Exact Pareto-Optimal Coordination of Two Translating Polygonal Robots on a Cyclic Roadmap

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

Abstract

We consider planning optimal collision-free motions of two polygonal robots under translation. Each robot has a reference point that must lie on a given graph, called a roadmap, which is embedded in the plane. The initial and the goal are given for each robot. Rather than impose an a priori cost scalarization for choosing the best combined motion, we consider finding motions whose cost vectors are Pareto-optimal. Pareto-optimal coordination strategies are the ones for which there exists no strategy that would be better for both robots. Our problem translates into shortest path problems in the coordination space which is the Cartesian product of the roadmap, as a cell complex, with itself. Our algorithm computes an upper bound on the cost of each motion in any Pareto-optimal coordination. Therefore, only a finite number of homotopy classes of paths in the coordination space need to be considered. Our algorithm computes all Pareto-optimal coordinations in time $O(2^{5\alpha}m^{1+5\alpha}n^2\log(m^{2\alpha}n))$, in which m is the number of edges in the roadmap, n is the number of coordination space obstacle vertices, and $\alpha = 1 + \lceil(5\ell + r)/b\rceil$ where ℓ is total length of the roadmap and r is total length of coordination space obstacle boundary and b is the length of the shortest edge in the roadmap.

1 Introduction

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. A decoupled approach typically generates paths for each robot independently, and then considers the interactions between the robots. In [10, 16], 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 Automated Guided Vehicles (AGV), the roadmap might represent all allowable mobility for each robot.

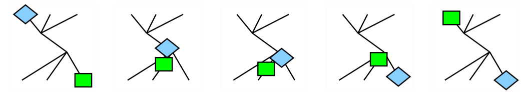
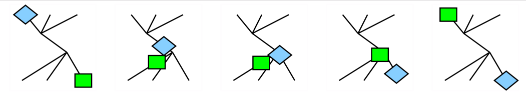
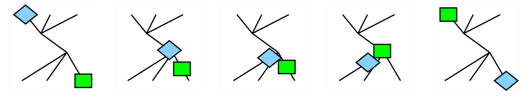
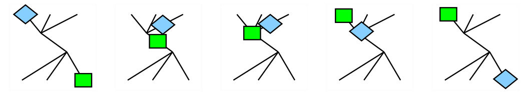
	Cost
	(8.9,14.8)
	(9.3,14.3)
	(14.4,13.7)
	(15.1,8.7)

Figure 1: The four Pareto-optimal solutions for a coordination problem in which the robots want to exchange place.

In this paper, we study the problem of planning optimal motions of two polygonal robots traveling on a given roadmap. The robots must be disjoint when they travel, and as a result, there are tradeoffs between the robots' completion times. One approach is to consider a scalar cost that combines the completion times. Minimizing the average time robots take to reach their goals [9, 12], and minimizing the time that the last robot takes have been studied before [15]. The problem with scalarization is that it eliminates many interesting coordination strategies, possibly even neglecting optimality for some robots [10]. Rather than impose an a priori scalarization for choosing the best combined motion, we consider finding motions whose cost vectors (cost of robot 1, cost of robot 2) are Pareto-optimal. Pareto-optimal coordination strategies are the ones for which there exists no strategy that would be better for both robots; see [14]. Optimal coordinations according to a scalar cost impose a predetermined preference between the robots, whereas having all Pareto-optimal coordinations beforehand gives the freedom to determine the preference at run-time. It was shown that the number of Pareto-optimal coordinations for n robots on any roadmap is

[*] Department of Computer Science, University of Illinois at Urbana-Champaign, {chitsaz,lavalle}@cs.uiuc.edu

[†] Department of Computer Science and Engineering, University of South Carolina, jokane@cse.sc.edu

finite [6]; therefore, it is plausible to seek all of them. A sample problem and its Pareto-optimal solutions are illustrated in Figure 1.

This work is inspired by previous approaches to multiple robot coordination. O'Donnell and Lozano-Pérez introduced coordination diagrams for planning motions of two robot manipulators [13]. Alt and Godau used similar coordination spaces in a different context to compute the Fréchet distance between two polygonal curves [1]. LaValle and Hutchinson gave the first approach to Pareto-optimal coordination of multiple robots [10]. They presented an approximation algorithm based on dynamic programming in the discretized coordination space. Ghrist et al. gave a characterization of Pareto-optimal coordinations of multiple robots using CAT(0) geometry [7]. They provided an algorithm to shorten a given coordination to a homotopic, possibly Pareto-optimal one. In our previous work, we gave an efficient algorithm for finding Pareto-optimal coordination strategies for two polygonal robots on an acyclic roadmap [4]. In this paper, we present an algorithm for the general case. Due to space limitations, proofs of the propositions, lemmas, and theorems are included in the appendix.

