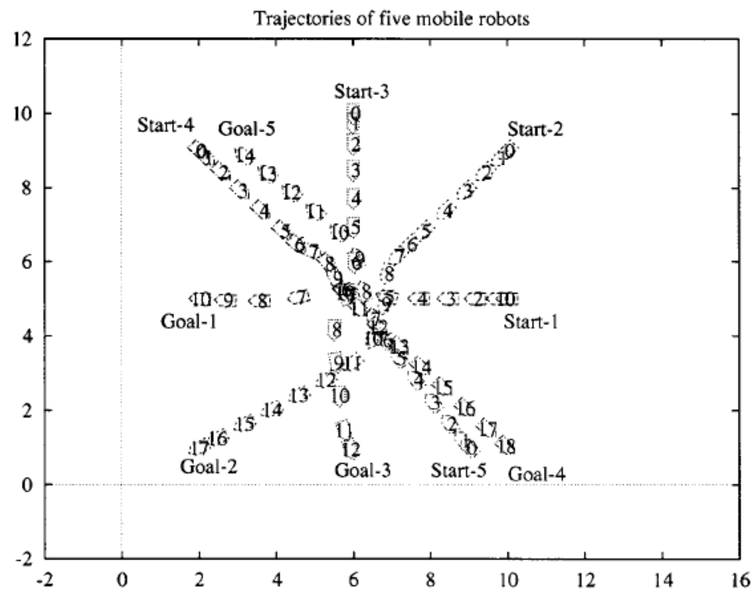


Figure 7: Example results for 5 robots in simulation performing adaptive navigation to avoid collisions, using the approach of Fujimori, et al. (from [34]).

multi-robot applications [121], including multi-robot soccer [63]. Other potential field approaches to multi-robot coordination include [27, 64, 120, 119]. A well-known issue in potential field methods, however, is their susceptibility to deadlock due to local minima in the potential field. Some techniques have been designed to overcome this shortcoming [10].

Other reactive approaches for collision avoidance based on local information include the work of Matarić [70], which proposes behavior-based avoidance rules in which robots either stop for a period of time or change directions. Similar rules were proposed by Arkin [4] and by Sugihara and Suzuki [107]. Shan and Hasegawa [99] present behavior-based techniques for avoiding robot collisions in narrow passages.

While all of the above techniques can work well for relatively unconstrained situations, they are not analyzed formally to provide guidance for setting the navigation parameters. On the other hand, a more formal method for determining reactive collision avoidance parameters is given by Fujimori, et al., in [34]. These authors propose a collision avoidance method based on an *adaptive navigation technique*, in which the navigation law is given by a first-order differential equation. Navigation of the robot to the goal and obstacle avoidance are handled by switching the direction angle adaptively. Robots are assigned priorities to determine which vehicles must yield to the others. The proper value of the direction angle is calculated theoretically, based on three robot modes of operation: *navigation mode*, in which the robot is moving toward the goal without interference; *cooperative avoidance mode*, in which the robot avoids other robots; and, *final mode*, when the robot is approaching near the goal. The approach has been implemented in simulation for up to five mobile robots, and on two physical robots. Figure 7 illustrates the type of motions generated by this approach in a five-robot simulation.

5.3 Coordinated Motion of Entire Team

A significant topic of current research is the control of robot motions to achieve a group objective, such as maintaining a formation while moving to a goal position, cooperatively tracking moving targets, collective coverage tasks, and so forth. Often, these topics are studied in the context of *swarm* robot systems, involving large numbers of homogeneous robots performing the same control algorithms. A complete survey of all the work in these areas is beyond the scope of this chapter. However, this section briefly outlines the areas of active research in this domain.

