Efficient Multi-Robot Motion Planning for Unlabeled Discs in Simple Polygons

Aviv Adler, Mark de Berg, Dan Halperin, *Fellow, IEEE*, and Kiril Solovey

***Abstract*—We consider the following motion-planning problem: we are given m unit discs in a simple polygon with n vertices, each at their own start position, and we want to move the discs to a given set of m target positions. Contrary to the standard (labeled) version of the problem, each disc is allowed to be moved to any target position, as long as in the end every target position is occupied. We show that this unlabeled version of the problem can be solved in $O(m^2 + mn)$ time, assuming that the start and target positions are at least some minimal distance from each other. This is in sharp contrast to the standard (labeled) and more general multi-robot motion planning problem for discs moving in a simple polygon, which is known to be strongly NP-hard.**

***Note to Practitioners*—Operating low-cost robots in large groups is becoming a widespread practice. Planning the motion of a group of robots among obstacles is known to be computationally hard. In particular, the worst-case running time of such algorithms grows exponentially in the number of robots. We present a very efficient algorithm for planning collision-free paths for a special case: a set of interchangeable disc (Roomba-like) robots moving in a simple-polygon environment. By interchangeable we mean that we do not care which robot reaches a specific target position, as long as all target positions are occupied at the end of the motion. Our technique is complete and as such it is guaranteed to find a solution if one exists, or report that none exist otherwise. To obtain the efficient solution, we assume a certain minimum separation between the robots in their initial and target positions—we believe that this is the crucial assumption needed to obtain an efficient solution.**

***Index Terms*—Computational geometry, motion planning, multi-robot motion planning.**

Manuscript received December 02, 2014; revised July 14, 2015; accepted August 12, 2015. Date of publication August 26, 2015; date of current version October 02, 2015. This paper was recommended for publication by Associate Editor T. Bretl and Editor F. A. van der Stappen upon evaluation of the reviewers' comments. The work of A. Adler was done while at Tel Aviv University, through a Melvin M. Goldberg Fellowship for Research, Israel. The work of M. de Berg supported was by the Netherlands Organization for Scientific Research (NWO) under Grant 024.002.003. The work of D. Halperin and K. Solovey was supported in part by the Seventh Framework Program for Research of the European Commission under FET-Open Grant 255827 (CGL—Computational Geometry Learning), by the Israel Science Foundation under Grant 1102/11, by the German-Israeli Foundation under Grant 1150-82.6/2011, and by the Hermann Minkowski–Minerva Center for Geometry, Tel Aviv University .

A. Adler is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: aadler1561@gmail.com).

M. de Berg is with the Department of Mathematics and Computing Science, TU Eindhoven, 5612 AZ, Eindhoven, The Netherlands (e-mail: m.t.d.berg@tue.nl).

D. Halperin and K. Solovey are with the Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (e-mail: danha@post.tau.ac.il; kirilsol@post.tau.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2015.2470096

I. Introduction

The MULTI-ROBOT motion-planning problem is to plan the motions of several robots operating in a common workspace. In its most basic form, the goal is to move each robot from its start position to some designated target position, while avoiding collision with obstacles in the environment and with other robots. Besides its obvious relevance to robotics, the problem has various other applications, for example, in the design of computer games or crowd simulation. Multi-robot motion planning is a natural extension of the single-robot motion planning problem, but it is much more complex due to the high number of *degrees of freedom* that it entails, even when the individual robots are as simple as discs.

A. Related Work

One of the first occurrences of the multi-robot motion-planning problem in the computational-geometry literature can be found in the series of papers on the *Piano Movers' Problem* by Schwartz and Sharir. They first considered the problem in a general setting [1] and then narrowed it down to the case of disc robots moving amidst polygonal obstacles [2]. In the latter work, an algorithm was presented for the case of two and three robots, with running time of $O(n^3)$ and $O(n^{13})$, respectively, where n is the complexity of the workspace. Later, Yap [3] used the *retraction method* to develop a more efficient algorithm, which runs in $O(n^2)$ and $O(n^3)$ time for the case of two and three robots, respectively. Several years afterwards, Sharir and Sifrony [4] presented a general approach based on *cell decomposition*, which is capable of dealing with various types of robot pairs and which has a running time of $O(n^2)$. Moreover, several techniques that reduce the effective number of degrees of freedom of the problem have been proposed [5], [6].

When the number of robots is no longer a fixed constant, the multi-robot motion-planning problem becomes hard. Hopcroft *et al.* [7] showed that even in the relatively simple setting of n rectangular robots moving in a rectangular workspace, the problem is already PSPACE-hard. Moreover, Spirakis and Yap [8] showed that the problem is strongly NP-hard for disc robots in a simple polygon.

In recent years, multi-robot planning has attracted a great deal of attention from the robotics community. This can be mainly attributed to two reasons. First, it is a problem of practical importance. Second, the emergence of the sampling-based techniques, which are relatively easy to implement, yet are highly effective, have made the problem relevant. These techniques attempt to capture the connectivity of the configuration space through random sampling [9], [10]. Although sampling-based algorithms are usually incomplete—they are not guaranteed to find a solution—they tend to be very efficient in practice. Hence, they are considered the method-of-choice for motion-planning problems that involve many degrees of freedom. While sampling-based tools for a single robot can be applied directly to the multi-robot problem by considering the group of robots as one large *composite robot* [11], there is a large body of work that attempts to exploit the unique properties of the multi-robot problem [12]–[16].

The aforementioned results deal exclusively with the classical formulation of the multi-robot problem, where the robots are distinct and every robot is assigned a specific target position. The *unlabeled* variant of the problem, where all the robots are assumed to be identical and thus interchangeable, was first considered by Kloder and Hutchinson [17], who devised a sampling-based algorithm for the problem. Recently, a generalization of the unlabeled problem—the k-color motion-planning problem—has been proposed, in which there are several groups of interchangeable robots [18]. Turpin *et al.* [19] considered a special setting of the unlabeled problem with disc robots, namely, where the collection of free configurations surrounding every start or target position is star-shaped. This condition allows them to devise an efficient algorithm that computes a solution in which the maximum path length is minimized. Unfortunately the star-shapedness condition is quite restrictive and, in general, it will not be satisfied. More recently, Solovey *et al.* [20] studied a slightly different setting of the problem in which the start and target positions are assumed to be separated by a distance of at least four (where the radius of every robot is 1), and the area surrounding each such position is obstacle free. They introduced an algorithm which returns a solution whose total length, i.e., the sum of lengths of the individual paths, is near optimal. The running time[1] of the algorithm is $\tilde{O}(m^4 + m^2n^2)$, where m is the number of robots and n represents the complexity of the workspace. Additionally, the cost of the returned solution is at most $\mathrm{OPT} + 4m$, where OPT is the cost of the optimal solution.

Other related work includes papers that study the number of moves required to move a set of discs between two sets of positions in an unbounded workspace, when a move consists of sliding a single disc—see, for example, the paper by Bereg *et al.* [21] which provides upper and lower bounds for the unlabeled case, or the paper by Dumitrescu and Jiang [22] who show that deciding whether a collection of labeled or unlabeled discs can be moved between two sets of positions within k steps is NP-hard. We mention the problem of pebble motions on graph, which can be considered as a discrete variant of the multi-robot motion planning problem. In this problem, pebbles need to be moved from one set of vertices of a graph to another, while following a certain set of rules—see, for example [23]–[28]. Finally, we mention the field of *swarm robotics*, which consists of performing various motion tasks with a large group of primitive robots—see, e.g., [29] and [30].

