Path Selection and Coordination for Multiple Robots via Nash Equilibria

Steven M. LaValle Seth A. Hutchinson
lavalle@cs.uiuc.edu seth@cs.uiuc.edu
Dept. of Electrical and Computer Engineering
University of Illinois, Urbana, IL 61801

Abstract

We present a method for analyzing and selecting time-optimal coordination strategies for n robots whose configurations are constrained to lie on a C-space roadmap (which could for instance represent a Voronoi diagram). We consider independent objective functionals, associated with each robot, together in a game-theoretic context in which maximal Nash equilibria represent the favorable strategies. Within this framework additional criteria, such as priority or the amount of sacrifice one robot makes, can be applied to select a particular equilibrium. An algorithm that determines all of the maximal Nash equilibria for a given problem is presented, along with several computed examples for two and three robots.

1 Introduction

There has been considerable interest from the motion planning community in the problem of coordinating multiple robots. Most traditional approaches to coordination either form a global objective that constructs a path through a joint configuration space, simultaneously solving each robot goal (*centralized planning*), or first generate plans using independent objectives, and then consider the interactions with other robots (*decoupled planning*). We argue that one of the fundamental issues of coordination is the incorporation of multiple, independent objectives into a single framework that directly models the competition or conflict that occurs. This consideration is naturally provided by a game-theoretic analysis, which leads to the determination of maximal Nash equilibria as the smallest set of useful coordination strategies.

With the centralized view, a multiple-robot planning problem can be viewed as a single robot planning problem by considering the combined configuration space of the robots [2, 10]. As an example, the problem of planning the motion of n disks in the plane was solved in [15] by performing a critical curve analysis on the configuration spaces of the disks, and planning a collision-free path through free cells in the combined configuration space.

A variety of decoupled planning approaches exist. Prioritized planning has been proposed, in which plans are generated sequentially for robots of decreasing priority, given the plans of the higher-priority robots [5, 10]. In [9] robot plans are generated independently, and modifications are made to accommodate robot interaction. In [13] robot paths are independently determined, and a coordination diagram is used to plan a collision-free trajectory along the paths.

Our approach can be considered somewhere between centralized planning and decoupled planning. We compute a roadmap for each independent robot (decoupled), and consider coordinating the paths and motions of the robots on the roadmap (centralized).

Our primary interest, however, is in a notion of optimality that applies to multiple robots and guides the selection of coordination strategies. Traditionally, the amount of sacrifice that each robot makes individually to accomplish its goals is not usually taken into account. For instance, it might be that one robot's goal is nearby, while the other robot has a distant goal. Optimizing the joint goal might produce a plan that is good for the robot that has the distant goal; however, the execution cost for the other robot would be hardly considered.

Instead of seeking a single, *optimal* coordination strategy, we show that there exists a natural partial ordering on the space of strategies, yielding a search for the set of *maximal* coordination strategies. For any other coordination strategy that can be considered, there will exist at least one maximal strategy that is clearly better or equivalent, and the set of all maximal strategies is typically small. Within this framework additional criteria, such as priority or the amount of sacrifice one robot makes, can be applied to select a particular strategy. If the same tasks are repeated and priorities change, then only a different maximal strategy would be selected, as opposed to re-exploring the space of coordination strategies.

These maximal strategies furthermore satisfy the Nash equilibrium condition, which intuitively means that each robot is satisfied with the outcome, given the actions taken by the other robots. This concept is borrowed from game theory, which is a well-studied subject that deals precisely with the multiple objective problem in which some form of competition or conflict occurs. From a modern viewpoint, game theory represents a generalization of decision theory, optimal control theory, and games considered in AI contexts. Particularly, an extensive amount of material can be found in optimization and control literature; numerous references can be found in [1]. The Nash equilibrium concept in our context generalizes the notion of optimality to a multiple robot case.