2 Problem Formulation

We give a brief formulation of the problem. For a more detailed exposition, see [4]. Let the robots, $\mathcal{R}_1$ and $\mathcal{R}_2$, be polygonal open sets embedded in the plane. They translate along a roadmap $\mathcal{G}$, which is an embedded graph in the plane[1]. Edges of $\mathcal{G}$ are piecewise-linear segments. The roadmap need not be connected, so effectively each robot can have its own roadmap. Each edge of $\mathcal{G}$ is weighted by its Euclidean length. In this way, $\mathcal{G}$ turns into a metric graph [3]. The robots have a maximum speed and are capable of instantly switching to any speed between zero and the maximum. By scaling the respective metric graphs, we assume without loss of generality that both robots have unit maximum speed. 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 $\mathcal{G}$.

We are given an initial and a goal configuration $q_i^{init}, q_i^{goal} \in \mathcal{G}$ for each robot $\mathcal{R}_i$. The obstacle region, denoted by $\mathcal{O} \subset \mathcal{G} \times \mathcal{G}$, is the set of configurations at which $\mathcal{R}_1$ and $\mathcal{R}_2$ collide. Since the robots are polygonal and roadmap paths are piecewise-linear, the obstacle region is a collection of polygonal, open connected components. A coordination is a continuous path in the coordination space $\mathcal{G} \times \mathcal{G}$, from $q^{init} = (q_1^{init}, q_2^{init})$ to $q^{goal} = (q_1^{goal}, q_2^{goal})$, that avoids $\mathcal{O}$.

The vector-valued cost $\mathcal{J} = (J_1, J_2)$ separately measures the time that each robot takes to reach its goal and stop. Define $d_\infty : ((x_1, x_2), (y_1, y_2)) \mapsto \max(d(x_1, y_1), d(x_2, y_2))$, in which d is the metric in $\mathcal{G}$. Let $\mathcal{L}^\infty$ be the functional that gives the length of each continuous path in $\mathcal{G} \times \mathcal{G}$ according to d_∞. For each coordination $\gamma = (\gamma_1, \gamma_2) : [0, 1] \to \mathcal{G} \times \mathcal{G}$, let $t_i = \min\{t \in [0, 1] : \gamma_i([t, 1]) = q_i^{goal}\}$. In that case, $J_i(\gamma) = \mathcal{L}^\infty(\gamma|_{[0,t_i]})$ and $\mathcal{J}(\gamma) = (J_1(\gamma), J_2(\gamma))$. Let C be the set of all coordinations. The cost $\mathcal{J} : \mathrm{C} \to [0, \infty)^2$ induces a partial order on the set of all coordinations C. Each minimal element in this partial order is called a Pareto-optimal coordination. The problem is to find all Pareto-optimal coordinations for the two robots.

3 Canonical Pareto-optimal Coordinations

Different paths that have the same end points can have equal $\mathcal{L}^\infty$ lengths in the coordination space. Consequently, there are different coordinations with equal cost. We fix a canonical form for equivalent Pareto-optimal coordinations based on Euclidean shortest paths.

Proposition 1 *For every Pareto-optimal coordination, there is an equivalent coordination that is composed of a finite sequence of Euclidean shortest segments between the vertices of the obstacle region, q^{init}, q^{goal}, and in some cases (x, q_2^{goal}) or (q_1^{goal}, x).*

The points (q_1^{goal}, x) and (x, q_2^{goal}) that need to be considered are characterized in [4]. A point (q_1^{goal}, x) or (x, q_2^{goal}) needs to be considered if there is a collision-free Euclidean shortest segment, with equal progression for $\mathcal{R}_1$ and $\mathcal{R}_2$, from an obstacle vertex or q^{init} to the point (q_1^{goal}, x) or (x, q_2^{goal}).