Many types of swarm behaviors have been studied, such as foraging, flocking, chaining, search, herding,

Table 1: Categories of swarm behaviors

Relative motion requirements	**Swarm Behaviors**
Relative to other robots	Formations [80, 107], Flocking, Natural herding (as in herds of cattle), Schooling, Sorting [13], Clumping [13], Condensation, Aggregation, Dispersion
Relative to the environment	Search [36], Foraging [7], Grazing, Harvesting, Deployment, Coverage, Localization, Mapping, Exploration
Relative to external agents	Pursuit, Predator-prey, Target tracking, Forced herding/shepherding (as in shepherding sheep)
Relative to other robots and the environment	Containment, Orbiting, Surrounding, Perimeter search
Relative to other robots, external agents, and the environment	Evasion, Tactical overwatch, Soccer [19, 115, 122, 104]

aggregation, and containment. The majority of these swarm behaviors deal with spatially distributed multi-robot motions, requiring robots to coordinate motions either (1) relative to other robots, (2) relative to the environment, (3) relative to external agents, (4) relative to robots and the environment, or (5) relative to all (i.e., other robots, external agents, and the environment). Table 1 categorizes swarm robot behaviors according to these groupings (see also [83]).

Much of the current research in swarm robotics is aimed at developing specific solutions to one or more of the swarm behaviors listed in Table 1. Some of these swarm behaviors have received particular attention, notably formations, flocking, search, coverage, and foraging. In general, most current work in the development of swarm behaviors is aimed at understanding the formal control theoretic principles that can predictably converge to the desired group behaviors, and remain in stable states. The following subsections outline research in some of these areas.

5.3.1 Flocking and Formations

Coordinating the motions of robots relative to each other has been a topic of interest in multiple mobile robot systems since the inception of the field. In particular, much attention has been paid to the flocking and formation control problems. The flocking problem can be viewed as a subcase of the formation control problem, requiring robots to move together along some path in the aggregate, but with only minimal requirements for paths taken by specific robots. Formations are more strict, requiring robots to maintain certain relative positions as they move through the environment. In these problems, robots are assumed to have only minimal sensing, computation, effector, and communications capabilities. A key question in both flocking and formation control research is determining the design of local control laws for each robot that generate the desired emergent collective behavior. Other issues include how robots cooperatively localize themselves to achieve formation control (e.g., [71, 72]), and how paths can be planned for permutation-invariant multi-robot formations (e.g., [56]).

Early solutions to the flocking problem in artificial agents were generated by Reynolds [91] using a rule-based approach. Similar behavior- or rule-based approaches have been used physical robot demonstrations and studies, such as in [70, 8]. These earlier solutions were based on human-generated local control rules that were demonstrated to work in practice. More recent work is based on control theoretic principles, with a focus on proving stability and convergence properties in multi-robot team behaviors. Examples of this work include [49, 14, 113, 31, 68, 37, 38, 110, 3].

5.3.2 Foraging and Coverage

Foraging is a popular testing application for multi-robot systems, particularly for those approaches that address swarm robotics, involving very large numbers of mobile robots. In the foraging domain, objects such as pucks or simulated food pellets are distributed across the planar terrain, and robots are tasked with collecting the objects and delivering them to one or more gathering locations, such as a home base. Foraging lends itself to the study of weakly cooperative robot systems, in that the actions of individual robots do not have to be tightly synchronized with each other. This task has traditionally been of interest in multi-robot systems because of its close analogy to the biological systems that motivate swarm robotics research. However, it also has relevance to several real-world applications, such as toxic waste cleanup, search and rescue, and demining. Additionally, since foraging usually requires robots to completely explore their terrain in order to discover the objects of interest, the *coverage* domain has similar issues to the foraging application. In coverage, robots are required to visit all areas of their environment, perhaps searching for objects (such as landmines) or executing some action in all parts of the environment (e.g., for floor cleaning). The coverage application has real-world relevance to tasks such as demining, lawn care, environmental mapping, and agriculture.

In foraging and coverage applications, a fundamental question is how to enable the robots to quickly explore their environments without duplicating actions or interfering with each other. Alternative strategies can include basic stigmergy [13], forming chains [28], and making use of heterogeneous robots [7]. Other research demonstrated in the foraging and/or coverage domain includes [86, 33, 116, 106, 94, 108, 21, 69].

