Exact Pareto-Optimal Coordination of Two Translating Polygonal Robots on an Acyclic Roadmap

Hamidreza Chitsaz Jason M. O'Kane Steven M. LaValle

Dept. of Computer Science
University of Illinois
Urbana, IL 61801 USA
{chitsaz, jokane, lavalle}@cs.uiuc.edu

***Abstract*— *We present an algorithm that computes the complete set of Pareto-optimal coordination strategies for two translating polygonal robots in the plane. A collision-free acyclic roadmap of piecewise-linear paths is given on which the two robots move. The robots have a maximum speed and are capable of instantly switching between any two arbitrary speeds. Each robot would like to minimize its travel time independently. The Pareto-optimal solutions are the ones for which there exist no solutions that are better for both robots. The algorithm computes exact solutions in time $O(mn^2 \log n)$, in which m is the number of paths in the roadmap, n is the number of coordination space vertices. An implementation is presented.*

I. Introduction

Collision-free coordination of multiple bodies is a fundamental problem that has received significant attention over the last couple of decades. Popular examples of multibody systems include reconfigurable robots [5], [9], [12], [23] and autonomous guided vehicles (AGVs). In this paper, we address cases in which each body is treated as a separate robot and a roadmap (network of paths) has been computed for each robot. Each roadmap avoids collisions with workspace obstacles, but as robots travel along their respective roadmaps, collisions may occur. The task is to schedule the motions of the robots in a way that avoids collisions between robots while minimizing the time taken to reach goals.

Previous approaches to multiple-robot motion planning are often categorized as *centralized* or *decoupled*. A centralized approach typically constructs a path in a composite configuration space, which is formed by the Cartesian product of the configuration spaces of the individual robots (e.g., [2], [3], [19]). A decoupled approach typically generates paths for each robot independently, and then considers the interactions between the robots (e.g., [1], [7], [17]). In [4], [16], [20] robot paths are independently determined, and a coordination diagram is used to plan a collision-free trajectory along the paths. In [13], [22], an independent roadmap is computed for each robot, and coordination occurs on the Cartesian product of the roadmap path domains. The suitability of one approach over the other is usually determined by the tradeoff between computational complexity associated with a given problem, and the amount of completeness that is lost. In some applications, such as the coordination of AGVs, the roadmap might represent all allowable mobility for each robot.

Suppose that all paths in a roadmap are parameterized with constant speed, and each robot is capable switching instantaneously between being at rest and moving at some fixed speed (obviously, this assumes transients are negligible, which is only true in some applications). What is a reasonable notion of optimality in this case? Minimizing the average time robots take to reach their goal? Minimizing the time that the last robot takes? Optimal coordination using such scalar criteria has been considered long ago in [11], [15], [21]. The problem with scalarization is that it eliminates many interesting coordination strategies, possibly even neglecting optimality for some robots [14].

We are interested instead in finding all Pareto-optimal [18] coordination strategies by treating coordination as a multiobjective optimization problem. Each robot has an independent criterion, which leads to a vector of costs. Each Pareto-optimal strategy is one for which there exists no strategy that would be better for all robots. The approach can be considered as filtering out all of the motion plans that are not worth considering, and presenting the user with a small set of the best alternatives. Within this framework additional criteria, such as priority or the amount of sacrifice one robot makes, can be applied to automatically select a particular motion plan. If the same tasks are repeated and priorities change, then one only needs to select an alternative minimal plan, as opposed to re-exploring the entire space of motion strategies.

In this paper, we introduce an exact algorithm for finding all Pareto-optimal coordination strategies for two polygonal robots, each translating along a fixed roadmap of paths. In [14], an approximate algorithm was presented for any number of robots and path types by developing a Dijkstra-like algorithm that finds all Pareto-optimal solutions. To the best of our knowledge, up to now there have been no exact algorithms for computing Pareto-optimal coordination strategies.

II. Problem Formulation

Suppose we have two polygonal robots $\mathcal{R}_1$ and $\mathcal{R}_2$. For brevity, let $i = 1, 2$ throughout the following sections. We assume $\mathcal{R}_i$ only translates in the plane. Therefore the configuration space of $\mathcal{R}_i$ is $\mathbb{R}^2$. We also assume that we are given a fixed roadmap $\mathcal{M}$ on which $\mathcal{R}_i$ moves. The roadmap $\mathcal{M}$ specifies a connection graph and a collection of continuous piecewise-linear paths associated with its edges in $\mathbb{R}^2$. More precisely, $\mathcal{M} = (\mathcal{G}, \gamma)$, in which the graph $\mathcal{G}$ consists of a finite number of 0-dimensional vertices $\mathcal{V}$ and 1-dimensional edges $\mathcal{E}$ assembled as follows. Each edge e is homeomorphic to the closed interval $[0, 1]$ attached to $\mathcal{V}$ along its boundary points $\{0\}$ and $\{1\}$ [1]. We assume $\mathcal{G}$ is simple, i.e. has no loops. Note that $\mathcal{G}$ need not necessarily be connected, which can be used to represent the case where each robot has its own roadmap.