B. Our Contribution

Surprisingly, the unlabeled version of the multi-robot motion-planning problem has hardly received any attention in the computational-geometry literature. Indeed, we do not know of any papers that solve the problem in an exact and complete manner, except in two settings studied by Turpin *et al.* [19] and Solovey *et al.* [20] that we mentioned above. We therefore study the following basic variant of the problem: given m unit-radius discs (or unit discs, in short) in a simple polygon with n vertices, each at their own start position, and m target positions, find collision-free motions for the discs such that at the end of the motions each disc occupies a target position. We make the additional assumption that the given start and target positions are well-separated. More precisely, any two of the given start and target positions should be at distance at least 4 from each other. Notice that we only assume this extra separation between the robots in their static initial and goal placements; we do not assume any extra separation (beyond noncollision) between a robot and the obstacles, nor do we enforce any extra separation between the robots during the motion. Even this basic version of the problem turns out to have a rich structure and poses several difficulties and interesting questions.

By carefully examining the various properties of the problem we show how to transform it into a discrete pebble-motion problem on graphs.[2] A solution to the pebble problem, which can be generated with rather straightforward techniques, can then be transformed back into a solution to our continuous motion-planning problem. Using this transformation, we are able to devise an efficient algorithm whose running time is $O(m^2 + mn)$, where m is the number of robots and n is the complexity of the workspace. As already mentioned, this is in sharp contrast to the standard (labeled) and more general multi-robot motion planning problem for discs moving in a simple polygon, which is known to be strongly NP-hard [8]. It should also be juxtaposed with a recent result which shows that the unlabeled problem for square robots amid polygonal obstacles is PSPACE-hard, when no separation constraints are imposed [31].

II. Preliminaries

We consider the problem of m indistinguishable unit-disc robots moving in a simple polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$ with n edges. We define $\mathcal{O} \triangleq \mathbb{R}^2 \setminus \mathcal{W}$ to be the complement of the workspace, and we call $\mathcal{O}$ the *obstacle space*. Since our robots are discs, a placement of a robot is uniquely specified by the location of its center. Hence, we will sometimes refer to points $x \in \mathcal{W}$ as *configurations*, and we will say that a robot is *at configuration* x when its center is placed at the point $x \in W$. For given $x \in \mathbb{R}^2$ and $r \in \mathbb{R}_+$, we define $\mathcal{D}_r(x)$ to be the open disc of radius r centered at x.

We consider the unit-disc robots to be open sets. Thus a robot avoids collision with the obstacle space if and only if its center is at distance at least 1 from $\mathcal{O}$, that is, when it is at a configuration

[1]The notation $\tilde{O}$ indicates that log factors are omitted.

[2]We mention that a similar approach was taken in [18] in the context of a sampling-based method, which generates a sequence of pebble graphs by sampling the joint (high-dimensional) configuration space. However, it should be emphasized that our current algorithm is *complete*, i.e., guaranteed to find a solution if one exists, or report that none exists, otherwise. Additionally, it has a polynomial running time. In contrast, the technique described in [18] is unable to detect situations in which a solution does not exist. Moreover, its running time can be exponential in the number of robots or the complexity of the environment. On the positive side, it can deal with various types of robots and does not require that the start and target positions will be well-separated, as we do in the current paper.

located in the *free space* $\mathcal{F} \triangleq \{x \in \mathbb{R}^2 : \mathcal{D}_1(x) \cap \mathcal{O} = \emptyset\}$. We require the robots to avoid collisions with each other, so if a robot is at configuration x then no other robot can be at a configuration $y \in \mathrm{Int}(\mathcal{D}_2(x))$; here $\mathrm{Int}(X)$ denotes the interior of the set X. Furthermore, the notation $\partial(X)$ will be used to refer the boundary of X. We call $\mathcal{D}_2(x)$ the *collision disc* of the configuration x.

Besides the simple polygon $\mathcal{W}$ forming the workspace, we are also given sets $S = \{s_1, s_2, \ldots, s_m\}$ and $T = \{t_1, t_2, \ldots, t_m\}$, such that $S, T \subset \mathcal{F}$. These are, respectively, the sets of *start* and *target* configurations of our m identical disc robots. We assume that the configurations in S and T are *well-separated*:

For any two distinct configurations $x, y \in S \cup T$, we have $\|x - y\| \geqslant 4$.

The problem is now to plan a collision-free motion for our m unit-disc robots such that each of them starts at a configuration in S and ends at a configuration in T. Since the robots are indistinguishable (or: *unlabeled*), it does not matter which robot ends up at which target configuration. Formally, we wish to find paths $\pi_i : [0,1] \rightarrow \mathcal{F}$, for $1 \leqslant i \leqslant m$, such that $\pi_i(0) = s_i$ and $\bigcup_{i=1}^{m} \pi_i(1) = T$. Additionally, we require that the robots do not collide with each other: for every $1 \leqslant i \neq j \leqslant m$ and every $\xi \in (0,1)$, we require $\mathcal{D}_1(\pi_i(\xi)) \cap \mathcal{D}_1(\pi_j(\xi)) = \emptyset$. Note that the requirement that the robots do not collide with the obstacle space $\mathcal{O}$ is implied by the paths π_i being inside the free space $\mathcal{F}$.

III. Basic Properties of the Free Space

Recall that the free space $\mathcal{F} \subset \mathcal{W}$ is the set of configurations at which a robot does not collide with the obstacle space. The free space may consist of multiple connected components. We denote these components by $F_1, \ldots, F_q$, where q is the total number of components. For any $i \in \{1, 2, \ldots, q\}$, we let $S_i \triangleq S \cap F_i$ and $T_i \triangleq T \cap F_i$. We assume from now on that $|S_i| = |T_i|$ for all $1 \leqslant i \leqslant q$—if this is not the case, then the problem instance obviously has no solution—and we define $m_i \triangleq |S_i| = |T_i|$ to be the number of robots in F_i.

Before we proceed, we need one more piece of notation. For any $x \in \mathcal{W}$, we define $\mathrm{obs}(x)$, the *obstacle set* of x, as $\mathrm{obs}(x) \triangleq \{y \in \mathcal{O} : \|x - y\| < 1\}$. In other words, $\mathrm{obs}(x)$ contains the points in the obstacle space overlapping with $\mathcal{D}_1(x)$. Note that $\mathrm{obs}(x) = \emptyset$ for $x \in \mathcal{F}$.

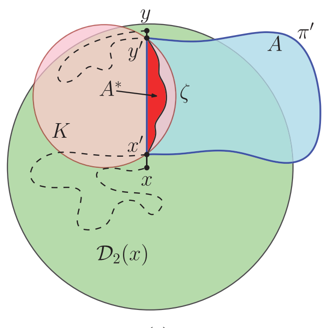
In the remainder of this section, we prove several crucial properties of the free space, which will allow us to transform our problem to a discrete pebble problem. We start with some properties of individual components F_i, and then consider the interaction between robots in different components.

A. Properties of a Single Connected Component of $\mathcal{F}$

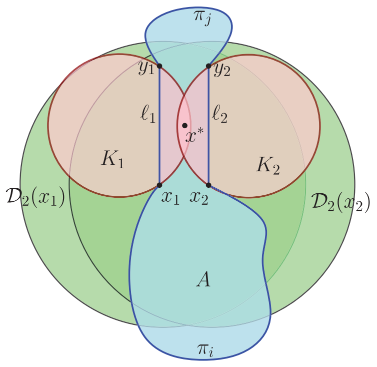
We start with a simple observation, for which we provide a proof for completeness.

Lemma 1: Each component F_i is simply connected.