In Section 2, we formally define our multi-robot coordination problem. Section 3 introduces the game-theoretic formulation of the robot objectives and solution space. A coordination space is considered, and a

1050-4729/94 $03.00 © 1994 IEEE

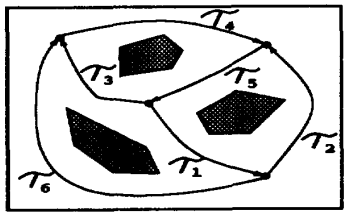

Figure 1. An example roadmap structure with oriented constant-speed paths.

key proposition based on path homotopy is presented that states that there is only one strategy per path class worth considering in a coordination diagram (this is utilized by our algorithm). Section 4 presents an algorithm that can generate the complete set of maximal Nash equilibria, given discretized representations of coordination diagrams. The algorithm was used to compute the results presented in Section 5. Section 6 presents some conclusions.

2 Problem Definition

We have chosen to work with a roadmap representation for our coordinated motion planning problem since several general methods exist for producing a roadmap in configuration space [10]. The *visibility graph* approach generates a roadmap by connecting certain vertices of the boundary of the free configuration space, $\mathcal{C}_{free}$, and is primarily suitable for two-dimensional polygonal $\mathcal{C}$-space planning problems. The topological *retraction* operation has been used in a roadmap generation approach that continuously retracts $\mathcal{C}_{free}$ onto its Voronoi diagram [14]. Other roadmap methods are described in [4, 6]. Roadmap coordination has also been recently considered in [16].

We use $\mathcal{A}_i$ to represent a robot, and assume that the position of each robot is represented by a point in a common configuration space, $\mathcal{C}$. We consider a *roadmap*, $\mathcal{R}$, to be a one-dimensional connected subset of $\mathcal{C}_{free}$ (see Figure 1). The roadmap is parametrized by a collection of regular curves, $\mathcal{T}$, such that for each $\tau_i \in \mathcal{T}$, $\tau_i : [0,1] \rightarrow \mathcal{R}$. We refer to each τ_i as a *segment* of $\mathcal{R}$. We assume without loss of generality that each parametrization is of constant speed. The endpoints of some paths coincide in $\mathcal{R}$, to form a network. For instance, in Figure 1, $\tau_3(1) = \tau_4(0) = \tau_6(1)$.

We assume that a roadmap representation has been a priori determined for a given problem for a workspace with static obstacles. The implication of working with this common roadmap is that the configurations in $\mathcal{R}$ must be collision free for each robot independently. We can alternatively consider independent roadmaps for each of the robots without significantly altering our approach.

A problem is specified by providing an initial configuration, $\mathbf{q}^i_{init} \in \mathcal{R}$, and a goal configuration $\mathbf{q}^i_{goal} \in \mathcal{R}$ for each robot, $\mathcal{A}_i$. As a minor extension one could additionally consider initial and goal configurations in $\mathcal{C}_{free}$; a roadmap can be extended in a straightforward manner to include the particular initial and goal states of a given problem [10].

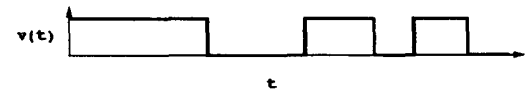

Figure 2. A sample velocity function $v(t) \in \mathcal{V}$.

The objective of each robot can be expressed as:

1. Determine a continuous path $\pi_i : [0,1] \rightarrow \mathcal{R}$ in which $\pi_i(0) = \mathbf{q}^i_{init}$ and $\pi_i(1) = \mathbf{q}^i_{goal}$.
2. Determine a velocity function $v_i(t)$ that specifies the motion of $\mathcal{A}_i$ along the path.
3. The functions π_i and $v_i(t)$ must be chosen so that the robots do not collide with each other (the roadmap already guarantees that the robots do not collide with obstacles).
4. Minimize the time required to reach $\mathbf{q}^i_{goal}$.

A position on $\mathcal{R}$ along π_i at time t is given by $\pi_i(\int_0^t v_i(t)dt/l(\pi_i))$, in which $l(\pi_i)$ is the arc length[1] of π_i. For $v_i(t)$ to be valid, we must have $\int_0^\infty v_i(t)dt = l(\pi_i)$, so that the robot is at $\pi_i(1)$ as $t \rightarrow \infty$.