In the definition of roadmap $\mathcal{M}$, $\gamma : \mathcal{G} \to \mathbb{R}^2$ is a continuous map such that for each edge $e \in \mathcal{E}$, $\gamma|_e : \mathcal{G} \to \mathbb{R}^2$ is a piecewise-linear path in the plane. The length of each such piecewise-linear path gives a measure of the length of the corresponding edge in $\mathcal{G}$. Note that $\mathcal{G}$ in this manner becomes a metric space with metric d which conforms to the meaning of length in the plane.

We are also given an initial and a goal configuration $C_i^{init}, C_i^{goal} \in \mathcal{G}$ for robot $\mathcal{R}_i$. Now the problem is to give an algorithm to find all Pareto-optimal coordinations for the two robots $\mathcal{R}_1$ and $\mathcal{R}_2$ moving on $\mathcal{G}$ from the initial configuration C_1^{init} and C_2^{init} to the goal configuration C_1^{goal} and C_2^{goal} respectively. In the following, we define the meaning of all those terms.

A *coordination* is a continuous, and piecewise smooth path in $\mathcal{G} \times \mathcal{G}$ which avoids collision between robots. Precisely, a continuous path $C : [0, 1] \to \mathcal{G} \times \mathcal{G}$ from (C_1^{init}, C_2^{init}) to (C_1^{goal}, C_2^{goal}) is a coordination for $\mathcal{R}_1$ and $\mathcal{R}_2$ if for all $t \in [0, 1]$, robot $\mathcal{R}_1$ at $\gamma(C_1(t))$ does not collide with robot $\mathcal{R}_2$ at $\gamma(C_2(t))$, in which $C(t) = (C_1(t), C_2(t))$. We use the term coordination for both the above function and its image wherever there are no ambiguities.

Finally, we are given a cost functional J that separately measures the time that each robot takes to reach its goal, under a particular coordination. Thus, it specifies a partial order on the set of all coordinations $\mathcal{C}$. Each minimal element in this partial order is called a Pareto-optimal coordination.

III. Algorithm Presentation

A. Basic concepts

Cost functional: As it is stated in Section II, we have an explicit cost functional J. Particularly, J_i denotes the amount of time that it takes $\mathcal{R}_i$ to reach its goal and stop. This time depends on the speed of $\mathcal{R}_i$ and the length of its path. We have so far introduced length in Section II. Without loss of generality, let us assume that our robots have a maximum speed of 1. Under this assumption, the distance function $d(x, y)$ gives the minimum amount of time that it takes $\mathcal{R}_i$ to go from x to y on $\mathcal{G}$.

To specify J, first we define a metric d^∞ in $\mathcal{G}\times\mathcal{G}$ which gives the minimum amount of time that it takes to get both $\mathcal{R}_1$ and $\mathcal{R}_2$ from (x_1, x_2) to (y_1, y_2). It is naturally defined by $d^\infty : ((x_1, x_2), (y_1, y_2)) \mapsto \max(d(x_1, y_1), d(x_2, y_2))$.

[1]We place upon $\mathcal{G}$ the topology given by the endpoint identifications.

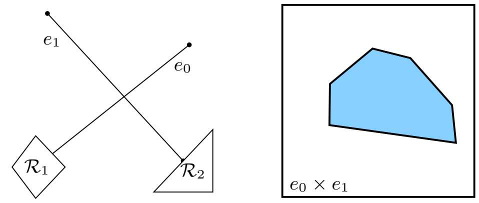


Fig. 1. A pair of path segments and their coordination cell.

It is easy to verify that d^∞ is actually a metric. Second, let $\mathcal{L}^\infty$ be the functional that gives the length of each continuous path in $\mathcal{G} \times \mathcal{G}$ according to d^∞.