4 Algorithm Presentation

To find canonical Pareto-optimal coordinations, our algorithm computes Euclidean shortest segments between obstacle vertices, initial and goal configurations, and some points (q_1^{goal}, x) and (x, q_2^{goal}) in the coordination space. Fixing the end points in the coordination space, there is only one shortest path in every homotopy class, which holds because the space is non-positively curved [6]. The roadmap can be cyclic, and consequently the universal cover of the coordination space can be unbounded. An incremental exploration of the unbounded universal cover may never stop, because there are multiple Pareto-optimal coordinations whose maximum length is unknown beforehand. Our algorithm constructs a bounded portion of the universal cover in which the shortest path algorithm is applied. Using

[1] If we assume that $\mathcal{G}$ is locally embedded in the plane, in which case its edges may intersect, then our algorithm correctly works and our results still hold. For the sake of clarity, we preferred to assume $\mathcal{G}$ is embedded.

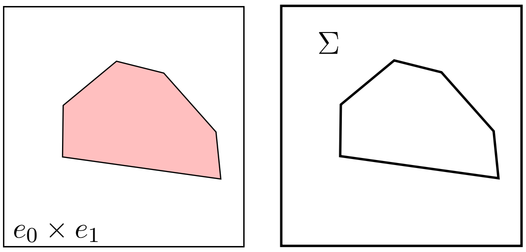


Figure 2: A coordination cell $e_0 \times e_1$, and its skeleton Σ.

shortest path algorithms in the plane such as continuous Dijkstra [8, 11] or visibility graph methods in the universal cover of the coordination space, one can compute the shortest paths. Using a cost upper bound computed in advance, our algorithm constructs the relevant part of the universal cover. The rest of the algorithm is essentially identical to the acyclic case applied to the universal cover [4].

4.1 Coordination Cost Upper Bound

In a scalar minimization problem, the cost of any feasible solution is an upper bound for the cost of an optimal solution. The key idea here is the same. The following lemma derives an upper bound on the cost of every motion in any Pareto-optimal coordination.

Lemma 2 *Let $\Delta_1, \Delta_2 \subseteq \mathcal{G}$ be such that $\{q_1^{goal}\} \times \Delta_2 = \{q_1^{goal}\} \times \mathcal{G} - \mathcal{O}$, and $\Delta_1 \times \{q_2^{goal}\} = \mathcal{G} \times \{q_2^{goal}\} - \mathcal{O}$. Let δ_i be the diameter of Δ_i as a metric graph. Let λ be the Euclidean length of an arbitrary coordination γ. Let τ be a Pareto-optimal coordination. In that case, $J_1(\tau), J_2(\tau) \leq \lambda + \delta$, in which $\delta = \max(\delta_1, \delta_2)$.*

To compute λ, which is the Euclidean length of an arbitrary coordination γ, we use the dimension reduction method of Aronov et al. [2]. Denote the boundary of obstacle region by $\partial\mathcal{O}$. Define $\Upsilon_1 = \{q_1^{init}\} \times \mathcal{G} - \mathcal{O}, \Upsilon_2 = \mathcal{G} \times \{q_2^{init}\} - \mathcal{O}, \Upsilon_3 = \{q_1^{goal}\} \times \mathcal{G} - \mathcal{O}, \Upsilon_4 = \mathcal{G} \times \{q_2^{goal}\} - \mathcal{O}$, and $\Sigma = \partial\mathcal{O} \cup (\bigcup_{j=1}^{4} \Upsilon_j)$. We call Σ the *skeleton* of $\mathcal{G} \times \mathcal{G} - \mathcal{O}$. See Figure 2 for a simple example. Note that the skeleton is a one-dimensional object. It is composed of five pieces: $\mathcal{R}_1$ at its initial, $\mathcal{R}_2$ at its initial, $\mathcal{R}_1$ at its goal, $\mathcal{R}_2$ at its goal, and $\mathcal{R}_1$ touching $\mathcal{R}_2$. The following lemma follows from Lemma 1 in [2].