To simplify the analysis, we place some restrictions on the choices for π_i and $v_i(t)$. Since paths are constrained to lie on $\mathcal{R}$, we choose paths, π_i, by selecting a route through the roadmap from $\mathbf{q}^i_{init}$ to $\mathbf{q}^i_{goal}$. Such a route can be specified by a sequence of segments, $\{\tau_{i_1}, \ldots, \tau_{i_k}\}$, such that $\mathbf{q}^i_{init}$ is contained in the segment parametrized by τ_{i_1}, and $\mathbf{q}^i_{goal}$ is contained in the segment parameterized by τ_{i_k}. We form π_i from a sequence of segments by constructing a constant-speed parametric function that traverses each segment. We denote the set of all valid paths for $\mathcal{A}_i$ as Π_i.

As an example (recall Figure 1), suppose that $\mathbf{q}^i_{init} = \tau_2(1/2)$, $\mathbf{q}^i_{goal} = \tau_3(1/2)$, $\{\tau_2, \tau_5, \tau_3\}$ is a connected segment sequence, $\tau_2(1) = \tau_5(0)$, $\tau_5(1) = \tau_3(0)$ and the segments are of equal arc length. The corresponding path, π_i, from $\mathbf{q}^i_{init}$ to $\mathbf{q}^i_{goal}$ is constructed as

$$\pi_i(s) = \begin{cases} \tau_2(\frac{1}{2} + 2s) & \text{if } 0 \leq s < \frac{1}{4} \\ \tau_5(2(s - \frac{1}{4})) & \text{if } \frac{1}{4} \leq s < \frac{3}{4} \\ \tau_3(2(s - \frac{3}{4})) & \text{if } \frac{3}{4} \leq s < 1 \end{cases} . \quad (1)$$

We allow segment sequences that revisit the same points in $\mathcal{R}$, such as $\{\tau_2, \tau_5, \tau_3, \tau_5, \tau_1\}$. If the segment sequence causes the robot to visit some $\pi_i(0)$ twice without visiting $\pi_i(1)$ (or equivalently, $\pi_i(1)$ twice without $\pi_i(0)$), then we travel a fixed distance into the segment before returning. A path of this type is useful when one robot needs to get past another on $\mathcal{R}$ (Figure 8 of Section 5 shows an example of this).

We assume that all robots are capable of moving at the same constant velocity. We do not model robot dynamics (with the exception of avoiding collision), and

[1] We use arc length to refer to the distance along a path.

consequently allow robots to instantaneously switch between being stopped and maintaining the fixed velocity (see Figure 2). This yields a set of valid velocity functions for $\mathcal{A}_i$, which we denote by $\mathcal{V}_i$. Others have considered trajectory coordination of robots along paths incorporating dynamics [3, 8], and we believe the ideas presented in this paper can be extended through considering additional constraints and forming a suitable class of velocity functions, $\mathcal{V}_i$.

3 Coordination Equilibria

The objective of each robot $\mathcal{A}_i$ is to determine a path and velocity function that minimizes the time to reach $\mathbf{q}^i_{goal}$. If each robot $\mathcal{A}_i$ was trying to meet its objective without the presence of other robots, we would be confronted with a standard optimization problem in which we simply select the shortest path from $\mathbf{q}^i_{init}$ to $\mathbf{q}^i_{goal}$, and move the robot at its fixed velocity from start to finish.

When considering interaction with other robots, however, this choice of path and velocity function may produce a collision. Furthermore, a collision-free velocity function might not even exist for the optimal arc-length path. In the coordination scenario, each robot is faced with this difficulty, even though independent optimal choices exist.

For this problem we seek a notion of "optimality" that guides the selection of coordination strategies, given that we have a vector of independent objectives (as opposed to a single objective in a standard optimization setting). We can formulate the problem defined in Section 2 as a single-stage game by defining the following three components:

Players	$\mathcal{A}_1, \ldots, \mathcal{A}_n$
Strategy Space	$\Gamma = \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_n$
Loss Functionals	$L_1(\gamma), \ldots, L_n(\gamma)$

The i^{th} component of the strategy space is defined as $\Gamma_i = \Pi_i \times \mathcal{V}_i$, which is the cartesian product of the set of all paths and the set of all valid velocity functions for $\mathcal{A}_i$. We call each $\gamma \in \Gamma$ a strategy for the game, and some $\gamma_i \in \Gamma_i$ a strategy for player i.