Now for each coordination C, we specify $J = (J_1, J_2)$ in the following three cases:

- Robot $\mathcal{R}_1$ reaches its goal sooner than $\mathcal{R}_2$. Thus there is $t_0 \in [0, 1)$ such that $\forall\ t_0 \le t \le 1 : \quad C(t) = (C_1^{goal}, C_2(t))$ and $C_2(t_0) \neq C_2^{goal}$. For the least such t_0 we define $J(C) = (\mathcal{L}^\infty(C|_{[0,t_0]}), \mathcal{L}^\infty(C))$.
- Robot $\mathcal{R}_2$ reaches its goal sooner than $\mathcal{R}_1$. Thus there is $t_0 \in [0, 1)$ such that $\forall\ t_0 \le t \le 1 : \quad C(t) = (C_1(t), C_1^{goal})$ and $C_1(t_0) \neq C_1^{goal}$. For the least such t_0 we define $J(C) = (\mathcal{L}^\infty(C), \mathcal{L}^\infty(C|_{[0,t_0]}))$.
- Otherwise, both robots reach their goals simultaneously. We define $J(C) = (\mathcal{L}^\infty(C), \mathcal{L}^\infty(C))$.

Coordination cell: Since $\mathcal{G}$ consists of 0-dimensional and 1-dimensional cells, $\mathcal{G} \times \mathcal{G}$ is a cube-complex. In fact, $\mathcal{G} \times \mathcal{G}$ consists of a number of 2-dimensional cells appropriately pasted to each other along their boundary edges and vertices. Each such 2D cell, $D = e_r \times e_s$, in which $e_r, e_s \subset \mathcal{G}$, can be seen as the coordination cell of the two robots on the paths $\gamma|_{e_r}$ and $\gamma|_{e_s}$ parametrized by unit speed. In particular, our coordination cell can be seen as $[0, l_r] \times [0, l_s]$, in which $l_k = l(e_k)$ is the length of e_k.

Within each coordination cell, we use the term *obstacle region* to refer to the set of points corresponding to positions in which the interiors of $\mathcal{R}_1$ and $\mathcal{R}_2$ intersect. The *free region* is set of points not in the obstacle region. In Figure 1, we see an example of a coordination cell and its obstacle region. Notice that our coordination cell is similar to the coordination diagram of [20], but since our robots are polygonal and our paths are piecewise-linear, the obstacle region in our coordination cell is a collection of polygonal connected components. If we confine our attention to a single coordination cell (as we will in Section III-B), a coordination is essentially a piecewise-smooth path from $(0, 0)$ to (l_r, l_s) inside its free region.

Equivalence and partial order: Different paths can have equal $\mathcal{L}^\infty$ lengths and consequently equal J costs. In general, equality of J cost defines an equivalence relation $\sim$ on the set of all coordinations $\mathcal{C}$. In fact, since our optimality criterion is based on the value of J, we can consider the set $\tilde{\mathcal{C}} = \mathcal{C}/\!\sim$ of equivalence classes and use term *coordination class* to refer to one of these maximal sets of equivalent coordinations.

Now we can define the partial order mentioned in Section II in more detail. Define a relation $\le$ on $\tilde{\mathcal{C}}$ as follows: For any two coordination classes $[C]$ and $[C']$, say that $[C] \le [C']$ if and only if $J_1(C) \le J_1(C')$ and

$J_2(C) \leq J_2(C')$. It is easy to see that the definition is independent of the choice of representative, so $\leq$ is well-defined. Any minimal element in this partial order is a Pareto-optimal coordination class. The algorithm proposed here computes a representative from each of these Pareto-optimal coordination classes.

To describe the algorithm, we first describe how to compute all Pareto-optimal coordinations in the simpler case of a single coordination cell, then extend the algorithm to the whole $\mathcal{G} \times \mathcal{G}$ which consists of a collection of such coordination cells.

B. Two fixed paths

In this section we describe how to compute all Pareto-optimal coordinations in a single coordination cell, i.e. for the two robots on two fixed paths. As it is stated in Section III-A, the obstacle (or collision) region of our coordination cell consists of a collection of polygons. Thus, we may use the terms *vertex* and *edge* of the obstacle region. To present the algorithm, we give some statements about the properties of Pareto-optimal coordinations.

Lemma 1: For every Pareto-optimal coordination class $[C_{\text{op}}]$ in a coordination cell $[0, l_r] \times [0, l_s]$ there is a representative $C_{\text{eq}} \in [C_{\text{op}}]$ such that C_{eq} is composed of a finite sequence of linear segments between the vertices of obstacle region, initial $(0,0)$ and goal (l_r, l_s) points, and in some cases a point on the boundary of the coordination cell, (t, l_s) or (l_r, t).