Proof: Suppose for a contradiction that F_i contains a hole. Then, $\mathrm{Compl}(\mathcal{F}) \triangleq \mathbb{R}^2 \setminus \mathcal{F}$, the complement of the free space, has multiple connected components. One of these is $C_{\mathcal{O}}$, the unbounded component containing $\mathcal{O}$. Let C be another component of $\mathrm{Compl}(\mathcal{F})$, and let $x \in C$. Since $x \notin \mathcal{F}$, there is a point $y \in \mathcal{O}$ with $\|x - y\| < 1$. But then $\|x' - y\| < 1$ for any point x' on the segment $\overline{xy}$, which implies $\overline{xy} \subset \mathrm{Compl}(\mathcal{F})$ and thus contradicts that C and $C_{\mathcal{O}}$ are different components. □



(a)



(b)

Fig. 1. (a) An illustration of Lemma 2. The disc $\mathcal{D}_2(x)$ is drawn in green. The closed curve λ, which consists of the curve π' and the straight-line $\overline{x'y'}$, is drawn in blue, and A represents the area that is bounded by λ. The disc K of radius 1 that touches x', y' is drawn in pink. Note that the area A^*, which is drawn in red, is contained in A. The dashed black lines represent $\pi \setminus \pi'$. (b) An illustration of Lemma 3, and in particular, the case where $A_1^* \cap A_2^* \neq \emptyset$. For simplicity of presentation, we assume that $\ell_1 = \overline{x_1y_1}$ and $\ell_2 = \overline{x_2y_2}$.

Now, consider any x in $\mathbb{R}^2$. Recall that $\mathcal{D}_2(x)$ denotes the collision disc of x, that is, $\mathcal{D}_2(x)$ is the set of all configurations y for which another robot placed at y collides with a robot at x. We now define $D^*(x)$ to be the part of $\mathcal{D}_2(x)$ that is in the same free-space component as x, that is, $D^*(x) \triangleq \mathcal{D}_2(x) \cap F_i$, where F_i is the free-space component such that $x \in F_i$.

The following three lemmas constitute the theoretical basis on which the correctness and efficiency of our algorithm relies.

Lemma 2: For any $x \in \mathcal{F}$, the set $D^*(x)$ is connected.

Proof: Assume for a contradiction that $D^*(x)$ is not connected. Let F_i be the free-space component containing x. Since by definition $x \in D^*(x)$, we can find some $y \in D^*(x)$ that is in a different connected component of $D^*(x)$ from x. Since $y \in D^*(x) \subset \mathcal{D}_2(x)$, the distance between x and y is at most 2. Hence, any point on the line segment $\overline{xy}$ is within a distance of 1 of either x or y. Since $x, y \in F_i$, we know that $\overline{xy} \subset \mathcal{W}$, otherwise either x or y would not be in $\mathcal{F}$. We also know that $\overline{xy} \not\subset F_i$, since, otherwise, x and y would not be in different connected components of $D^*(x)$. Because $x, y \in F_i$, by definition there exists a simple path $\pi \subset F_i$ from x to y. Since the workspace is a polygon with finite description complexity, we may assume that π has finite complexity as well, which implies that $\pi \cap \overline{xy}$ is composed of finitely many isolated points and closed segments. See Fig. 1(a) for an illustration.

We now define x', y' as the points on $\pi \cap \overline{xy} \subset D^*(x)$ such that x', y' are in different connected components of $D^*(x)$ and $\|x' - y'\|$ is minimized given the first condition. Let π' be the subpath of π joining x' to y'. Notice that $\pi \cap \overline{x'y'} = \{x', y'\}$. Indeed, if there exists a point $z \in \pi \cap \mathrm{Int}(\overline{x'y'})$, then z must be in a different connected component of $D^*(x)$ than either x' or y', and $\|x' - y'\|$ would not be the minimum. Since π is a simple path, this means that $\lambda \triangleq \pi' \cup \overline{x'y'}$ is a simple closed curve. The area enclosed by λ (including λ) will be referred to as A. We note that $\lambda \subset \mathcal{W}$ since $\pi' \subset \mathcal{F} \subset \mathcal{W}$ and $\overline{x'y'} \subset \overline{xy} \subset \mathcal{W}$. This immediately implies that $A \subset \mathcal{W}$, since $\mathcal{W}$ is a simple polygon.

Let $A^* \triangleq A \backslash \mathcal{F}$. We claim that $A^* \subset \mathrm{Int}(\mathcal{D}_2(x))$, which implies that there exists a path in F_i from x' to y' that goes along $\partial(A^*)$ and is fully contained in $\mathcal{D}_2(x)$. But this contradicts that x' and y' are in different components of $D^*(x)$ and, hence, proves the lemma. It thus remains to prove the claim that $A^* \subset \mathrm{Int}(\mathcal{D}_2(x))$.

Note that for any point $z \in A^*$ and any $w \in \mathrm{obs}(z)$ we have $\overline{zw} \cap \pi' = \emptyset$, since $\pi' \subset \mathcal{F}$. Furthermore, for any $v \in \pi'$ we have $\|w - v\| \geqslant 1$, and as $x', y' \in \pi'$ it follows that $\|w - x'\| \geqslant 1, \|w - y'\| \geqslant 1$. Assume without loss of generality that $\overline{x'y'}$ is vertical and that locally A lies to the right of $\overline{x'y'}$, as in Fig. 1(a). Let K be the circle of radius 1 that passes through x' and y', and whose center lies to the left of $\overline{x'y'}$—such a circle always exists since $\|x' - y'\| \leqslant \|x - y\| \leqslant 2$. (If $\|x' - y'\| = 2$ then the center of the circle lies on $\overline{x'y'}$.) Let ζ be the arc of this circle lying to the right of $\overline{x'y'}$; note that this is the shorter of the two arcs joining x' and y' if they are of different lengths. Then A^* is contained entirely within the area enclosed by ζ and $\overline{x'y'}$. Furthermore, $A^* \subset \mathrm{Int}(A) \cup \mathrm{Int}(\overline{x'y'})$ since $\pi' \subset \mathcal{F}$. Therefore, since $\overline{x'y'}$ is a subsegment of $\overline{xy}$ and ζ cannot cross $\partial(\mathcal{D}_2(x))$, it follows that:

$$A^* \subset (\mathrm{Int}(A) \cup \mathrm{Int}(\overline{x'y'})) \cap \mathrm{Int}(\mathcal{D}_2(x)) \subset \mathrm{Int}(\mathcal{D}_2(x))$$

which finishes the proof of the lemma. □

B. Interference Between Different Connected Components of $\mathcal{F}$

Let F_i, F_j be two distinct components of $\mathcal{F}$, and let $x \in F_i$ be such that $\mathcal{D}_2(x) \cap F_j \neq \emptyset$. We then call x an interference configuration from F_i to F_j, and define the interference set from F_i to F_j as $I_{(i,j)} \triangleq \{x \in F_i : \mathcal{D}_2(x) \cap F_j \neq \emptyset\}$. We also define the mutual interference set of F_i, F_j as $I_{\{i,j\}} \triangleq I_{(i,j)} \cup I_{(j,i)}$. Intuitively, an interference configuration from F_i to F_j is a configuration for a robot in F_i that could block a path in F_j, and the interference set is the set of all such points. The mutual interference set of F_i, F_j is the set of all single-robot configurations in either component that might block a valid single-robot path in the other component.

Lemma 3: For any mutual interference set $I_{\{i,j\}}$ and any two configurations $x_1, x_2 \in I_{\{i,j\}}$ we have $\mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2) \neq \emptyset$.