The loss functionals encode the objectives of the players. We take $L_i(\gamma)$ to be the time required for player $\mathcal{A}_i$ to reach $\mathbf{q}^i_{goal}$ if strategy γ is implemented and $\mathcal{A}_i$ does not collide with another robot. If $\mathcal{A}_i$ collides with another robot, then we declare $L_i(\gamma) = \infty$. Other criteria, such as total distance or energy could be combined into a loss functional.

We can consider a partial ordering, $\preceq$, on the space of strategies as follows: $\gamma \preceq \gamma'$ if $L_i(\gamma) \leq L_i(\gamma')$ for each i. If it further holds that $L_j(\gamma) < L_j(\gamma')$ for some j, we say that γ is *better* than γ'. The strategies are *equivalent* if $L_i(\gamma) = L_i(\gamma')$ for each i. Two strategies, γ and γ', are *incomparable* if there exists some i, j such that $L_i(\gamma) < L_i(\gamma')$ and $L_j(\gamma) > L_j(\gamma')$. Hence we can consider γ to be either better than, *worse* than, equivalent to, or incomparable to γ'. We say that γ^* is a *maximal* strategy if for all $\gamma \neq \gamma^*$ such that γ and γ^* are not incomparable, we have $\gamma^* \preceq \gamma$.

Only the maximal coordination strategies, γ^*, are reasonable to consider since for any other $\gamma \in \Gamma$ there exists a maximal strategy that is better. In general game theory, however, it could be the case that player i can further reduce its loss by selecting a different γ_i^*, given the γ_j^* chosen by the other players, even though γ^* is maximal. In other words, player i might not be satisfied with the outcome, given the γ_i taken by the other players. However, for the loss function considered for this multiple robot coordination problem and a maximal strategy $\gamma^* = (\gamma_1^* \ldots \gamma_n^*)$, the following holds for each i and each $\gamma_i \in \Gamma_i$:

$$L_i(\gamma_1^*, \ldots, \gamma_i^*, \ldots \gamma_n^*) \leq L_i(\gamma_1^*, \ldots, \gamma_i, \ldots \gamma_n^*). \quad (2)$$

This is referred to as a *Nash equilibrium*, and we will hereafter refer to a γ^* as a *maximal Nash equilibrium*. The goal of our algorithm is to determine all of these maximal Nash equilibria for a given problem.

For the remainder of this section we discuss how velocity functions are determined for a given set of paths, $\{\pi_1, \ldots, \pi_n\}$. The key result is Proposition 1, which states that very few velocity functions are candidates for maximal Nash equilibria. This is exploited by our algorithm, which is presented in Section 5.

We begin by considering a representation referred to in robotics literature as a *coordination space* [3, 13]. We denote this space with $\mathcal{S}$, which represents the unit n-cube spanned by the coordinates $(s_1, s_2, \ldots, s_n)$. Each point in $\mathcal{S}$ corresponds to a fixed position, $\pi_i(s_i)$, for each $\mathcal{A}_i$.

We define the obstacle region in $\mathcal{S}$ as the set of n-tuples, $(s_1, s_2, \ldots, s_n)$ such that:

$$\mathcal{A}_i(\pi_i(s_i)) \bigcap \mathcal{A}_j(\pi_j(s_j)) \neq \emptyset \text{ for } i \neq j. \quad (3)$$

This states that some pair of robots, $\mathcal{A}_i$ and $\mathcal{A}_j$, collide at configurations $\pi_i(s_i)$ and $\pi_j(s_j)$. We will use the notation $\mathcal{S}_{valid}$ to denote the closure of the obstacle-free subset of $\mathcal{S}$. Figure 6 shows a discretized example of $\mathcal{S}$ for which $n = 2$. The shaded region indicates places in $\mathcal{S}$ in which two robots collide.

The objective is to provide a continuous path, denoted by f, from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$, whose image is contained in $\mathcal{S}_{valid}$. This corresponds to moving each robot from $\pi_i(0)$ to $\pi_i(1)$, without colliding with other robots. Robots are allowed to touch (at the boundary of $\mathcal{S}_{valid}$); however, the obstacle region in $\mathcal{S}$ can be appropriately modified if this aspect is undesirable.