Proof: First, notice that there is an equivalent coordination to C_{op} which is piecewise-linear. By an argument similar to the one in [6], which is essentially based on shortening, we get the result. As a remark, notice that in cases where for example robot $\mathcal{R}_1$ reaches its goal sooner than R_2, the final segment of each coordination in $[C_{\text{op}}]$ lies over the boundary of coordination cell and in particular is of the form $(l_r, t) - (l_r, l_s)$. That is why in some cases C_{eq} passes through a point on the boundary which may neither be an obstacle vertex nor an endpoint. $\square$

As a consequence of Lemma 1, it is sufficient to consider only coordinations composed of a sequence of linear segments between the vertices of obstacle region, initial and goal points, and in some cases a point on the boundary of coordination cell. We call such Pareto-optimal coordinations *visibility Pareto-optimal*. The next lemma explains this naming and characterizes the set of vertices on the boundary.

Lemma 2: Suppose $[C_{\text{op}}]$ is a visibility Pareto-optimal coordination class with $C_{\text{op}} \in [C_{\text{op}}]$ of the form described in Lemma 1. Let (t_1, t_2) denote the last vertex of C_{op} which is not on the boundary (that is, that last vertex such that $t_1 \neq l_r$ and $t_2 \neq l_s$). There are three cases:

(i) If $J_1(C_{\text{op}}) < J_2(C_{\text{op}})$, then the line segment $(t_1, t_2) - (l_r, t_2 + l_r - t_1)$ is collision free and furthermore, is exactly a segment of C_{op}.

(ii) If $J_1(C_{\text{op}}) > J_2(C_{\text{op}})$, then the line segment $(t_1, t_2) - (t_1 + l_s - t_2, l_s)$ is collision free and furthermore, is exactly a segment of C_{op}.

(iii) If $J_1(C_{\text{op}}) = J_2(C_{\text{op}})$, then there is at most one such $[C_{\text{op}}]$ in $\tilde{\mathcal{C}}$ and it is represented by the shortest path on the visibility graph of obstacle vertices and endpoints.

```
SINGLECELLPARETOOPTIMALCOORD(e_r, e_s, R_1, R_2)
S ← ∅
P ← OBSTACLEPOLYGONS(e_1, e_2, R_1, R_2)
VG ← VISIBILITYGRAPH(P ∪ {(0,0), (l_r, l_s)})
DIJKSTRA(VG, (0,0), L^∞)
S ← S ∪ SHORTEST((l_r, l_s))

for each vertex v = (x_1, x_2) of each polygon in P
    ⟨⟨Is R_1 is nearer the goal than R_2?⟩⟩
    if x_1 − x_2 > l_r − l_s
        q ← (l_r, x_2 + l_r − x_1)
        if FREE(P, v, q) and FREE(P, q, (l_r, l_s))
            S ← S ∪ {(SHORTEST(v), q, (l_r, l_s))}

    ⟨⟨Is R_2 is nearer the goal than R_1?⟩⟩
    if x_1 − x_2 < l_r − l_s
        q ← (x_1 + l_s − x_2, l_s)
        if FREE(P, v, q) and FREE(P, q, (l_r, l_s))
            S ← S ∪ {(SHORTEST(v), q, (l_r, l_s))}

S ← PRUNESOLUTIONS(S)
return S
```

Fig. 2. The basic algorithm for two fixed paths.

Proof: In the first two cases, if the line segment is not collision free, we can always find another coordination which reduces both J_1 and J_2 contradicting the optimality of C_{op}. Furthermore, taking the line segment is the best strategy. In other words, if the line segment is not part of C_{op}, we can replace it in and find a better coordination. In the third case, it is obvious that $[C_{\text{op}}]$, if exists, is unique, because for any coordination $[C'] \neq [C_{\text{op}}]$ with $J_1(C') = J_2(C')$, either $[C'] < [C_{\text{op}}]$ or $[C'] > [C_{\text{op}}]$. In fact, C_{op} is the $\mathcal{L}^2$-shortest path from $(0,0)$ to (l_r, l_s) in the interior of coordination cell. In other words, C_{op} is the shortest path according to the Euclidean metric. As a remark, notice that an $\mathcal{L}^2$-shortest path is also $\mathcal{L}^\infty$-shortest but the converse need not be true. $\square$

Corollary 3: The number of Pareto-optimal coordinations is finite.

Note that case (i) of in Lemma 2, $(l_r, t_2 + l_r - t_1)$ is simply the intersection of the line $x_1 = l_r$ and the line with slope 1 through (t_1, t_2). Similar remarks can be made for cases (ii) and (iii). Intuitively, we can think of shooting a ray at slope 1 from each obstacle vertex (t_1, t_2) and stopping when that ray hits a point with either $x_1 = l_r$ or $x_2 = l_s$, corresponding respectively to $\mathcal{R}_1$ or $\mathcal{R}_2$ reaching its goal. Lemmas 1 and 2 tell us that every Pareto-optimal coordination class has a representative that ends with such a slope-1 segment.