Proof: The proof is similar in spirit to the proof of Lemma 2 albeit slightly more involved. Assume for a contradiction that $x_1, x_2 \in I_{\{i,j\}}$ and $\mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2) = \emptyset$. By definition there exist $y_1 \in \mathcal{D}_2(x_1)$ and $y_2 \in \mathcal{D}_2(x_2)$ such that each pair $\{x_1, y_1\}, \{x_2, y_2\}$ contains one point in F_i and one point in F_j. As shown in the proof for Lemma 2, the segments $\overline{x_1y_1}, \overline{x_2y_2}$ are entirely contained in $\mathcal{W}$. We may assume that $\overline{x_1y_1}$ does not cross $\overline{x_2y_2}$, since if it did the crossing point would be in $\mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2)$ and we would be done. Therefore, there exists a simple closed curve $\lambda \subset \mathcal{W}$ composed of the union of two simple curves π_i, π_j and two line segments ℓ_1, ℓ_2 such that $\pi_i \subset F_i$ and $\pi_j \subset F_j$, and $\ell_1 \subset \overline{x_1y_1}, \ell_2 \subset \overline{x_2y_2}$. Note that both ℓ_1 and ℓ_2 have one endpoint in F_i and the other in F_j; see Fig. 1(b) for an illustration. The endpoints of ℓ_1 consist of x_1', y_1', such that x_1, x_1' and y_1, y_1' belong to the same connected components, and minimize the distance $\|x_1' - y_1'\|$ (ℓ_2 is similarly defined).

We refer to the region enclosed by λ (including λ) as A. Because $\lambda \subset \mathcal{W}$ and $\mathcal{W}$ is a simple polygon, we know that $A \subset \mathcal{W}$. Furthermore, since $\pi_i, \pi_j \subset \mathcal{F}$, for any $x \in \mathrm{Int}(A)$ and $y \in \mathrm{obs}(x)$ (by definition, $y \in \mathbb{R}^2 \backslash \mathcal{W}$ so $y \notin A$; thus, $\overline{xy} \cap \lambda \neq \emptyset$), we know that $\overline{xy} \cap \pi_i = \overline{xy} \cap \pi_j = \emptyset$. Thus, $\overline{xy} \cap \mathrm{Int}(\ell_1) \neq \emptyset$ or $\overline{xy} \cap \mathrm{Int}(\ell_2) \neq \emptyset$, or both. Let $A^* \triangleq A \backslash \mathcal{F}$ and denote by A_1^* the set of configurations $x \in A^*$ for which there exists $y \in \mathrm{obs}(x)$ such that $\overline{xy} \cap \mathrm{Int}(\ell_1) \neq \emptyset$; the set A_2^* is defined in a similar manner, only that now $\overline{xy} \cap \mathrm{Int}(\ell_2) \neq \emptyset$. Note that $A^* = A_1^* \cup A_2^*$.

We claim that $A_1^* \cap A_2^* \neq \emptyset$. Indeed, if $A_1^* \cap A_2^* = \emptyset$ then there is a path from x_1 to y_1 along $\partial(A_1^*)$ that stays in $A \setminus A^*$ and, hence, stays in $\mathcal{F}$, which would contradict that $x_1 \in F_i$ and $y_1 \in F_j$ for $i \neq j$. Thus, there exists a point $x^* \in A_1^* \cap A_2^*$. We define the unit circles K_1, K_2 whose boundaries lie on the endpoints of ℓ_1, ℓ_2 respectively, and whose centers are located outside A. Thus, we have $A_1^* \subset K_1$ and $A_2^* \subset K_2$. Hence, $x^* \in K_1 \cap K_2$, which implies $x^* \in \mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2)$, so $\mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2) \neq \emptyset$, contradicting our initial assumption. □

The next lemma is a generalization of the previous one. Intuitively, instead of considering a cycle of length 2 among interacting free-space components, we now consider larger cycles.

Lemma 4: Let $\{\phi(1), \phi(2), \ldots, \phi(h)\} \subset \{1, 2, \ldots, q\}$, and let $x_1, x_2, \ldots, x_h$ be points such that for all i, $x_i \in I_{\{\phi(i), \phi(i+1)\}}$, where $\phi(h+1) \equiv \phi(1)$. (Thus, the list is circular with respect to its index). Then, there exists some $i \neq j$ such that $\mathcal{D}_2(x_i) \cap \mathcal{D}_2(x_j) \neq \emptyset$.

Proof: This can be proved in a manner completely analogous to the proof of Lemma 3; we will outline the proof here. We assume for a contradiction that $\mathcal{D}_2(x_i) \cap \mathcal{D}_2(x_j) = \emptyset$ for all $i \neq j$. We can argue that we can construct a simple closed curve $\lambda \subset \mathcal{W}$ passing through $F_{\phi(1)}, F_{\phi(2)}, \ldots, F_{\phi(h)}$ (in that order), which is composed of simple closed curves $\pi_i \subset F_{\phi(i)}$ and line segments $\ell_i \subset \mathcal{W}$ with endpoints in $F_{\phi(i)}$ and $F_{\phi(i+1)}$. We then consider the area A enclosed by λ and note that $A \subset \mathcal{W}$. Let $A^* \triangleq A \setminus \mathcal{F}$. If there exists some simple curve $\pi^* \subset A^*$ connecting ℓ_i to ℓ_j for some $i \neq j$, we can show that there exists some k such that $\mathcal{D}_2(x_i) \cap \mathcal{D}_2(x_k) \neq \emptyset$, contradicting our assumption. Therefore no such π^* exists for any $i \neq j$. But this means that there exists some simple path $\pi' \subset A \cap \mathcal{F}$ that joins π_i and π_j for some $i \neq j$, which contradicts the fact that π_i and π_j belong to different components of $\mathcal{F}$. □

IV. Algorithm for a Single Component

In this section, we consider a single component F_i of $\mathcal{F}$. We present an algorithm that solves the problem within F_i, ignoring the possibility that robots in F_i might collide with robots in other

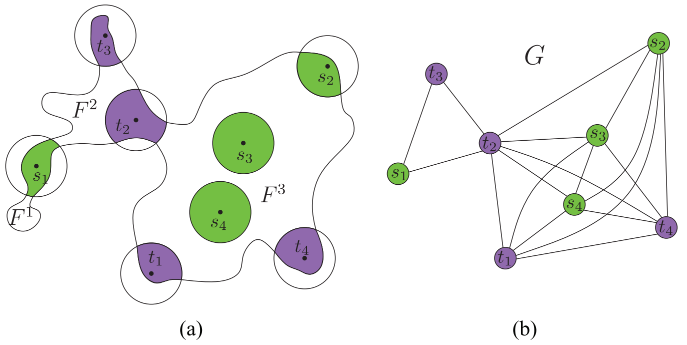


Fig. 2. (a) A partition of a maximal connected component F. The start and target positions consist of the elements $S' = \{s_1, s_2, s_3, s_4\}, T' = \{t_1, t_2, t_3, t_4\}$, respectively, where the areas $D^*(s)$ for $s \in S'$ are drawn in green and $D^*(t)$ for $t \in T'$ are drawn in purple. F^* consists of the parts F^1, F^2, F^3. (b) A motion graph of F.

components F_j. In the next section we will show how to avoid such collisions without changing the motion plans within the individual components. As before we set $S_i \triangleq S \cap F_i$ and $T_i \triangleq T \cap F_i$, and assume $|S_i| = |T_i|$.

A. The Motion Graph

The motion graph G_i of F_i is a graph whose vertices represent start or target configurations, and whose edges represent "adjacencies" between these configurations, as defined more precisely below.