Since we made the assumption that the robots can either move at some fixed velocity along the path or halt, there are some restrictions imposed on f:

- f must be nondecreasing with respect to each s_i
- f is a polygonal curve
- Each linear piece of f must correspond to having some number of robots moving at their fixed velocity, and some robots stopped.

We next consider path classes in $\mathcal{S}_{valid}$. Two paths f_1 and f_2 in $\mathcal{S}_{valid}$ are *homotopic* (with endpoints fixed) if there exists a continuous map $h : [0, 1] \times [0, 1] \rightarrow \mathcal{S}_{valid}$ with $h(s, 0) = f_1(s)$ and $h(s, 1) = f_2(s)$ for all $s \in [0, 1]$, and $h(0, t) = h(0, 0)$ and $h(1, t) = h(1, 0)$ for all $t \in$

$[0,1]$. This homotopy determines an equivalence relation, $\simeq$, among the monotonic paths from $(0,0,\ldots,0)$ to $(1,1,\ldots,1)$ in $\mathcal{S}_{valid}$. We will use $[f]$ to denote the equivalence class of paths that contains f. Note that since f is monotone, the path classes defined here do *not* represent the fundamental group from homotopy theory.

Using these path classes we have the following proposition:

Proposition 1 *There exists an f with corresponding γ such that for every other path $f' \in [f]$ and its corresponding strategy γ', we have $\gamma' \preceq \gamma$ (i.e., each path class contains a best strategy)*

Overview of Proof For $n = 2$: Suppose to the contrary that there exists some $f' \in [f]$ such that γ and γ' are incomparable. Consider the points in $\mathcal{S}_{valid}$ at which f and f' intersect, which includes $(0,0)$ and $(1,1)$. Between each consecutive pair of intersection points, say (x_1, y_1) and (x_2, y_2), a path, σ, can be found from (x_1, y_1) to (x_2, y_2) (using the fact that $[f] = [f']$ and monotonicity) whose losses L_1 and L_2 (in terms of time) are less than or equal to the losses associated with the corresponding pieces of f and f' from (x_1, y_1) to (x_2, y_2). The strategy that corresponds to piecing together the σ for each consecutive pair of intersection points is better than both γ and γ', which is a contradiction.

For $n > 2$: Suppose again that γ and γ' are incomparable. This implies that for some pair of indices i, j, we have $L_i(\gamma) < L_i(\gamma')$ and $L_j(\gamma) > L_j(\gamma')$. Consider S^2 as the coordination space generated by $s_i s_j$. The projections of f and f' onto S^2 lie in distinct path classes by the $n = 2$ part of this proposition. This can be used to show that $[f] \neq [f']$ in $\mathcal{S}_{valid}$, which is a contradiction. □

4 A Coordination Algorithm

In this section we describe the components of the following algorithm which obtains the maximal Nash equilibria:

1. Initialize strategy list, G and let $k = n$
2. For each index set in Σ_k, execute Steps 3 and 4 using the corresponding paths $\pi_1, \ldots, \pi_n$
3. If there does not exist some strategy in $\gamma \in G$ such that $L_i(\gamma) \leq l(\pi_i)$ for each i, then execute Step 4 (otherwise skip to the next index set in Σ_k)
4. For the choice of paths, construct a discretized representation of $\mathcal{S}_{valid}$, and consider the addition of one maximal strategy from each path class $[f]$ to G
5. If Step 4 was executed at least once (or until some maximum k), then let $k \leftarrow k + 1$ and go to Step 2
6. Return G as the set of maximal Nash equilibria

Throughout the execution of the algorithm, we maintain and update a set of strategies, G. Initially, G is empty, and in Step 6, G will result in the set of maximal Nash equilibria.

The path choices for each robot are iteratively considered in a systematic manner. Let $l(\pi_i)$ represent the arc length of the path π_i. First, we use an efficient algorithm that repeatedly returns the n^{th} shortest path for $\mathcal{A}_i$ (minimizing $l(\pi_i)$), given the previous shortest $n-1$ paths. Detailed analysis of n^{th} shortest path algorithms is provided in [12].

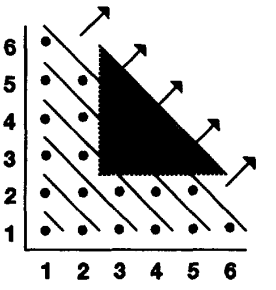

Figure 3. The progression of alternative path consideration in our algorithm.