Lemma 3 (Aronov et al. [2]) *There is a collision-free path from q^{init} to q^{goal} in the coordination space if and only if there is a path from q^{init} to q^{goal} in Σ, the skeleton of $\mathcal{G} \times \mathcal{G} - \mathcal{O}$.*

Our algorithm constructs Σ by gluing $\partial\mathcal{O}$ and Υ_j along their intersection points. We discussed how to compute the obstacle region in [4]. To compute Υ_j, first we compute $\mathcal{M} = \mathcal{R}_1 \ominus \mathcal{R}_2$, the Minkowski difference. By intersecting polygon $\mathcal{M}$ positioned respectively at q_1^{init} and q_1^{goal} with $\mathcal{G}$, we compute $\Gamma_2 = \mathcal{G} - (\{q_1^{init}\} \oplus \mathcal{M})$ and $\Delta_2 = \mathcal{G} - (\{q_1^{goal}\} \oplus \mathcal{M})$. By intersecting $-\mathcal{M}$ positioned respectively at q_2^{init} and q_2^{goal} with $\mathcal{G}$, we compute $\Gamma_1 = \mathcal{G} - (\{q_2^{init}\} \ominus \mathcal{M})$ and $\Delta_1 = \mathcal{G} - (\{q_2^{goal}\} \ominus \mathcal{M})$. It is enough to observe that $\Upsilon_1 = \{q_1^{init}\} \times \Gamma_2$, $\Upsilon_2 = \Gamma_1 \times \{q_2^{init}\}$, $\Upsilon_3 = \{q_1^{goal}\} \times \Delta_2$, and $\Upsilon_4 = \Delta_1 \times \{q_2^{goal}\}$. Dijkstra's algorithm yields γ and the minimum distance of q^{goal} from q^{init} in Σ which is taken as λ. Finally, the diameter, or an overestimate of the diameter, of Δ_i yields δ_i. Recall that the upper bound is $\lambda + \max(\delta_1, \delta_2)$.

4.2 Universal Cover of $\mathcal{G} \times \mathcal{G}$

Given the upper bound computed in Section 4.1, we only need to consider a finite portion of the universal cover. Here we describe an algorithm to construct it. Let $\mathcal{X}$ be the universal cover of $\mathcal{G}$ as a cell complex. In that case, $\mathcal{X} \times \mathcal{X}$ is the universal cover of $\mathcal{G} \times \mathcal{G}$, and it is enough to build the relevant part of $\mathcal{X}$ to construct the relevant part of $\mathcal{X} \times \mathcal{X}$.

Since $\mathcal{X}$ is composed of disjoint copies of a fundamental domain glued along identified vertices, we describe how to build a fundamental domain, denoted by $\mathcal{X}_0$. Let $\mathcal{T}$ be any spanning tree of $\mathcal{G}$ (a collection of trees if $\mathcal{G}$ is not connected). Let $e_i = (u_i, v_i), i = 1, \ldots, k$ be those edges of $\mathcal{G}$ that are not in $\mathcal{T}$. Obtain $\mathcal{X}_0$, the fundamental domain of $\mathcal{X}$, by adding k new vertices u_i^* and k edges (v_i, u_i^*) to $\mathcal{T}$. Note that the length of (v_i, u_i^*) is the same as that of (u_i, v_i). Cycles of $\mathcal{G}$ are opened into paths in $\mathcal{X}_0$. Vertices u_i^* must be identified with u_i in neighboring copies of the fundamental domain. We call u_i and u_i^* *gluing spots* of $\mathcal{X}_0$, because $\mathcal{X}$ is obtained by iteratively gluing disjoint copies of the fundamental domain to $\mathcal{X}_0$ such that $u_i \in \mathcal{X}_0$ is identified with u_i^* in one copy and $u_i^* \in \mathcal{X}_0$ is identified with u_i in another copy. If you want to see an illustration, see Figure 3 in the appendix.

Our algorithm builds $\mathcal{X}_0$ first, and initializes $\mathcal{Y} = \mathcal{X}_0$. It inserts u_i and u_i^* onto a list. For every vertex in the list, the algorithm generates a copy of $\mathcal{X}_0$ and glues it to $\mathcal{Y}$ along the relevant vertex. It then inserts the gluing spots of the newly generated copy in the list. It iterates over these steps until $\mathcal{Y}$ covers the relevant part of $\mathcal{X}$. For that purpose, the distance between the vertex and the initial copy of $\mathcal{X}_0$ is computed at each iteration. If that distance is more than the upper bound, then the vertex is neglected and no copies of $\mathcal{X}_0$ is glued. Eventually, the algorithm stops when there are no more vertices in the list.