Recall that for any $x \in F_i$ we defined $D^*(x) \triangleq \mathcal{D}_2(x) \cap F_i$ as the part of the collision disc of x inside F_i, and recall from Lemma 2 that $D^*(x)$ is connected. Moreover, for any two distinct configurations $x_1, x_2 \in S_i \cup T_i$ we have $D^*(x_1) \cap D^*(x_2) = \emptyset$, because $\mathcal{D}_2(x_1) \cap \mathcal{D}_2(x_2) = \emptyset$ by the assumption that the start and target positions are well-separated. The vertices of our motion graph G_i correspond to the start and target configurations in $S_i \cup T_i$. From now on, and with a slight abuse of notation we will not distinguish between configurations in $S_i \cup T_i$ and their corresponding vertices in G_i.

Now, consider $F_i^* \triangleq F_i \setminus \bigcup_{x \in S_i \cup T_i} D^*(x)$, the complement of the collision discs of the given start and target configurations in F_i. This complement consists of several connected components, which we denote by $F_i^1, F_i^2, \ldots$. If the motion graph G_i contains an edge (x_1, x_2) then there is a component F_i^ℓ that is adjacent to both $D^*(x_1)$ and $D^*(x_2)$. In other words, two configurations x_1 and x_2 are connected in G_i if there is a path from x_1 to x_2 that stays inside F_i and does not cross the collision disc of any other configuration $x_3 \in S_i \cup T_i$. Fig. 2 illustrates the definition of G_i. The following observation summarizes the main property of the motion graph.

1) Observation 1: Suppose all robots in F_i are located at a start or target configuration in $S_i \cup T_i$, and let (x_1, x_2) be any edge in G_i. Then a robot located at x_1 can move to x_2 without colliding with any of the other robots.

Remark: We could also work with the dual graph of the partitioning of F_i into cells induced by the collision discs. This dual graph would, in addition to vertices representing start and target configurations, also have vertices for the regions F_i^ℓ. For the pebble-motion problem discussed below it is easier to work with the graph as we defined it. This graph may have many more edges, but in the implementation of our algorithm described in Section VI, we avoid computing it explicitly.

B. The Unlabeled Pebble-Motion Problem

Using the motion graph G_i we can view the motion-planning problem within F_i as a pebble-motion problem. (A similar approach was taken in [18], where a sampling-based algorithm for multi-robot motion planning produces multiple pebble problems by random sampling of the configuration space.) To this end, we represent a robot located at configuration $x \in S_i \cup T_i$ by a *pebble* on the corresponding vertex in G_i. The pebbles are indistinguishable, like the robots, and they can move along the edges of the graph. At the start of the pebble-motion problem for a graph with vertex set $S_i \cup T_i$, with $|S_i| = |T_i|$, there is a pebble on every vertex $x \in S_i$. The goal is to move the pebbles such that each pebble ends up in vertex in T_i, under the following conditions: 1) no two pebbles may occupy the same vertex at the same time; and 2) pebbles can only halt at vertices; and 3) at most one pebble may move (that is, be in transit along an edge) at any given time. We call this problem the unlabeled pebble-motion problem. The following lemma follows immediately from Observation 1.

Lemma 5: Any solution to the unlabeled pebble-motion problem on G_i can be translated into a valid collision-free motion plan for the robots in F_i.

Kornhauser [25, Section 3, Lemma 1] proved that the unlabeled pebble-motion problem is, in fact, always solvable, and he gave an algorithm to find a solution. Since he did not analyze the running time of his algorithm, we sketch the solution in the proof of the lemma below.

Lemma 6: For any graph G with vertex set $S \cup T$, where $|S| = |T|$, there exists a solution to the unlabeled pebble-motion problem. Moreover, a solution can be found in $O(|S|^2)$ time [25]

Proof: Let $\mathcal{T}_G$ be a spanning tree of G. The algorithm performs $O(|S|)$ phases. In each phase, one or more pebbles may be moved and one leaf will be removed from $\mathcal{T}_G$, possibly with a pebble on it. After the phase ends, the algorithm continues with the next phase on the modified tree $\mathcal{T}_G$, until all pebbles have been removed and the problem has been solved.

A phase proceeds as follows. If there are leaves v that are target vertices then we select such a leaf v. If v does not yet contain a pebble, we find a pebble closest to v in $\mathcal{T}_G$—this can be done by a simple breadth-first search—and move it to v along the shortest path in G. Note that the vertices on the shortest path cannot contain other pebbles, since we took a closest pebble. We now remove the leaf v, together with the pebble occupying it, and end the phase. If all leaves in $\mathcal{T}_G$ are start vertices, then let v be such a leaf. If v is not occupied by a pebble it can be removed from $\mathcal{T}_G$, and the phase ends. Otherwise a pebble resides in v, which we move away, as follows. We find the closest unoccupied vertex w to v of $\mathcal{T}_G$ and move all pebbles on this shortest path (including the pebble on v) one step closer to w, in order of decreasing distance from w. After we evacuated v we remove it from $\mathcal{T}_G$ to end the phase.

The algorithm produces paths of total length $O(|S|^2)$, and it can easily be implemented to run in $O(|S|^2)$ time. In some cases $\Omega(|S|^2)$ moves are required, e.g., when G is a single path with all start positions in the first half of the path and all target positions in the second half. □

Lemma 7: Suppose we have an instance of our multi-robot path planning problem, where $|S_i| = |T_i|$ for every component F_i of the free space $\mathcal{F}$. Then, for each F_i there exists a motion plan Π_i that brings the robots in F_i from S_i to T_i, such that they do not collide with the obstacle space nor with the other robots in F_i.

V. Combining Single-Component Plans

We now consider possible interactions between robots contained in different components F_i and F_j of $\mathcal{F}$. As before, we assume that $|S_i| = |T_i|$ for all i. We will show that there exists a permutation $\sigma : \{1, 2, \ldots, \ell\} \to \{1, 2, \ldots, \ell\}$ such that we can independently execute the single-component motion plans for each component F_i as long as we do so in the order $F_{\sigma(1)}, F_{\sigma(2)}, \ldots, F_{\sigma(\ell)}$.

To obtain this order, we define a directed graph representing the structure of $\mathcal{F}$, which we call the directed-interference forest $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes in $\mathcal{V}$ correspond to the components F_i. We add the directed edge (F_i, F_j) to $\mathcal{E}$ if either there exists a start position $s \in S$ such that $s \in I_{(i,j)}$, or there exists a target position $t \in T$ such that $t \in I_{(j,i)}$. For any $i \in \{1, 2, \ldots, \ell\}$, we additionally define $N^+(i)$ to be the set of indices of the vertices in the out-neighborhood of v_i; similarly, $N^-(i)$ is defined as the set of indices of the vertices in the in-neighborhood of v_i.

Note that by Lemma 3 and since S, T are well separated, we cannot have more than one start or target position in $I_{\{i,j\}}$. This implies that $\mathcal{E}$ cannot contain both (v_i, v_j) and (v_j, v_i). Lemma 4 and the well-separatedness condition additionally imply that $\mathcal{G}$ cannot have an undirected cycle. Thus, $\mathcal{G}$ is a directed forest.