We describe the consideration of paths for $n = 2$, and a similar idea applies when $n > 2$. Let $[i, j]$ represent the choice of the i^{th} best path for $\mathcal{A}_1$ and the j^{th} best path for $\mathcal{A}_2$. For each execution of Step 2, with value k, the algorithm considers all choices of $[i, j]$ for which $i+j = k$, denoted by Σ_k. Figure 3 shows the progression of paths that we consider (ignore the shaded triangle for now). The horizontal axis represents path choices for $\mathcal{A}_1$, and the vertical axis represents path choices for $\mathcal{A}_2$. The first execution of Step 2 considers $[1, 1]$, then $[1, 2]$ and $[2, 1]$. The third execution considers $[1, 3]$, $[2, 2]$, and $[3, 1]$, etc. Proposition 2 indicates when the algorithm will halt.

In Step 4, we construct a discretized representation of $\mathcal{S}$, which was considered for two robots in [10, 13]. This is constructed by dividing each path π_i into a sequence of w_i path segments, determined by the intervals $\delta_{i,k_i} = [s_{i,k_i}, s_{i,k_i+1}]$, with $k_i = 0, \ldots, w_i - 1$, $s_{i,0} = 0$ and $s_{i,w_i} = 1$. In our experiments, we take the intervals δ_{i,k_i} to correspond to motions of equal length along each path π_i. This subdivision transforms the continuous $\mathcal{S}$ space into an n-dimensional array of $\prod_i w_i$ closed rectangular cells. A cell, represented by $\delta_{0,k_0} \times \cdots \times \delta_{n,k_n}$, is classified as $EMPTY$ if

$$\{\mathcal{A}_i(\pi_i(s_i)) | s \in \delta_{i,k_i}\} \cap \{\mathcal{A}_j(\pi_j(s_j)) | s \in \delta_{j,k_j}\} = \emptyset \quad (4)$$

for all $1 \leq i, j \leq n$ with $i \neq j$. The cell is classified as $FULL$ otherwise. Hence, the robots are guaranteed not to collide in an $EMPTY$ cell, while they may collide in a $FULL$ cell. Each path f is constrained not to pass through a $FULL$ cell. We note that completeness for a given problem (i.e., whether all path classes are guaranteed be found) will in general depend on the selected resolution for the discretization.

Once the discretized representation of $\mathcal{S}$ has been built, the next step is to determine the maximal strategy for each path class $[f]$. We accomplish this by generating one representative path from each $[f]$, and iteratively shortening it until a maximal path $f^* \in [f]$ is obtained. The path f^* minimizes the time for each player to reach $\mathbf{q}^i_{goal}$, given the path class $[f]$.

Each time a new strategy, γ, is found in Step 4, one of the following actions is taken:

- If $\gamma \preceq \gamma'$ for some $\gamma' \in G$, then γ is added to G, and all strategies worse than γ are removed

- If $\gamma' \preceq \gamma$ for some $\gamma' \in G$, then γ is discarded
- If γ is incomparable to every element in G, then γ is added to G

If the goal is to determine equivalent maximal strategies which use different paths (as opposed to one representative for each equivalent loss), then strategies that are equivalent to some element in G are added.

The following proposition provides the motivation for Step 5:

Proposition 2 *If there exists a game strategy for each $\mathcal{A}_i$ that allows it to reach q^i_{goal} optimally (independently of other robots), then the presented algorithm provides the complete set of maximal Nash equilibria in Γ.*

Overview of Proof Recall that we assumed the robots move at some fixed velocity. Without loss of generality, we will assume that this fixed velocity is one unit per second, measured in the same units as path length, $l(\pi_i)$. Also, assume further that $n = 2$. For each choice of paths, $[i, j]$ (yielding π^i_1 and π^j_2) $l(\pi^i_1)$ and $l(\pi^j_2)$ represent a lower bound on the loss values ($L_1(\gamma)$ and $L_2(\gamma)$) that could be obtained from any γ represented in $\mathcal{S}$. If there exists a game strategy, γ^1 that allows $\mathcal{A}_1$ to reach q^1_{goal} in time $l(\pi^1_1)$, and another strategy γ^2 that allows $\mathcal{A}_2$ to reach q^2_{goal} in time $l(\pi^1_2)$, then only a finite number of pairs $[i, j]$ exist that can potentially produce better or incomparable strategies.