Now we are ready to present the algorithm in Figure 2. The function OBSTACLEPOLYGONS computes the obstacle region polygons. As it is stated in Section III-A, the obstacle region is a collection of polygons which can be computed by collision detection algorithm along each pair of linear path segments. More precisely, we build the Minkowski sum of $\mathcal{R}_2$ on $\mathcal{R}_1$ which is a bigger polygon around $\mathcal{R}_1$ representing the position of the center of $\mathcal{R}_2$ while $\mathcal{R}_1$ and $\mathcal{R}_2$ touch each other. The intersection points of linear path segments with this polygon gives the boundary of obstacle region. The visibility graph of the vertices of obstacle region and endpoints is computed in VISIBILITYGRAPH according to the well-known radial sweep algorithm in [6]. The function FREE checks to see whether a line segment is contained in the free region of the coordination cell. This can be performed by simple geometric tests. The optimal path candidates described in Lemma 2 are then added to S. Lastly, we notice that some of the added paths may not be actually optimal. These are removed in PRUNESOLUTIONS by simple pairwise comparisons.

Theorem 4: The algorithm SINGLECELLPARETOOPTIMALCOORD in Figure 2 correctly computes all Pareto-optimal coordinations of the two robots on two fixed piecewise-linear paths.

Proof: The result directly follows from Lemma 1 and Lemma 2. □

If n denotes the number of obstacle vertices, then VISIBILITYGRAPH takes $O(n^2 \log n)$ time. Since each of the other steps can be done in $O(n^2)$ time, the time complexity of SINGLECELLPARETOOPTIMALCOORD is also $O(n^2 \log n)$.

C. Acyclic roadmap

In this section we extend the coordination cell algorithm in Figure 2 to the general case of two robots on an acyclic roadmap $\mathcal{G}$. The theory developed in [10] easily shows that if $\mathcal{G}$ is acyclic, $\mathcal{G} \times \mathcal{G}$ with L^2 metric is non-positively curved (NPC) and consequently it has unique Euclidean geodesics. For some applications of NPC spaces and Gromov's hyperbolic group theory, see [8], [9]. These results imply:

Proposition 5: Assume $\mathcal{G} \times \mathcal{G}$ is equipped with L^2 metric in which $\mathcal{G}$ is an acyclic graph. Note that $\mathcal{G}$ need not necessarily be connected. Between any two points $x, y \in \mathcal{G} \times \mathcal{G}$ there is exactly one geodesic connecting x and y if they are in the same connected component.

This nice property makes $\mathcal{G} \times \mathcal{G}$ similar to the plane, because geodesics in $\mathcal{G} \times \mathcal{G}$ play the role of lines in plane. In fact, geodesics inside a coordination cell coincide with the usual Euclidean lines. Thus, we have the following lemmas similar to the ones in Section III-B. From now on, we assume $\mathcal{G}$ is acyclic. Note that since in each coordination cell the obstacle region is polygonal, the obstacle region in $\mathcal{G} \times \mathcal{G}$ is also polygonal.

Lemma 6: For every Pareto-optimal coordination class $[C_{\rm op}]$ in $\mathcal{G}$ from (C_1^{init}, C_2^{init}) to (C_1^{goal}, C_2^{goal}) there is a representative $C_{\rm eq} \in [C_{\rm op}]$ such that $C_{\rm eq}$ is composed of a finite sequence of geodesic segments between the vertices of obstacle region, initial and goal points, and in some cases a point on the boundary, (x, C_2^{goal}) or (C_1^{goal}, x).

Proof: Very similar to the proof of Lemma 1. □

```
PARETOOPTIMALCOORD(M, R1, R2, C^init, C^goal)
S ← ∅ ⟨⟨S is the set of candidate solutions.⟩⟩
P ← ∅
for each pair of edges e_i, e_j ∈ G
    P ← P ∪ OBSTACLEPOLYGONS(e_i, e_j, R1, R2)
VG ← GENVISIBILITYGRAPH(P ∪ {C^init, C^goal})
DIJKSTRA(VG, C^init, L^∞)
S ← S ∪ SHORTEST(C^goal)

for each vertex v = (x1, x2) of each polygon in P
    ⟨⟨Is R1 is nearer to C^goal than R2?⟩⟩
    if d(x1, C1^goal) < d(x2, C2^goal)
        q ← (C1^goal, x2 + δx1)
        if FREE(P, v, q) and FREE(P, q, C^goal)
            S ← S ∪ {(SHORTEST(v), q, C^goal))}

    ⟨⟨Is R2 is nearer to C^goal than R1?⟩⟩
    if d(x1, C1^goal) > d(x2, C2^goal)
        q ← (x1 + δx2, C2^goal)
        if FREE(P, v, q) and FREE(P, q, C^goal)
            S ← S ∪ {(SHORTEST(v), q, C^goal))}

S ← PRUNESOLUTIONS(S)
return S
```

Fig. 3. The algorithm for finding all Pareto-optimal coordinations of two robots on an acyclic piecewise-linear roadmap.