5.3.3 Multi-Target Observation

The domain of multi-target observation requires multiple robots to monitor and/or observe multiple targets moving through the environment. The objective is to maximize the amount of time, or the likelihood, that the targets remain in view by some team member. The task can be especially challenging if there are more targets than robots. This application domain can be useful for studying strongly cooperative task solutions, since robots may have to coordinate their motions or the switching of targets to follow in order to maximize their objective. In the context of multiple mobile robot applications, the planar version of this testbed was first introduced by Parker in [82] as CMOMMT (Cooperative Multi-robot Observation of Multiple Moving Targets). Similar problems have been studied by several researchers, and extended to more complex problems such as environments with complex topography or three dimensional versions for multiple aerial vehicle applications. This domain is also related to problems in other areas, such as art gallery algorithms, pursuit evasion, and sensor coverage. This domain has practical application in many security, surveillance, and reconnaissance problems. Research applied to the multi-target observation problem in multi-robot systems includes [12, 123, 66, 60, 57, 51, 111].

6 Future Directions

Many open issues in multi-robot path planning and coordination remain. Current techniques typically do not scale well to very large numbers of robots (e.g., thousands), and many still have limitations for extensions to three dimensions (e.g., aerial robots). Many approaches have difficulty in highly stochastic environments; dynamic, online replanning of paths and coordination strategies is important in these contexts. Creating provably correct interaction strategies in these domains is an ultimate goal. Developing path planning and motion coordination techniques that incorporate practical motion and sensing constraints of physical robots is still an open issue. Integrating these techniques onto physical robots remains uncommon, due to the practical need to integrate these path planning and coordination algorithms with complete sensing, navigation, and reasoning systems, as well as the practical difficulty of experiments involving large numbers of fallible robots. Certainly, ongoing work is addressing these important issues in multi-robot path planning and coordination; it is likely that the research community will be successful in developing solutions to extend the state of the art in this domain.

Of course, understanding how to coordinate the motions of robots in a shared workspace has both practical and scientific interest. From a practical perspective, many real-world applications can potentially benefit from the use multiple mobile robot systems. Example applications include container management in ports [1], extra-planetary exploration [105], search and rescue [50], mineral mining [98], transportation [112], industrial and household maintenance [84], construction [103], hazardous waste cleanup [81], security [30, 44], agriculture [89], and warehouse management [46]. To date, relatively few real-world implementations of these multi-robot systems have occurred, primarily due to the complexities of multiple robot systems and the relative newness of the supporting technologies. Nevertheless, many proof-of-principle demonstrations of physical multi-robot systems have been achieved, and the expectation is that these systems will find their way into practical implementations as the technology continues to mature. Because of the fundamental need for motion coordination for all applications of multi-robot systems, the work described in this chapter is of critical importance.

From a scientific perspective, understanding interactions between multiple autonomous robots might lead to insights in understanding other types of complex systems, from natural interactions in biology and social systems to engineered complex systems involving multiple interacting agents. Because multi-robot systems operate in stochastic and unpredictable settings, the study of the interaction dynamics in these settings can lead to discoveries of broader impact to a wide range of complex nonlinear systems.

References

[1] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi-robot cooperation in the MARTHA project. *IEEE Robotics and Automation Magazine*, 5(1):36–47, 1998.

[2] R. Alami, F. Robert, F. Ingrand, and S. Suzuki. Multi-robot cooperation through incremental plan-merging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2573–2578, 1995.

[3] G. Antonelli and S. Chiaverini. Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, 22(6):1285–1292, 2006.

[4] Arkin. Cooperation without communication: multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9:351–364, 1992.

[5] H. Asama, K. Ozaki, H. Itakura, A. Matsumoto, Y. Ishida, and I. Endo. Collision avoidance among multiple mobile robots based on rules and communication. In *Proceedings of IEEE/RJS International Conference on Intelligent Robots and Systems*, 1991.

[6] K. Azarm and G. Schmidt. Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3526–3533, 1997.

[7] T. Balch. The impact of diversity on performance in robot foraging. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 92–99. ACM Press, 1999.

[8] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, December 1998.

[9] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot motion planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22, 1992.