If G in the algorithm represents the maximal strategies in Γ, then by inspection of (2) it can be seen that every element of G is a Nash equilibrium.

The same argument holds for $n > 2$. □

For instance, suppose we find a strategy γ from path choice $[1, 1]$ (meaning the shortest path for each player) such that $L_1(\gamma) < l(\pi^3_1)$ and $L_2(\gamma) < l(\pi^3_2)$. Then the shaded region in Figure 3 does not need to be considered because all strategies obtained from path choices in that region will be worse than (or equivalent to) γ.

5 Examples

In this section we present examples that were computed using the algorithm in Section 4. We have presently considered 2- and 3-robot coordination problems. Using Allegro Common Lisp on a SPARC 10 workstation, the computation time ranged from seconds to about an hour. Each of these examples involves a few segments, which makes the robot motions tightly constrained. More complex roadmaps tend to generate similar types of robot interaction illustrated here, or offer alternative paths that avoid collision altogether. To simplify the collision detection, (4), we used circular robots in these examples. For each strategy, several frames are shown that indicate the location of the robots during the strategy execution.

Figures 4 and 5 show the two unique-loss maximal equilibria for an "H"-shaped roadmap coordination problem in which two robots attempt to reach opposite corners. The black and white circles in the two figures indicate the positions of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. The black and white triangles in Figures 4.1 and 5.1 indicate the goal positions.

Figure 4 is time-optimal for $\mathcal{A}_1$, but $\mathcal{A}_2$ must make a sacrifice by waiting. Figure 5 is time-optimal for $\mathcal{A}_2$, but $\mathcal{A}_1$ waits instead. Figure 6 shows the corresponding discretized coordination diagram generated from the optimal length paths for $\mathcal{A}_1$ and $\mathcal{A}_2$, π^1_1 and π^1_2, which yields two paths that are homotopically distinct. For this problem, these paths generate the two maximal Nash equilibria.

Figure 7 shows a maximal Nash equilibria (among several) for three robots. The black, white, and shaded circles in the two figures indicate the positions of $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, respectively. Figure 7 is time-optimal for $\mathcal{A}_3$.

Figure 8 shows a maximal equilibrium for a problem in which three robots must reverse their ordering to accomplish the goal. Robots $\mathcal{A}_2$ and $\mathcal{A}_3$ wait in side segments, while $\mathcal{A}_1$ passes.

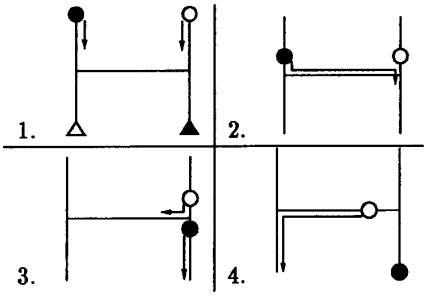

Figure 4. One of two maximal Nash equilibria

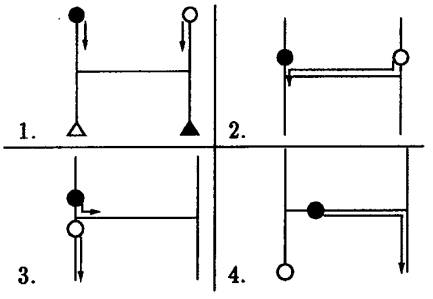

Figure 5. One of two maximal Nash equilibria

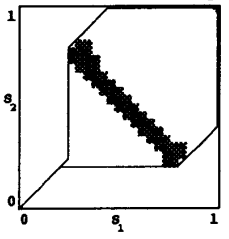

Figure 6. The coordination diagram (with $w_1 = w_2 = 25$)

6 Conclusions

We have determined that maximal Nash equilibria sufficiently characterize the interesting strategies that