Lemma 7: Assume $[C_{\rm op}]$ is a Pareto-optimal coordination class. and $C_{\rm op}$ is of the form described in Lemma 6. Once again, there are three cases:

(i) If $J_1(C_{\rm op}) < J_2(C_{\rm op})$, then the geodesic segment A to (C_1^{goal}, y) with equal progression for $\mathcal{R}_1$ and $\mathcal{R}_2$ is collision free and furthermore, is exactly a segment of $C_{\rm op}$.

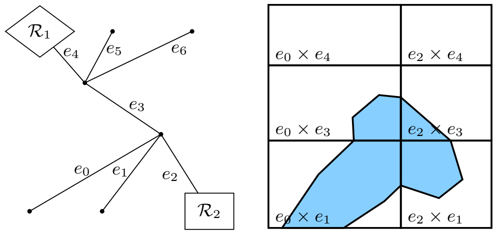
(ii) If $J_1(C_{\rm op}) > J_2(C_{\rm op})$, then the geodesic segment A to (y, C_2^{goal}) with equal progression for $\mathcal{R}_1$ and $\mathcal{R}_2$ is collision free and furthermore, is exactly a segment of $C_{\rm op}$.

(iii) If $J_1(C_{\rm op}) = J_2(C_{\rm op})$, then there is at most one such $[C_{\rm op}]$ in $\tilde{\mathcal{C}}$ and it is represented by the shortest path on the generalized visibility graph of obstacle vertices and endpoints.

Above, $A = (x_1, x_2)$ is the last vertex of $C_{\rm op}$ which is not on the boundary, i.e. $x_1 \neq C_1^{goal}$ and $x_2 \neq C_2^{goal}$.

Proof: Very similar to the proof of Lemma 2. □

In PARETOOPTIMALCOORD in Figure 3, GENVISIBILITYGRAPH is a generalization of visibility graph algorithm in [6]. More precisely, we do a radial sweeping algorithm. This can be done because the radial geodesics are unique. To sweep about vertex v, we just sort all the obstacle vertices throughout the cell complex in their geodesic angle order. We extend the standard algorithm by maintaining a separate balanced binary tree for each 2-cell in $\mathcal{G} \times \mathcal{G}$ intersected by the sweep ray. Edges in



(a) (b)

Fig. 4. (a) A coordination problem on a roadmap with 7 edges. (b) A subset of $\mathcal{G} \times \mathcal{G}$ for this problem.

each tree remain ordered according to their distance from v. To check whether a geodesic is collision free, we check collision for all the nearest edges given by our tree data structure in those cells that are traversed by the geodesic. The remainder of the algorithm is essentially unchanged from SINGLECELLPARETOOPTIMALCOORD.

Theorem 8: The algorithm PARETOOPTIMALCOORD in Figure 3 correctly computes all Pareto-optimal coordinations of the two robots on $\mathcal{M}$ from C^{init} to C^{goal}.

Proof: The result directly follows from Lemma 6 and Lemma 7. □

Complexity: Let m denote the number of edges in $\mathcal{M}$ and let n denote total number of obstacle vertices. Since each geodesic passes through at most $2m$ cells, in computing the visibility graph, we perform $O(mn^2)$ balanced binary tree operations, each taking $O(\log n)$ time. The visibility graph therefore requires $O(mn^2 \log n)$ time to compute. Both Dijkstra's algorithm and the pruning of S take $O(n^2)$ time. Finally, notice that the number of Pareto-optimal coordinations is less than or equal to the total number of obstacle vertices plus two. Thus, the complexity of algorithm output is $O(n)$. Hence, the total complexity of our algorithm is $O(mn^2 \log n)$.

IV. EXPERIMENTAL RESULTS

We have implemented a simplified version of the described algorithm using naïve data structures and algorithms in several places. An implementation more faithful to the description in Section III-C can be expected to perform better than the present implementation. The run times below are for C++ compiled under Linux and executed on 2.5GHz processor.