[10] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *International Journal of Robotics Research*, 16(6):759–774, 1997.

[11] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research*, 20(6):628–649, 1991.

[12] R. W. Beard, T. W. McLain, and M. Goodrich. Coordinated target assignment and intercept for unmanned air vehicles. In *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 2002.

[13] R. Beckers, O. Holland, and J. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R. Brooks and P. Maes, editors, *Proceedings of the 14th International Workshop on Synthesis and Simulation of Living Systems*, pages 181–189. MIT Press, 1994.

[14] C. Belta and V. Kumar. Abstraction and control for groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, 2004.

[15] M. Bennewitz, W. Burgard, and S. Thrun. Optimizing schedules for prioritized path planning of multi-robot systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 271–276, 2001.

[16] M. Bennewitz, W. Burgard, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobiel robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002.

[17] Z. Bien and J. Lee. A minimum-time trajectory planning method for two robots. *IEEE Transactions on Robotics and Automation*, 8:414–418, 1992.

[18] J. E. Bobrow. Optimal robot path planning using the minimum-time criterion. *IEEE Transactions on Robotics and Automation*, 4(4):443–450, 1988.

[19] B. Browning, J. Bruce, M. Bowling, and M. Veloso. STP: Skills, tactics and plays for multi-robot control in adversarial environments. *IEEE Journal of Control and Systems Engineering*, 219:33–52, 2005.

[20] S. J. Buckley. Fast motion planning for multiple moving robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 322–326, 1989.

[21] Z. J. Butler, A. A. Rizzi, and R. L. Hollis. Cooperative coverage of rectilinear environments. In *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 2000.

[22] S. Carpin and E. Pagello. A distributed algorithm for multi-robot motion planning. In *Proceedings of the Fourth European Conference on Advanced Mobile Robotics*, 2001.

[23] C. Chang, M. J. Chung, and B. H. Lee. Collision avoidance of two general robot manipulators by minimum delay time. *IEEE Transactions on Robotics and Automation*, 24(3):517–522, 1994.

[24] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, Cambridge, MA, 2005.

[25] L. Chun, Z. Zheng, and W. Chang. A decentralized approach to the conflict-free motion planning for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1544–1549, 1999.

[26] C. M. Clark, S. M. Rock, and J.-C. Latombe. Motion planning for multiple mobile robot systems using dynamic networks. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 4222–4227, 2003.

[27] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos. A feedback stabilization adn collision avoidance scheme for multiple independent non-point agents. *Automatica*, 42(2):229–243, 2006.

[28] A. Drogoul and J. Ferber. From Tom Thumb to the Dockers: Some experiments with foraging robots. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 451–459, 1992.

[29] M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.

[30] H. R. Everett, R. T. Laird, D. M. Carroll, G. A. Gilbreath, T. A. Heath-Pastore, R. S. Inderieden, T. Tran, K. J. Grant, and D. M. Jaffee. Multiple Resource Host Architecture (MRHA) for the Mobile Detection Assessment Response System (MDARS). In *SPAWAR Systems Technical Document 3026, Revision A*, 2000.

[31] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9), 2004.

[32] C. Ferrari, E. Pagello, J. Ota, and T. Arai. Multirobot motion coordination in space and time. *Robotics and Autonomous Systems*, 25:219–229, 1998.

[33] M. Fontan and M. Matarić. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 15(5):815–822, 1998.

[34] A. Fujimori, M. Teramoto, P. Nikiforuk, and M. Gupta. Cooperative collision avoidance between multiple mobile robots. *Journal of Robotic Systems*, 17(7):347–363, 2000.

[35] K. Fujimura. *Motion Planning in Dynamic Environment.* Computer Science Workbench. Springer-Velag, Tokyo, 1991.

[36] Douglas Gage. Randomized search strategies with imperfect sensors. In *Proceedings of SPIE Mobile Robots VIII*, pages 270–279. SPIE, September 1993.

[37] V. Gazi. Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Transactions on Robotics*, 21(6):1208–1214, 2005.