We now produce the desired ordering using $\mathcal{G}$. Consider $F_i \in \mathcal{V}$, and suppose that for all $j \in N^+(i)$, every robot in F_j is at a start position, and for all $j \in N^-(i)$, every robot in F_j is at a target position. Additionally, suppose that for all $j \notin N^+(i) \cup N^-(i)$, every robot in F_j is at a start or target position. Then, by the definition of $\mathcal{G}$, no robot is at a configuration in $I_{\{i,j\}}$ for any $j \neq i$; thus any motion plan for the robots in F_i, such as the one described in Section IV, can be carried out without being blocked by the robots not in F_i. Hence, if we have an ordering $\sigma : \{1, 2, \ldots, \ell\} \to \{1, 2, \ldots, \ell\}$ such that for all (directed) edges $(v_i, v_j) \in \mathcal{E}$, $\sigma^{-1}(i) < \sigma^{-1}(j)$, where σ^{-1} is the inverse permutation of σ, then we can execute the motion plans for the robots in $F_{\sigma(1)}, F_{\sigma(2)}, \ldots, F_{\sigma(\ell)}$ in that order. Since $\mathcal{G}$ is a directed forest such an ordering can be produced using topological sorting on the vertices of $\mathcal{G}$. Thus, combining this result with Lemma 7 we obtain the following.

Theorem 8: Let there be a collection of m unlabeled unit-disc robots in a simple polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$, with start and target configurations S, T that are well-separated. Then, if for every maximal connected component F_i of $\mathcal{F}$ (where $\mathcal{F}$ is the free space for a single unit-disc robot in $\mathcal{W}$) with $|S \cap F_i| = |T \cap F_i|$, there exists a collision-free motion plan for these robots starting at S that terminates with every position of T occupied by a robot.

VI. Algorithmic Details

In this section, we fill in a few missing details in the description of our algorithm. Specifically, we present an efficient method for generating motion graphs and describe a technique for generating configuration-space paths that correspond to edges in the motion graphs. We also consider the complexity of the various subsets of $\mathcal{F}$ used throughout the algorithm.

A. Partitioning $\mathcal{F}$

We analyze the combinatorial complexity of $\mathcal{F}^* \triangleq \mathcal{F} \setminus \bigcup_{x \in S \cup T} D^*(x)$ and $\mathbb{D} \triangleq \bigcup_{x \in S \cup T} D^*(x)$.

Lemma 9: The combinatorial complexity of $\mathcal{F}^*$ and $\mathbb{D}$ is $O(m + n)$.

Proof: We start with $\mathcal{F}^*$ and decompose the complement of the workspace polygon into $O(n)$ trapezoids—this is doable by standard vertical decomposition. We define a set X, which consists of the trapezoids, and in addition a collection of $O(m)$ unit discs that are centered at the start and target positions. We now observe that the regions in X are pairwise interior disjoint (and convex). Hence, it is known [32] that the complexity of the union of the regions in X, each Minkowski-summed with a unit disc, is linear in the number of regions plus the sum of the complexities of the original regions. As the result of the Minkowski sum operation of X with a unit disc is the area $\mathcal{F}^*$, we conclude that the complexity of $\mathcal{F}^*$ is $O(m + n)$.

Due to the fact that every vertex of the boundary of $\mathbb{D}$ is either a vertex of $\mathcal{F}^*$ or a vertex of $\mathcal{F}$, it immediately follows that the complexity of $\mathbb{D}$ is $O(m + n)$ as well. □

Note that this upper bound still holds if we consider instead of $\mathcal{F}^*$ the union of $F_i^* \triangleq F_i \setminus \bigcup_{x \in S_i \cup T_i} D^*(x)$, for all $1 \leqslant i \leqslant q$.

B. Generating the Motion Graphs

We consider a specific component F of $\mathcal{F}$ and construct its motion graph G. Denote $F^* \triangleq F \setminus \bigcup_{x \in (S \cup T) \cap F} D^*(x)$. Note that by the analysis in Section V we can ignore the influence of $\mathcal{D}_2(x)$ on connected components in $\mathcal{F}$ that do not contain x. We assume that F^* breaks into k maximal connected components $F^1, \ldots, F^k$. The construction of G, along with the paths in F that correspond to the edges of G, is carried out in two steps. First, for every F^i, we generate the portion of G, denoted by G^i, whose vertices represent start and target positions that touch the boundary of F^i. Then, we connect between the various parts of G.

We consider a specific connected component F^i of F^* and describe how the respective portion of the motion graph, namely G^i, is generated. We split the start and target positions that share a boundary with F^i into two subsets: B^i are those positions for which the collision disc intersects the boundary of F and H^i are those positions for which the collision disc floats inside F. See Fig. 3. We first handle positions in B^i. Consider the outer boundary Γ^i of $F^i \setminus \bigcup_{x \in S \cup T} D^*(x)$. We argue that each $x \in B^i$ can contribute exactly one piece to Γ^i.

Lemma 10: If $x \in B^i$ then $\partial(\mathcal{D}_2(x)) \cap \partial(F^i)$ consists of a single component.

Proof: By contradiction, assume that the intersection consists of two maximal connected components. Denote by y, y' two configurations on the two components. As F^i consists of a single connected component of F there exists a path $\pi_{yy'} \subset F$ from y to y'. Additionally, as y, y', x belong to the same connected component of $\mathcal{F}$ there exist two paths—π_{xy} from x to y and $\pi_{xy'}$ from x to y'—that lie entirely in $D^*(x)$. Thus, the

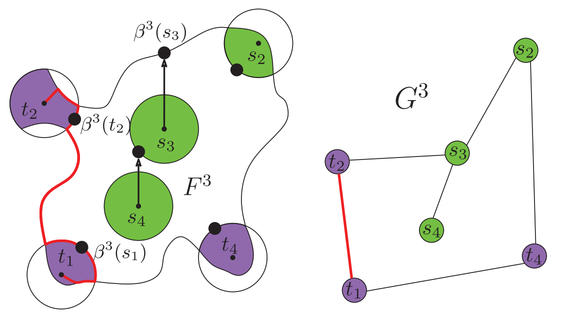


Fig. 3. (a) An illustration of a component F^3 of F^* and the structures used for generating the relevant portion of the motion graph. The boundary positions of F^3 consist of $B^3 := \{s_2, t_1, t_2, t_4\}$, while the hole positions consist of $H^3 := \{s_3, s_4\}$. For every $x \in H^3$ its boundary representative $\beta^3(x)$ is illustrated as a large black dot. A path between t_1 and t_2 is illustrated in red. (b) The motion graph G^3 induced by F^3.

area that is bounded by the three paths $\pi_{yy'}, \pi_{xy}, \pi_{xy'}$ contains a patch of forbidden space, which contradicts the fact the our workspace is a simple polygon. □

For every $x \in B^i$, we arbitrarily select a representative point $\beta^i(x) \in \partial(\mathcal{D}_2(x)) \cap F^i$. We order the points $\beta^i(x)$ clockwise around Γ^i, and store them in a circular list $\mathcal{L}^i$. We now incorporate the remaining start and target positions H^i, namely, those positions x for which $D_2(x) \cap \partial(F) = \emptyset$. Each position in H^i will be connected either to Γ^i or to the boundary of a collision disc of another position in H^i as follows. For each $x \in H^i$ we shoot a vertical ray upwards until it hits $\partial(F^i)$. Denote the point where the ray hits $\partial(F^i)$ by c. If $c \in \partial(\mathcal{D}_2(x'))$ for some $x' \in H^i, x' \neq x$ then an edge between x and x' is added to G^i. Otherwise, we let $\beta^i(x) \triangleq c$ and insert it into the circular list $\mathcal{L}^i$ representing the points $\beta^i(x)$ along Γ^i collected so far. After all positions in H^i have been handled in this manner, for each pair of consecutive points $\beta^i(x'), \beta^i(x'')$ in $\mathcal{L}^i$ (along Γ^i) we add an edge in G^i between the vertices x' and x''. (Notice that some of the positions x whose $\beta^i(x)$ appear in $\mathcal{L}^i$ belong to H^i; for example, s_3 in Fig. 3.) Finally, the connection between portions of the motion graph that represent different parts of F^* is achieved through positions shared between two sets B^i, B^j, for $i \neq j$.