4.3 Applying the Acyclic Algorithm

We showed how to compute $\mathcal{Y}$, the relevant portion of the universal cover of $\mathcal{G}$, in Section 4.2. Note that $\mathcal{Y} \subset \mathcal{X}$ is contractible. Therefore, it is acyclic and we may now apply our acyclic Pareto-optimal coordination algorithm to it [4]. The acyclic algorithm computes the visibility graph in $\mathcal{Y} \times \mathcal{Y}$ among obstacle vertices and the initial and goal configurations, augments it with some extra edges, and finds the shortest paths. Obstacles are computed once in $\mathcal{G} \times \mathcal{G}$, and they are copied multiple times to obtain obstacles in $\mathcal{Y} \times \mathcal{Y}$. There are several copies of q^{goal} in $\mathcal{Y} \times \mathcal{Y}$ all of which need to be considered in the visibility graph. Any collision-free path from q^{init} to any q^{goal} copy is a coordination. Consequently, there are several copies of visibility graph points (x, q_2^{goal}) and (q_1^{goal}, x) that need to be considered.

4.4 Complexity Analysis

Let m denote the number of edges in $\mathcal{G}$ and let n denote total number of obstacle vertices in $\mathcal{G} \times \mathcal{G}$. Let ℓ be the total length of $\mathcal{G}$ and r the total length of obstacle boundary. Let b denote the length of the shortest edge in $\mathcal{G}$. Define $\alpha = 1 + \lceil (5\ell + r)/b \rceil$.

Theorem 4 *The time complexity of our algorithm is* $O(2^{5\alpha} m^{1+5\alpha} n^2 \log(m^{2\alpha} n))$.

5 Conclusion

We presented an algorithm to compute all Pareto-optimal coordinations of two polygonal robots on a network of piecewise-linear paths in the plane. The key insight was an upper bound on the cost of each motion in a Pareto-optimal coordination. Our algorithm applies the previous acyclic algorithm to a finite portion of the universal cover of the coordination space [4]. This method can be applied to find all Pareto-optimal coordinations, provided the configuration space of each robot is $\mathcal{G}$, all paths in $\mathcal{G} \times \mathcal{G}$ are allowed, and the obstacle regions in $\mathcal{G} \times \mathcal{G}$ are polygonal.

Acknowledgment

We thank Jeff Erickson for his helpful comments. H. Chitsaz and S.M. LaValle are partially supported on NSF Award 0528086 (MSPA-MCS).

References

[1] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.*, 5:75–91, 1995.

[2] B. Aronov, M. de Berg, A. van der Stappen, P. Švestka, and J. Vleugels. Motion planning for multiple robots. *Discrete and Computational Geometry*, 22(4):505–525, 1999.

[3] M. Bridson and A. Haefliger. *Metric Spaces of Nonpositive Curvature.* Springer-Verlag, Berlin, 1999.

[4] H. Chitsaz, J. M. O'Kane, and S. M. LaValle. Exact pareto-optimal coordination of two translating polygonal robots on an acyclic roadmap. In *Proc. IEEE International Conference on Robotics and Automation*, pages 3981 – 3986, New Orleans, LA, Apr. 2004.

[5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications.* Springer, Berlin, 2000.

[6] R. Ghrist and S. M. LaValle. Nonpositive curvature and pareto-optimal coordination of robots. *SIAM J. Control & Optimization*, to appear 2006.

[7] R. Ghrist, J. M. O'Kane, and S. M. LaValle. Computing pareto optimal coordinations on roadmaps. *International Journal of Robotics Research*, 24(11):997–1010, Nov. 2005.

[8] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Computing*, 28:2215–2256, 1999.

[9] 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, Nov. 1991.

[10] 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, Dec. 1998.

[11] J. S. B. Mitchell. Shortest paths among obstacles in the plane. *Int. J. Comput. Geom. & Appl.*, 6(3):309–332, 1996.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

Appendix

Proposition 1

Proof sketch: We first choose Euclidean shortest paths as canonical form for $\mathcal{L}^\infty$-shortest paths in $\mathcal{G} \times \mathcal{G} - \mathcal{O}$. Note that a Euclidean shortest path is also $\mathcal{L}^\infty$-shortest. An argument similar to the one in [5], which is essentially based on shortening, shows Euclidean shortest paths in $\mathcal{G} \times \mathcal{G} - \mathcal{O}$ comprise Euclidean shortest segments between the vertices of the obstacle region $\mathcal{O}$ and the two end points.