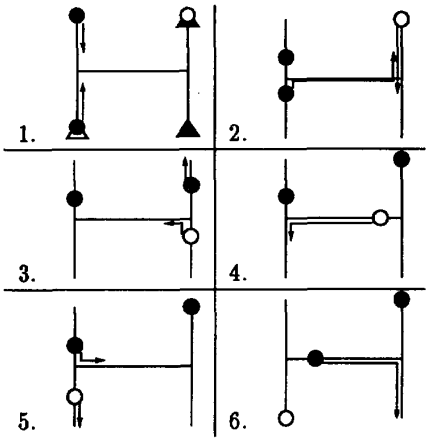

Figure 7. One of several maximal Nash equilibria

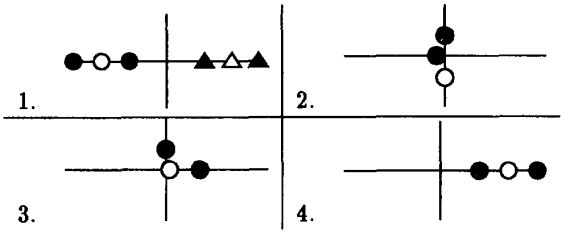

Figure 8. One of two maximal Nash equilibria

exist in the roadmap coordination problem. Furthermore, through the analysis of path classes in $\mathcal{S}$ and the systematic progression of path choices, we presented an algorithm that can find all the maximal Nash equilibria for a given problem.

After obtaining these equilibria, one can execute a specific strategy by agreeing on a certain global policy. For instance, we could decide to select a strategy that minimizes the amount of sacrifice that a robot will have to make, in comparison to its optimal strategy in the presence of no other robots. Or it could be the case that one robot, say A_i, is designated with highest priority, and a strategy, γ, is selected that minimizes $L_i(\gamma)$. If this global policy changes over time (assuming that the coordination problem is repeated) then a different equilibrium will be selected, as opposed to reinvestigating the space of all possible strategies.

We believe the general game-theoretic principles discussed in this paper apply to other robotic tasks as well. A variety of robot tasks can be expressed in optimization terms, and the concept of Nash equilibria provide a useful extension of optimality to multiple robots. Furthermore, additional structure from modern game theory can be incorporated. As an example, stochastic uncertainty in sensing and control can be introduced [7, 11].

7 Acknowledgements

This work was sponsored by NSF under grant #IRI-9110270. Helpful comments were made by Steve Sullivan. We are grateful for the detailed comments made by the anonymous reviewers.

References

[1] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Academic Press, London, 1982.

[2] J. Barraquand, B. Langlois, and J. C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Trans. Syst., Man, Cybern.*, 22(2):224–241, 1992.

[3] Z. Bien and J. Lee. A minimum-time trajectory planning method for two robots. *IEEE Trans. on Robotics and Automation*, 8(3):414–418, June 1992.

[4] R. A. Brooks. Solving the find-path problem by good representation of free space. *IEEE Trans. Syst., Man, Cybern.*, 13(3):190–197, 1983.

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

[6] J. F. Canny. *The Complexity of Robot Motion Planning.* MIT Press, Cambridge, MA, 1988.

[7] K. Y. Goldberg. *Stochastic Plans for Robotic Manipulation.* PhD thesis, Carnegie-Mellon, Pittsburgh, PA, August 1990.

[8] Y.-R. Hu and A. A. Goldenberg. Dynamic control of multiple coordinated redundant robots. *IEEE Trans. Syst., Man, Cybern.*, 22(3):568–574, May/Jun 1992.

[9] V. G. Kountouris and H. E. Stephanou. Dynamic modularization and synchronization for intelligent robot coordination: The concept of functional time-dependency. In *IEEE International Conference on Robotics and Automation*, pages 508–513, Sacramento, CA, April 1991.

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

[11] S. M. LaValle and S. A. Hutchinson. An objective-based stochastic framework for manipulation planning. In review for *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1994.

[12] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart, and Winston, New York, NY, 1976.

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

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

[15] J. T. Schwartz and M. Sharir. On the piano movers' problem: III. Coordinating the motion of several indepepdent bodies. *Int. J. of Robot. Res.*, 2(3):97–140, 1983.

[16] T. Shibata and T. Fukuda. Coordinative behavior by genetic algorithm and fuzzy in evolutionary multi-agent system. In *IEEE International Conference on Robotics and Automation*, pages 1:760–765, 1993.