C. *Transforming Graph Edges Into Paths in the Free Space*

There are three different types of transformations depending on how the edge was created. Let (x, x') be an edge in G_i. Consider Fig. 3 for an illustration. (i) If both x and x' belong to H^i (see (s_4, s_3) in the figure) then the path simply consists of the two straight-line segments $\overline{xc}$ and $\overline{cx'}$. For the remaining two cases we note that if either vertex, say x, is in B^i, then part of the path is a simple curve connecting x to $\beta^i(x)$ within $D^*(x)$ (see the red curves from s_1 and t_2 in the figure). We denote this curve by δ_x. (ii) $x, x' \in B^i$ and the points $\beta^i(x)$ and $\beta^i(x')$ are consecutive along Γ^i (see (t_1, t_2) in the figure). The path corresponding to the edge (x, x') in this case is a concatenation of three sub-paths: δ_x, the portion of Γ^i between $\beta^i(x)$ and $\beta^i(x')$ (not passing though the boundary of any other collision disc), and $\delta_{x'}$. (iii) $x \in H^i$ and $x' \in B^i$ (see (s_3, s_2) in the figure). The path is again a concatenation of three paths: the line segment $\overline{x\beta^i(x)}$, the portion of Γ^i between $\beta^i(x)$ and $\beta^i(x')$ (not passing though the boundary of any other collision disc), and $\delta_{x'}$.

Notice that for all path types above if a robot r moves from x to x', x' is not occupied, and all other robots occupy positions only at $S \cup T \setminus \{x, x'\}$, r will not collide with any other robot during the motion.

D. *Complexity Analysis*

We provide complexity analysis of our algorithm and show that a solution to the problem can be produced within $O(mn + m^2)$ operations.

Recall that the pebble problem solver (Section IV) operates in $O(m)$ phases, where in each phase a leaf node is removed from the spanning tree of G. We show, using Lemma 9 that each phase can be transformed into a set of movements for the robots whose combinatorial complexity is $O(m+n)$. The crucial observation is that in one phase each edge of the motion graph is used at most once. Thus the set of robot movements in one phase is bounded by the complexity of the movements corresponding to all the edges in the graph together. These comprise $O(m)$ line segments, portions of the boundaries Γ^i (whose complexity is $O(m+n)$ by Lemma 9), and the paths δ_x inside the $D^*(x)'s$ (whose complexity is $O(m+n)$ as well). A path of the latter type, δ_x, might be traversed twice: once for reaching x and once for leaving x. Asymptotically all the movements together have complexity $O(m+n)$. Therefore, the overall complexity of the motions is described by $O(m^2 + mn)$.

The generation of $\mathcal{F}$ can be performed by constructing the medial axis of $\mathcal{W}$ [33] in $O(n)$ time. The construction of $\mathcal{F}^*$ and $\mathbb{D}$ is done straightforwardly by performing the following two operations for every $x \in S \cup T$. First, we check whether $\mathcal{D}_2(x) \cup \partial(\mathcal{F}) \neq \emptyset$ by intersecting $\mathcal{D}_2(x)$ with every edge of $\mathcal{F}$. In case that no intersection is found, we find the connected component of $\mathcal{F}$ that contains x by point-in-polygon test [34]. These two checks take $O(n)$ time for every such x, and $O(mn)$ time in total. In conclusion, we have the following theorem.

Theorem 11: Let $\mathcal{W}$ be a simple polygon with n vertices and let $S = \{s_1, \ldots, s_m\}, T = \{t_1, \ldots, t_m\}$ be two sets of m points in $\mathcal{W}$. Additionally, assume that for every two distinct element x, x' of $S \cup T$ it holds that $\|x - x'\| \geqslant 4$. Then, given m unlabeled unit disc robots, our algorithm can determine whether a path moving the m robots from S to T exists, and find a solution in $O(m^2 + mn)$ time.

VII. Separation and Solvability

Our results from the previous section imply that a separation distance of four ensures that the problem always has a solution, assuming that each connected component contains the same number of start and target positions. However, for smaller separation values this does not have to be the case.

We define a magnitude λ to be the minimum separation between start and target positions of unlabeled discs in a simple polygon, that guarantees that a solution always exist (assuming that each connected component contains the same number of start and target positions). Our work in the previous sections

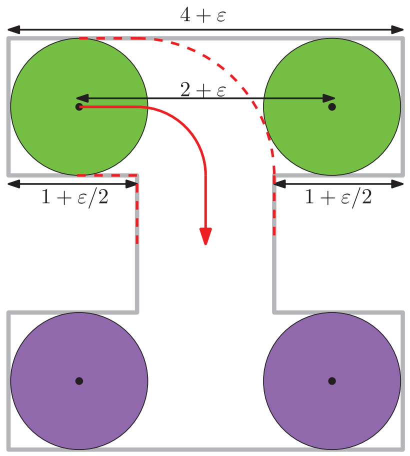


Fig. 4. In the given example, the two robots need to leave through the corridor of width 2 located below their initial positions. This is only possible when $\epsilon \geqslant 4\sqrt{2}-2$, as otherwise the robots will not be able to get to the corridor without colliding with each other. The dashed red segments depict the trace of the boundary of the left robot, while getting to the corridor (whose width is 1). The red arrow depicts the path of the left robot. When $\epsilon < 4\sqrt{2}-4$, the right dashed line will penetrate the robot located in the right start position, and consequently the two robots will be in collision.

shows that $\lambda \leqslant 4$. The following proposition provides a lower bound for the value of λ.

Proposition 12: $\lambda \geqslant 4\sqrt{2}-2 (\approx 3.646)$.

Proof: We describe a concrete example where the value of λ has to be greater than $4\sqrt{2}-2$ to guarantee solvability (see Fig. 4). While the proof is rather technical and straightforward, we mention that the main observation here is that the distance between the left corner of the corridor, around which the left robot has to rotate, has to be at distance of at least three from the start position from the right robot. □

VIII. Open Problems and Future Work

We have studied a basic variant of the multi-robot motion-planning problem, where the goal is to find collision-free motions that bring a given set of indistinguishable unit discs in a simple polygon to a given set of target positions. Under the condition that the start and target positions are separated from each other by a distance of at least four, we developed an algorithm that solves the problem in time polynomial in the complexity of the polygon as well as in the number of discs: quadratic in the number of robots and near-linear in the complexity of the polygon.

Our result should be contrasted with the labeled counterpart of the problem, which is NP-hard [8]. In the NP-hardness proof, the discs have different radii, however, and there is no restriction on the separation of the start and target position. Thus, one of the main open questions resulting from our study is to settle the complexity of the unlabeled problem without this extra separation condition. Very recently, we have made progress in this direction, by showing that the general unlabeled problem is PSPACE-hard [31]. It should be noted that this proof applies to a slightly different setting that consists of unit-square robots translating amidst polygonal obstacles.

A natural question that arises is what happens when the separation of start or target positions is equal to $2+\varepsilon$, where $0 < \varepsilon < 2$? Would it be possible to design an algorithm, whose running time is polynomial in m and n, and also depends in some manner on ε? We believe that the work by Alt *et al.* [35] would be a good starting point for this line of work. Following our discussion in Section VII, it will also be interesting to find the exact threshold above which a problem is always solvable.