We now choose a canonical form for a Pareto-optimal coordination γ. If robot $\mathcal{R}_1$ reaches its goal first under γ, then the final segment of γ is (q_1^{goal}, x) to (q_1^{goal}, q_2^{goal}) for some $x \in \mathcal{G}$. In that case, let $\tilde{\gamma}$ be that part of γ that goes from q^{init} to (q_1^{goal}, x). Likewise, the final segment of γ is (x, q_2^{goal}) to (q_1^{goal}, q_2^{goal}) if robot $\mathcal{R}_2$ reaches its goal first. In that case, let $\tilde{\gamma}$ be that part of γ that goes from q^{init} to (x, q_2^{goal}). If both robots simultaneously reach their goals, then let $\tilde{\gamma} = \gamma$. It is obvious that $\tilde{\gamma}$ is an $\mathcal{L}^\infty$-shortest path; otherwise, γ cannot be Pareto-optimal. Given the canonical form for $\mathcal{L}^\infty$-shortest paths, there is a path equivalent to $\tilde{\gamma}$ that is composed of a finite sequence of Euclidean shortest segments between the vertices of the obstacle region, q^{init}, and the final point of $\tilde{\gamma}$. Eventually, that part of γ that is not in $\tilde{\gamma}$ can be made a Euclidean shortest path. □

Lemma 2

Proof sketch: We claim that either $J_1(\tau) \leq \lambda$ or $J_2(\tau) \leq \lambda$. Suppose on the contrary, $J_1(\tau) > \lambda \geq J_1(\gamma)$ and $J_2(\tau) > \lambda \geq J_2(\gamma)$. In that case, γ is a better coordination than τ. That is contradictory to Pareto-optimality of τ. Suppose that $J_1(\tau) \leq \lambda$ and $\mathcal{R}_1$ reaches its goal first. Once $\mathcal{R}_1$ stops at its goal, robot $\mathcal{R}_2$ needs to travel along a roadmap path whose length is at most the diameter of the free portion of the roadmap. The free portion of the roadmap is Δ_2. Hence, $J_2(\tau) \leq J_1(\tau) + \delta_2 \leq \lambda + \delta_2$ because the travel time for both $\mathcal{R}_1$ and $\mathcal{R}_2$ is $J_1(\tau)$ up to the moment $\mathcal{R}_1$ stops, and is at most δ_2 for $\mathcal{R}_2$ afterwards. Thus, $J_1(\tau), J_2(\tau) \leq \lambda + \delta_2$.

Similarly, $J_1(\tau), J_2(\tau) \leq \lambda + \delta_1$ if $J_2(\tau) \leq \lambda$ and $\mathcal{R}_2$ reaches its goal first. Therefore, $J_1(\tau), J_2(\tau) \leq \lambda + \delta$, in which $\delta = \max(\delta_1, \delta_2)$. □

Theorem 4

Proof sketch: We claim that the upper bound in Section 4.1 is not more than $5\ell + r$. Total length of Σ is at most $4\ell + r$. Since λ is the length of a path in Σ, $\lambda \leq 4\ell + r$. Also, δ_i are not more than the total length of the roadmap, so $\delta \leq \ell$. Therefore, $\lambda + \delta \leq 5\ell + r$.

Every copy of the fundamental domain contributes at least b to the distance of a gluing spot from the initial copy of $\mathcal{X}_0$. Since $\mathcal{X}_0$ has no more than $2m$ gluing spots, at most $(2m)^\alpha$ copies of $\mathcal{X}_0$ are used in the construction of $\mathcal{Y}$. Therefore, $\mathcal{Y}$ has at most $2^\alpha m^{1+\alpha}$ edges. The number of obstacle vertices in $\mathcal{Y} \times \mathcal{Y}$ is at most $(2m)^{2\alpha} n$.

The complexity of our previous acyclic algorithm is $O(mn^2 \log n)$ [4]. Since the last step in this algorithm is the acyclic algorithm applied to $\mathcal{Y}$, which has at most $2^\alpha m^{1+\alpha}$ edges and $(2m)^{2\alpha} n$ obstacle vertices, the last step, which is the dominating step, takes $O(2^{5\alpha} m^{1+5\alpha} n^2 \log(m^{2\alpha} n))$ time. □