Figure 4 shows an example coordination problem on a connected roadmap with 7 edges. Each robot is shown in its initial state and the goal is for the robots to switch places with one another. For this problem $\mathcal{G} \times \mathcal{G}$ contains 31 obstacle polygons totalling 174 obstacle vertices. The complete set of 4 Pareto-optimal coordinations illustrated in Figure 5 took appoximately 0.2 seconds to compute.

As a second example, consider the star graph S_n with vertex set $\{v_0, \ldots, v_{n-1}\}$ and edge set $\{(v_0, v_i) \ : \ 1 \leq i < n\}$. Coordination on this family of graphs is unusual because because every cell of $\mathcal{G} \times \mathcal{G}$ has a non-empty obstacle region. In Figure 6, $\mathcal{R}_1$ and $\mathcal{R}_2$ navigate on an embedding of S_{16}. The obstacle region has $15^2 = 225$ obstacles with 933 vertices in total. The two Pareto-optimal

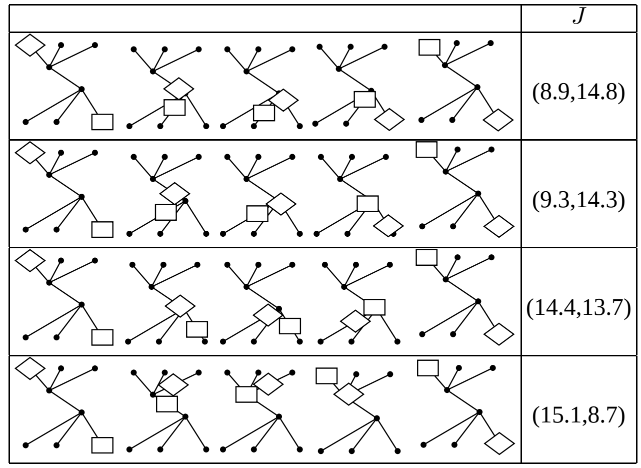
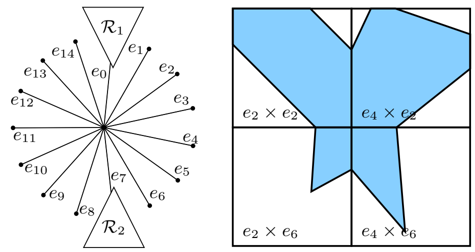


Fig. 5. The four Pareto-optimal solutions for the problem in Figure 4.



(a) (b)

Fig. 6. (a) A coordination problem on the star graph S_{16}. (b) A subset of $\mathcal{G} \times \mathcal{G}$ for this problem.

solutions are shown in Figure 7. Our implementation took 25 seconds to solve this problem.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented an algorithm to compute all Pareto-optimal coordinations of two polygonal translating robots, which have a maximum speed and are capable of instantly switching between any two speeds bounded by the maximum speed, on an acyclic roadmap of piecewise-linear paths in the plane. We showed that the algorithm works correctly and showed that its complexity is $O(mn^2 \log n)$, in which m is the number of edges of roadmap and n is the total number of obstacle vertices.

However, notice that instead of assuming the robots are translating polygons on a piecewise-linear roadmap, we may assume that the configuration space of each robot while moving on the roadmap is $\mathcal{G}$, the underlying acyclic graph of the roadmap, and the obstacle regions in $\mathcal{G} \times \mathcal{G}$ are polygonal. In that case, exactly the same algorithm can be applied to find all Pareto-optimal coordinations.

More generally, even in cases where the obstacle regions are not polygonal but we can compute bitangents and consequently the generalized visibility graph, we may trivially modify the algorithm presented in this paper to compute all Pareto-optimal coordinations of such robots. In this regard, for example in case of car-like mobile robots on a network of $\mathcal{SA}$ paths (see [20]), we may think of computing bitangents of the obstacle region in $\mathcal{G} \times \mathcal{G}$ to compute the generalized visibility graph. We can then

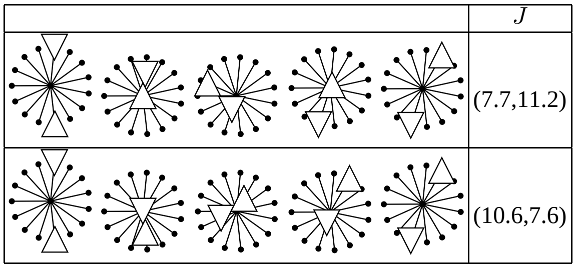


Fig. 7. The two Pareto-optimal solutions for the problem in Figure 6.

find all Pareto-optimal coordinations.