[38] S. S. Ge and C.-H. Fua. Queues and artificial potential trenches for multirobot formations. *IEEE Transactions on Robotics*, 21(4):646–656, 2005.

[39] R. Ghrist, J. M. O'Kane, and S. M LaValle. Pareto optimal coordination on roadmaps. In *Proceedings of the Workshop on Algorithmic Foundations of Robotics*, pages 185–200, 2004.

[40] H. H. Gonzalez-Banos, D. Hsu, and J. C. Latombe. Chapter: Autonomous mobile robots: Sensing, control, decision-making, and applications. In *Motion Planning: Recent Developments*. CRC Press, 2006.

[41] N. C. Griswold and J. Eem. Control for mobile robots in the presence of moving objects. *IEEE Transactions on Robotics and Automation*, 6(2):263–268, 1990.

[42] D. D. Grossman. Traffic control of multiple robot vehicles. *IEEE Transactions on Robotics and Automation*, 5(5):491–497, 1988.

[43] Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.

[44] Y. Guo, L. E. Parker, and R. Madhavan. Towards collaborative robots for infrastructure security applications. In *Proceedings of International Symposium on Collaborative Technologies and Systems*, pages 235–240, 2004.

[45] E. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on System Science and Cybernetics*, SSC-4(2):100–107, 1968.

[46] C. Hazard, P. R. Wurman, and R. D'Andrea. Alphabet soup: A testbed for studying resource allocation in multi-vehicle systems. In *Proceedings of AAAI Workshop on Auction Mechanisms for Robot Coordination*, pages 23–30, 2006.

[47] J. E. Hopcroft, J. T. Schwartz, and M Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-Hardness of the "Warehouseman's Problem". *The International Journal of Robotics Research*, 3(4):76–88, 1984.

[48] Y. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219–291, 1992.

[49] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), 2003.

[50] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *Proceedings of the 8th International Conference on Advanced Robotics*, pages 193–200, 1997.

[51] B. Jung and G. Sukhatme. Tracking targets using multiple mobile robots: The effect of environment occlusion. *Autonomous Robots*, 13(3):191–205, 2002.

[52] K. Kant and S. W. Zucker. Toward efficient trajectory planning: the path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.

[53] S. Kato, S. Nishiyama, and J. Takeno. Coordinating mobile robots by applying traffic rules. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1535–1541, July 1992.

[54] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[55] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[56] S. Kloder and S. Hutchinson. Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics*, 22(4):650–665, 2006.

[57] A. Kolling and S. Carpin. Multirobot cooperation for surveillance of multiple moving targets – a new behavioral approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1311–1316. IEEE, 2006.

[58] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[59] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[60] S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 731–736. IEEE, 1997.

[61] S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14:912–925, 1998.

[62] B. H. Lee and C. S. Lee. Collision-free motion planning of two robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(1):21–32, 1987.

[63] B.-J. Lee, S.-O. Lee, and G.-T.-Park. Trajectory generation and motion tracking for the robot soccer game. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, pages 1149–1154, 1999.

[64] J. Lee and Z. Bien. Collision-free trajectory control for multiple robots based on neural optimization network. *Robotica*, 8:185–194, 1990.

[65] C. F. Lin and W. H. Tsai. Motion planning for multiple robots with multi-mode operations via disjunctive graphs. *Robotica*, 9:393–408, 1991.

[66] S. Luke, K. Sullivan, L. Panait, and G. Balan. Tunably decentralized algorithms for cooperative target observation. In *Proceedings of the fourth international joint conference on Autonomous Agents and Multiagent Systems*, pages 911–917. ACM Press, 2005.

[67] V. J. Lumelsky and K. R. Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Autonomous Robots*, 4(1):121–135, 1997.

[68] J. A. Marshall, M. E. Broucke, and B. R. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11), 2004.

[69] M. Matarić. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 19(2-3):323–336, 1997.

[70] M. J. Matarić. Designing emergent behaviors: From local interactions to collective intelligence. In J. Meyer, H. Roitblat, and S. Wilson, editors, *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 432–441, Honolulu, Hawaii, 1992. MIT Press.