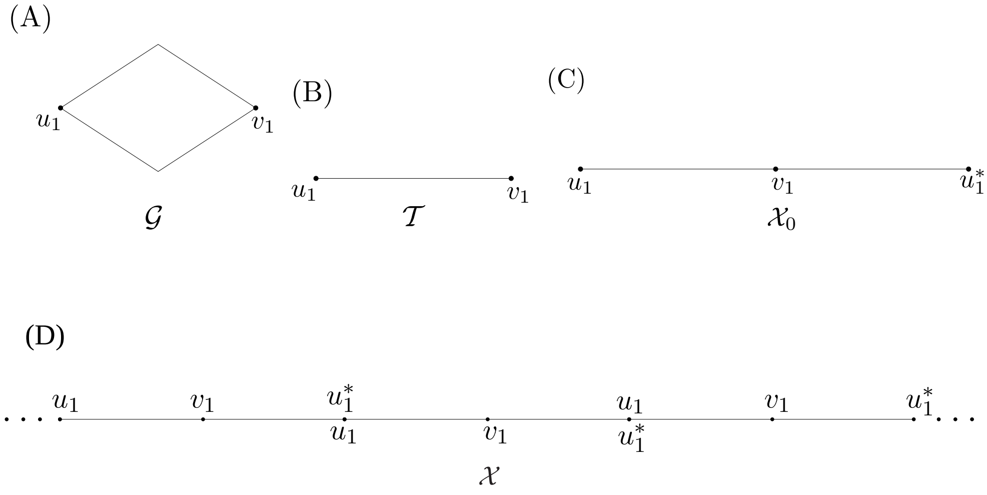


Figure 3: (A) 2-cycle roadmap $\mathcal{G}$. (B) An arbitrary spanning tree of $\mathcal{G}$. (C) The fundamental domain of the universal cover of $\mathcal{G}$. (D) The universal cover of $\mathcal{G}$.

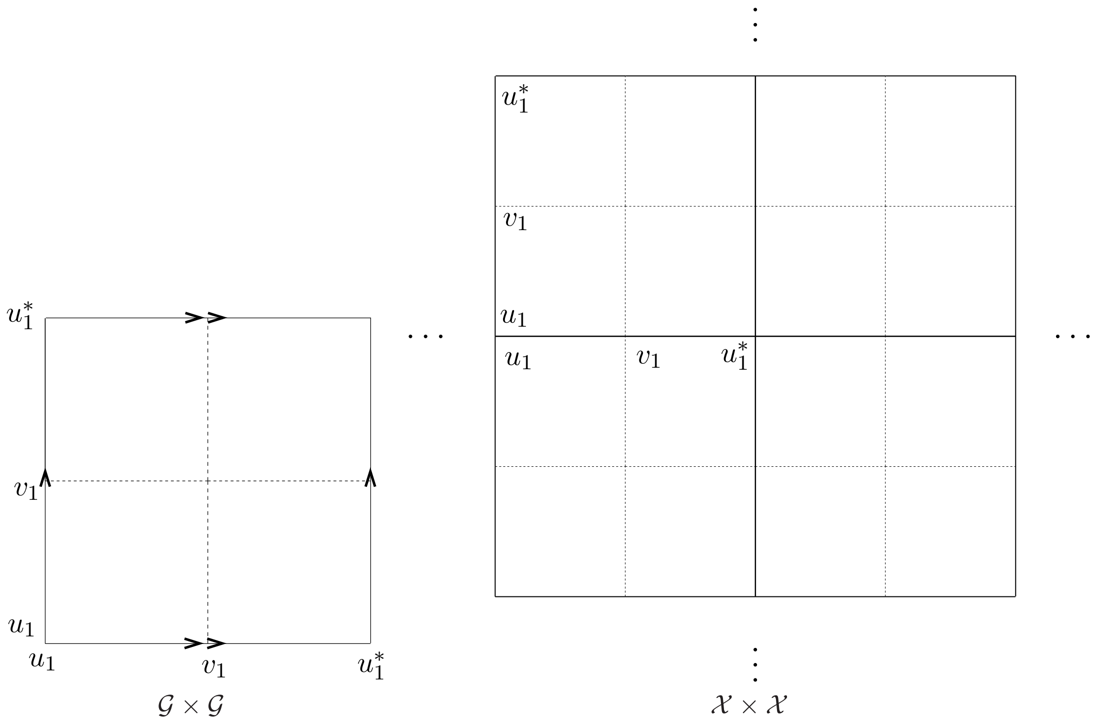


Figure 4: The coordination space of the 2-cycle roadmap, in Figure 3, and its universal cover. [left] The coordination space $\mathcal{G} \times \mathcal{G}$ which is a flat torus. [right] The universal cover of the coordination space.