As future work, we may think of solving the problem for n robots on a roadmap. In that case, we have to find Pareto-optimal collision free coordinations in the n-dimensional cube complex $\mathcal{G}^n = \mathcal{G} \times \mathcal{G} \times \cdots \times \mathcal{G}$. Notice that since the collision of any two robots is considered a failure of the whole configuration, the obstacle regions in each cell of $\mathcal{G}^n$ are cylindrical. This property may be exploited to solve the problem.

ACKNOWLEDGMENT

This work was funded in part by NSF IIS Award 9875304 and ONR Grant N00014-02-1-0488.

VI. REFERENCES

[1] S. Akella and S. Hutchinson. Coordinating the motions of multiple robots with specified trajectories. In *IEEE Int. Conf. Robot. & Autom.*, pages 624–631, 2002.

[2] M. D. Ardema and J. M. Skowronski. Dynamic game applied to coordination control of two arm robotic system. In R. P. Hämäläinen and H. K. Ehtamo, editors, *Differential Games - Developments in Modelling and Computation*, pages 118–130. Springer-Verlag, Berlin, 1991.

[3] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Int. J. Robot. Res.*, 10(6):628–649, December 1991.

[4] C. Chang, M. J. Chung, and B. H. Lee. Collision avoidance of two robot manipulators by minimum delay time. *IEEE Trans. Syst., Man, Cybern.*, 24(3):517–522, 1994.

[5] G. Chirikjian, A. Pamecha, and I. Ebert-Uphoff. Evaluating efficiency of self-reconfiguration in a class of modular robots. *Journal of Robotic Systems*, 13(5):717–338, 1996.

[6] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 1997.

[7] M. Erdmann and T. Lozano-Perez. On multiple moving objects. In *IEEE Int'l. Conf. Robot. & Autom.*, pages 1419–1424, 1986.

[8] R. Ghrist. Configuration spaces and braid groups on graphs in robotics. In *Braids, Links, and Mapping Class Groups: the Proceedings of Joan Birman's 70th Birthday*, volume 24, pages 29–40. AMS/IP Studies in Mathematics, 2001.

[9] R. Ghrist. Shape complexes for metamorphic robot systems. In *Proc. Workshop on the Algorithmic Foundations of Robotics*, December 2002.

[10] M. Gromov. Hyperbolic groups. In S.M. Gersten, editor, *Essays in Group Theory*, pages 75–264. Springer-Verlag, 1987.

[11] H. Hu, M. Brady, and P. Probert. Coping with uncertainty in control and planning for a mobile robot. In *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 1025–1030, Osaka, Japan, November 1991.

[12] K. Kotay, D. Rus, M. Vora, and C. McGray. The self-reconfiguring robotic molecule: Design and control algorithms. In P.K. Agarwal, L. Kavraki, and M. Mason, editors, *Robotics: The Algorithmic Perspective*. AK Peters, Natick, MA, 1998.

[13] S. M. LaValle and S. A. Hutchinson. Path selection and coordination of multiple robots via Nash equilibria. In *Proc. 1994 IEEE Int'l Conf. Robot. & and Autom.*, pages 1847–1852, May 1994.

[14] S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Trans. on Robotics and Automation*, 14(6):912–925, December 1998.

[15] J. Miura and Y. Shirai. Planning of vision and motion for a mobile robot using a probabilistic model of uncertainty. In *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, pages 403–408, Osaka, Japan, May 1991.

[16] P. A. O'Donnell and T. Lozano-Pérez. Deadlock-free and collision-free coordination of two robot manipulators. In *IEEE Int'l. Conf. Robot. & Autom.*, pages 484–489, 1989.

[17] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. In *Proc. Fifth International Workshop on the Algorithmic Foundations of Robotics*, 2002.

[18] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, New York, NY, 1985.

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

[20] T. Simeon, S. Leroy, and J.-P. Laumond. Path coordination for multiple mobile robots: a resolution complete algorithm. *IEEE Trans. Robot. & Autom.*, 18(1), February 2002.

[21] S.-H. Suh and K. G. Shin. A variational dynamic programming approach to robot-path planning with a distance-safety criterion. *IEEE Trans. Robot. & Autom.*, 4(3):334–349, June 1988.

[22] P. Svestka and M. H. Overmars. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *IEEE Int'l. Conf. Robot. & Autom.*, pages 1631–1636, 1995.

[23] M. Yim. *Locomotion with a Unit-Modular Reconfigurable Robot*. PhD thesis, Stanford Univ., December 1994. Stanford Technical Report STAN-CS-94-1536.