[71] A. I. Mourikis and S. I. Roumeliotis. Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5):917–931, 2006.

[72] A. I. Mourikis and S. I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4):666–681, 2006.

[73] N.J. Nilsson. *Principles of Artificial Intelligence*. Springer, 1982.

[74] P. A. O'Donnell and T. Lozano-Perez. Deadlock-free and collision-free coordination of two robot manipulators. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 484–489, 1989.

[75] C. O'Dunlaing and C. K. Yap. A retraction method for planning the motion of a disc. *Journal of Algorithms*, 6:104–111, 1982.

[76] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics*, 23(6):1170–1183, 2007.

[77] T. J. Pan and R. C. Luo. Motion panning for mobile robots in a dynamic environment. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 578–583, 1990.

[78] S. Park and B. Lee. A new analytical representation to robot path generation with collision avoidance through the use of the collision map. *International Journal of Control, Automation, and Systems*, 4(1):77–86, 2006.

[79] L. E. Parker. A robot navigation algorithm for moving obstacles. Master's thesis, The University of Tennessee, Knoxville, 1988.

[80] L. E. Parker. Designing control laws for cooperative agent teams. In *Proceedings of the IEEE Robotics and Automation Conference*, pages 582–587. IEEE, 1993.

[81] L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[82] L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing*, 5(1):5–19, 1999.

[83] L. E. Parker. Chapter 40: Multiple mobile robot systems. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*. Springer, 2008.

[84] L. E. Parker and J. Draper. Robotics applications in maintenance and repair. In S. Nof, editor, *Handbook of Industrial Robotics*, pages 1023–1036. Wiley Publishers, 2nd edition, 1999.

[85] D. Parsons and J. Canny. A motion planner for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 8–13, 1990.

[86] K. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, 2002.

[87] M. Peasgood, C. Clark, and J. McPhee. A complete and scalable strategy for coordinating multiple robots within roadmaps. *IEEE Transactions on Robotics*, to appear, 2008.

[88] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *The International Journal of Robotics Research*, 24(4):295–310, 2005.

[89] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz. The demeter system for automated harvesting. In *Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems*, 1999.

[90] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[91] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21:25–34, 1987.

[92] M. Rude. Collision avoidance by using space-time representations of motion processes. *Autonomous Robots*, 4:101–119, 1997.

[93] M. R. K. Ryan. Graph decomposition for efficient multi-robot path planning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2003–2008, 2007.

[94] P. Rybski, S. Stoeter, C. Wyman, and M. Gini. A cooperative multi-robot approach to the mapping and exploration of mars. In *Proceedings of AAAI/IAAI-97*. AAAI, 1997.

[95] G. Sanchez and J. C. Latombe. On delaying collision checking in PRM planning: application to multi-robot coordination. *The International Journal of Robotics Research*, 21(1):5–26, 2002.

[96] J. T. Schwartz and M. Sharir. On the 'piano movers' problem: Iii. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal obstacles. *International Journal of Robotics Research*, 2(3):46–75, 1983.

[97] J. T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artificial Intelligence Journal*, 37:157–169, 1988.

[98] G. Shaffer and A. Stentz. A robotic system for underground coal mining. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 633–638, 1992.

[99] L. Shan and T. Hasegawa. Space reasoning from action observation for motion planning of multiple robots: mutual collision avoidance in a narrow passage. *Journal of the Robotics Society of Japan*, 14:1003–1009, 1996.

[100] M. Sharir. Algorithmic motion planning. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, 2nd Ed.* Chapman and Hall/CRC Press, 2004.

[101] Z. Shiller and H. H. Lu. Robust computation of path constrained time optimal motions. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 144–149, 1990.

[102] T. Simeon, S. Leroy, and J. Laumond. Path coordination for multiple mobile robots: a resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, 24(1):42–49, 2002.

[103] R. Simmons, S. Singh, D. Hershberger, J. Ramos, and T. Smith. First results in the coordination of heterogeneous robots for large-scale assembly. In *Proc. of the ISER Seventh International Symposium on Experimental Robotics*. Springer-Verlag, 2000.