References

[1] J. T. Schwartz and M. Sharir, "On the Piano Movers problem: II. General techniques for computing topological properties of real algebraic manifolds," *Adv. Appl. Math.*, vol. 4, no. 3, pp. 298–351, 1983.
[2] J. T. Schwartz and M. Sharir, "On the Piano Movers problem: III. Coordinating the motion of several independent bodies," *Int. J. Robot. Res.*, vol. 2, no. 3, pp. 46–75, 1983.
[3] C.-K. Yap, "Coordinating the motion of several discs," Michigan State Univ., Courant Inst. Math. Sci., New York, NY, USA, Tech. Rep., 1984.
[4] M. Sharir and S. Sifrony, "Coordinated motion planning for two independent robots," *Ann. Math. Artif. Intell.*, vol. 3, no. 1, pp. 107–130, 1991.
[5] B. Aronov, M. de Berg, A. F. van der Stappen, P. Švestka, and J. Vleugels, "Motion planning for multiple robots," *Discrete Comput. Geomet.*, vol. 22, no. 4, pp. 505–525, 1999.
[6] J. van den Berg, J. Snoeyink, M. C. Lin, and D. Manocha, "Centralized path planning for multiple robots: Optimal decoupling into sequential plans," in *Proc. Robot.: Sci. Syst. (RSS)*, Seattle, WA, USA, 2009.
[7] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "Warehouseman's problem"," *Int. J. Robot. Res.*, vol. 3, no. 4, pp. 76–88, 1984.
[8] P. G. Spirakis and C.-K. Yap, "Strong NP-hardness of moving many discs," *Inform. Process. Lett.*, vol. 19, no. 1, pp. 55–59, 1984.
[9] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
[10] J. J. Kuffner and S. M. Lavalle, "RRT-Connect: An efficient approach to single-query path planning," in *Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 995–1001.
[11] G. Sanchez and J.-C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Washington, DC, USA, 2002, pp. 2112–2119.
[12] S. Hirsch and D. Halperin, "Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane," in *Proc. Workshop Algorithmic Foundations of Robot. (WAFR)*, 2002, pp. 239–255.
[13] O. Salzman, M. Hemmer, and D. Halperin, "On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 529–538, Apr. 2015.
[14] K. Solovey, O. Salzman, and D. Halperin, "Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning," in *Proc. Algorithmic Found. Robot. XI*, 2014, pp. 591–607.
[15] P. Švestka and M. H. Overmars, "Coordinated path planning for multiple robots," *Robot. Auton. Syst.*, vol. 23, pp. 125–152, 1998.
[16] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artif. Intell.*, vol. 219, pp. 1–24, 2015.
[17] S. Kloder and S. Hutchinson, "Path planning for permutation-invariant multi-robot formations," in *Proc. ICRA*, 2005, pp. 1797–1802.
[18] K. Solovey and D. Halperin, "k-color multi-robot motion planning," *Int. J. Robot. Res.*, 2013, to be published.
[19] M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of interchangeable robots," *Auton. Robots*, vol. 37, no. 4, pp. 401–415, 2014.
[20] K. Solovey, J. Yu, O. Zamir, and D. Halperin, "Motion planning for unlabeled discs with optimality guarantees," in *Proc. Robot.: Sci. Syst. (RSS)*, Rome, Italy, 2015.
[21] S. Bereg, A. Dumitrescu, and J. Pach, "Sliding disks in the plane," *Int. J. Comput. Geometry Appl.*, vol. 18, no. 5, pp. 373–387, 2008.
[22] A. Dumitrescu and M. Jiang, "On reconfiguration of disks in the plane and related problems," *Computat. Geometry: Theory Appl.*, vol. 46, no. 3, pp. 191–202, 2013.
[23] O. Goldreich, "Shortest move-sequence in the generalized 15-puzzle is NP-hard," *Manuscript, Laboratory for Computer Sci., MIT*, vol. 1, 1984.

[24] G. Goraly and R. Hassin, "Multi-color pebble motion on graphs," *Algorithmica*, vol. 58, no. 3, pp. 610–636, 2010.
[25] D. Kornhauser, "Coordinating pebble motion on graphs, the diameter of permutation groups, and applications," M.Sc. thesis, Dept. Elect. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA, USA, 1984.
[26] A. Krontiris, R. Luna, and K. E. Bekris, "From feasibility tests to path planners for multi-agent pathfinding," in *Proc. Symp. Combinatorial Search*, Leavenworth, WA, USA, 2013.
[27] C. H. Papadimitriou, P. Raghavan, M. Sudan, and H. Tamaki, "Motion planning on a graph," in *Found. Comput. Sci.*, 1994, pp. 511–520.
[28] J. Yu and S. M. LaValle, "Distance optimal formation control on graphs with a tight convergence time guarantee," in *Proc. IEEE Int. Conf. Decision Control*, 2012, pp. 4023–4028.
[29] T.-R. Hsiang, E. M. Arkin, M. A. Bender, S. P. Fekete, and J. S. Mitchell, "Algorithms for rapidly dispersing robot swarms in unknown environments," in *Algorithmic Foundations of Robotics V*. New York, NY, USA: Springer, 2004, pp. 77–93.
[30] A. Becker, S. P. Fekete, A. Kröller, S. Lee, J. McLurkin, and C. Schmidt, "Triangulating unknown environments using robot swarms," in *Proc. Symp. Comput. Geometry, SoCG'13*, Rio de Janeiro, Brazil, Jun. 17–20, 2013, pp. 345–346.
[31] K. Solovey and D. Halperin, "On the hardness of unlabeled multi-robot motion planning," in *Proc. Robot.: Sci. Syst. (RSS)*, Rome, Italy, 2015.
[32] K. Kedem, R. Livne, J. Pach, and M. Sharir, "On the union of jordan regions and collision-free translational motion amidst polygonal obstacles," *Discrete Comput. Geomet.*, vol. 1, pp. 59–70, 1986.
[33] F. Y. L. Chin, J. Snoeyink, and C. A. Wang, "Finding the medial axis of a simple polygon in linear time," *Discrete Comput. Geomet.*, vol. 21, no. 3, pp. 405–420, 1999.
[34] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars*, Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin, Germany: Springer-Verlag, 2008.
[35] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig, "Approximate motion planning and the complexity of the boundary of the union of simple geometric figures," *Algorithmica*, vol. 8, no. 1–6, pp. 391–406, 1992.

Aviv Adler received the B.A. degree in mathematics from Princeton University, Princeton, NJ, USA, in 2014. He is currently working towards the Ph.D. degree at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA.

His primary research interests are algorithms, computational geometry, graph theory, and motion planning.

Mark de Berg received the Ph.D. degree in computer science from Utrecht University, Utrecht, The Netherlands, in 1988.

After receiving the Ph.D. degree, he became a faculty member at Utrecht University. In 2002, he moved to the TU Eindhoven, where he became a Full Professor and Head of the TU/e Algorithms Group. His research area is algorithms and data structures and, in particular, computational geometry. He is (co-)author of two books on computational geometry, one of which has become the standard textbook in the area, and he published over 190 papers in peer-reviewed journals and conferences.

Prof. de Berg was on the program committee of numerous international conferences on computational geometry and on algorithms, and he serves on the editorial board of three international journals.

Dan Halperin received the Ph.D. degree in computer science from Tel-Aviv University, Tel-Aviv, Israel.

He then spent three years at the Computer Science Robotics Laboratory, Stanford University. In 1996, he joined the Department of Computer Science, Tel-Aviv University, where he is currently a Full Professor and for two years was the Department Chair. His main field of research is computational geometry and its applications. A major focus of his work has been in research and development of robust geometric software, principally as part of the CGAL project and library. The application areas he is interested in include robotics and automated manufacturing, and algorithmic motion planning.

Kiril Solovey is currently working towards the Ph.D. degree at the School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, under the supervision of Prof. Halperin.

His research interests include sampling-based algorithms for motion planning, multirobot motion planning, and computational geometry.