[104] P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.

[105] A. Stroupe, A. Okon, M. Robinson, T. Huntsberger, H. Aghazarian, and E. Baumgartner. Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. *Autonomous Robots*, 20(2):113–123, 2006.

[106] K. Sugawara and M. Sano. Cooperative behavior of interacting simple robots in a clockface arranged foraging field. In H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, editors, *Distributed Autonomous Robotic Systems*, pages 331–339. Springer, 2002.

[107] K. Sugihara and I. Suzuki. Distributed algorithms for formation of goemetric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.

[108] S. Sun, D. Lee, and K. Sim. Artificial immune-based swarm behaviors of distributed autonomous robotic systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3993–3998. IEEE, 2001.

[109] P. Svestka and M. Overmars. Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23:125–152, 1998.

[110] P. Tabuada, G. Pappas, and P. Lima. Motion feasibility of multi-agent formations. *IEEE Transactions on Robotics*, 21(3):387–392, 2005.

[111] Z. Tang and U. Ozguner. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21(5):898–908, 2005.

[112] C. Thorpe, T. Jochem, and D. Pomerleau. The 1997 automated highway free agent demonstration. In *Proceedings of IEEE Conference on Intelligent Transportation System*, pages 496–501, 1997.

[113] C. M. Topaz and A. L. Bertozzi. Swarming patterns in two-dimensional kinematic model for biological groups. *SIAM Journal of Applied Math*, 65(1):152–174, 2004.

[114] P. Tournassoud. A strategy for obstacle avoidance and its application to multi-robot systems. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1224–1229, 1986.

[115] M. Veloso, P. Stone, and K. Han. The CMUnited-97 robotic soccer team: Perception and multiagent control. *Robotics and Autonomous Systems*, 29(2-3):133–143, 1999.

[116] I. Wagner, M. Lindenbaum, and A. M. Bruckstein. Mac vs. PC – determinism and randomness as complementary approaches to robotic exploration of continuous unknown domains. *International Journal of Robotics Research*, 19(1):12–31, 2000.

[117] Jing Wang. Fully distributed traffic control strategies for many-AGV systems. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, pages 1199–1204, 1991.

[118] Jing Wang and Gerardo Beni. Distributed computing problems in cellular robotic systems. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, pages 819–826, 1990.

[119] P. K. C. Wang. Interaction dynamics of multiple autonomous mobile robots in bounded spatial domains. *International Journal of Control*, 50(6):2109–2124, 1989.

[120] P. K. C. Wang. Interaction dynamics of multiple mobile robots with simple navigation strategies. *Journal of Robotic Systems*, 6(1):77–101, 1989.

[121] C. W. Warren. Multiple robot path coordination using artificial potential fields. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 500–505, 1990.

[122] T. Weigel, J.-S. Gutmann, M. Dietl, A. Kleiner, and B. Nebel. CS Freiburg: coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 5(18):685–699, 2002.

[123] B. B. Werger and M. J. Matarić. Broadcast of local eligibility for multi-target observation. In L. E. Parker, G. Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 347–356. Springer, 2000.

[124] M. Z. Yannakakis, C. H. Papadimitriou, and H. T. Kung. Locking policies: Safety and freedom for deadlock. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 286–297, 1979.

[125] S. Yuta and S. Premvuti. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent robots and systems*, pages 1566–1574, July 1992.

Books and Reviews

References

[1] T. Arai and J. Ota. Motion planning of multiple mobile robots. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent robots and systems*, pages 1761–1768, July 1992.

[2] T. Arai, E. Pagello, and L. E. Parker. Editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.

[3] Y. Cao, A. Fukunaga, and A. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23, 1997.

[4] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[5] H. Choset Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113-126, 2001.

[6] D. Nardi, A. Farinelli, L. Iocchi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(5):2015–2028, 2004.

[7] L. E. Parker. Current research in multirobot teams. *Artificial Life and Robotics*, 7(2-3):1–5, 2005.

[8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.