Motion Planning for Multiple Robots

Boris Aronov*   Mark de Berg†   A. Frank van der Stappen†   Petr Švestka‡

Jules Vleugels†

Abstract

We study the motion-planning problem for pairs and triples of robots operating in a shared workspace containing n obstacles. A standard way to solve such problems is to view the collection of robots as one composite robot, whose number of degrees of freedom is d, the sum of the numbers of degrees of freedom of the individual robots. We show that it is sufficient to consider a constant number of robot systems whose number of degrees of freedom is at most $d-1$ for pairs of robots, and $d-2$ for triples. (The result for a pair assumes that the sum of the number of degrees of freedom of the robots constituting the pair reduces by at least one if the robots are required to stay in contact; for triples a similar assumption is made. Moreover, for triples we need to assume that a solution with positive clearance exists.)

We use this to obtain an $O(n^d)$ time algorithm to solve the motion-planning problem for a pair of robots; this is one order of magnitude faster than what the standard method would give. For a triple of robots the running time becomes $O(n^{d-1})$, which is two orders of magnitude faster than the standard method. We also apply our method to the case of a collection of bounded-reach robots moving in a low-density environment. Here the running time of our algorithm becomes $O(n \log n)$ both for pairs and triples.

1   Introduction

One of the ultimate goals of robotics research is to create robots that are capable of autonomous action, and accept and execute high-level task descriptions while requiring little or no human supervision. A fundamental task for an autonomous robot is to plan its own motion: it should be able to figure out how to get from one position (for instance, where it has picked up some object) to another position (where the object is to be delivered). In general there will be obstacles in the workspace of the robot, which it has to avoid. In many applications the situation is further complicated by the fact that the robot has to share its workspace with other robots. Examples of this are the transportation systems found at modern airports, ports, and factories. This is the setting of the motion-planning problem we study: given a collection $R_1, \ldots, R_m$ of robots with $d_1, \ldots, d_m$ degrees of freedom, respectively, and operating in a shared workspace $\mathcal{W}$ with n obstacles, find a collection of paths bringing each robot from a specified start position to a specified goal position without colliding with the obstacles or the other robots, or report that the problem is not solvable. We assume m is a small constant; we will mainly study the cases $m = 2$ and $m = 3$. The problem becomes PSPACE-complete if the number of robots is not constant [13].

Two established approaches to this problem are *decoupled planning* and *centralized planning*.

The decoupled planning approach [1, 8, 15, 23] first plans the motion of each robot individually while ignoring the existence of the other robots, then tries to combine the resulting paths by

resolving possible collisions between the paths. Algorithms following this approach are usually incomplete in the sense that they are not guaranteed to find a solution if one exists.

In centralized planning [17, 19] the m robots are regarded as one multi-robot, that is, one robot with several independent body parts that are not necessarily connected to each other. Collisions between the robots R_i turn into collisions between different body parts—in other words, self-collisions—of the multi-robot. The configuration space of the multi-robot is $\mathcal{CS}_1 \times \cdots \times \mathcal{CS}_m$, where $\mathcal{CS}_i$ is the configuration space of robot R_i. The dimension of this configuration space is $d_1 + \cdots + d_m$, the sum of the dimensions of the individual configuration spaces. Using an algorithm of Basu et al. [2] for constructing roadmaps, one can thus solve the motion-planning problem in $O(n^{d_1+\cdots+d_m+1})$ time. Under certain general-position assumptions on the configuration-space obstacles one can use Canny's roadmap algorithm [4] to improve the time bound to $O(n^{d_1+\cdots+d_m} \log n)$. (In a later paper [5] Canny showed how to eliminate the general-position assumption, but unfortunately the adapted version cannot report an actual path, it can only decide on the existence of a path.) Since we wish to keep our results as general as possible, we stick to using the roadmap algorithm of Basu et al. from this point on.

Centralized approaches have the advantage that they allow for complete planners, which are guaranteed to find a solution whenever one exists. Their main drawback is that the dimension of the configuration space of the multi-robot is much higher than that of the individual robots. This increases both the combinatorial and the algebraic complexity of the problem.

Sharir and Sifrony [19] present a general centralized-planning method that is sometimes more efficient than applying the method of Basu et al. to the multi-robot. Their method requires a cell decomposition for the free portion of each individual configuration space, such that each cell has constant complexity and is adjacent to at most a constant number of other cells. The cell decompositions are then combined into a representation of the free space of the multi-robot. The complexity of this representation is the product of the complexities of the individual configuration spaces. This approach can take advantage of the fact that sometimes the complexity of (the cell decomposition of) the free space of a robot R_i is significantly smaller than $\Theta(n^{d_i})$. For instance, it is well known that the complexity of the free space of a polygon translating amidst n polygonal obstacles in the plane is only $O(n)$ [12]. Thus the method of Sharir and Sifrony can plan the motion of two polygons in such a workspace in $O(n^2)$ time. Applying the method of Basu et al. to the multi-robot would lead to an $O(n^5)$ algorithm, because the number of degrees of freedom of the multi-robot is four.

We present a refinement of the centralized-planning approach for pairs and triples of robots. Our technique is quite general: it works for any type of robots, and it can be combined with roadmap methods and with cell-decomposition methods. The technique reduces the dimension of the configuration space one has to consider for the multi-robot. For pairs of robots it does so by stipulating that at all times either one of the robots should be at its start or goal position, or the robots should touch each other; for triples of robots the configurations of the multi-robot are constrained in a similar fashion. (Throughout the paper we make the assumption that the sum of the number of degrees of freedom of two robots reduces by at least one if the robots are required to stay in contact; for triples a similar assumption is made. Moreover, for triples we need to assume that a solution with positive clearance exists.)

The approach of reducing the dimension of the configuration space to be searched was also used by Hopcroft and Wilfong [11, 10] and Fortune et al. [9]. Hopcroft and Wilfong [11] showed that if the robots form a single connected component at their start and goal configurations and if there exists a collision-free motion of the multi-robot in which the robots need not touch, then there exists a path such that the robots form a connected component throughout the entire motion. This implies that the search in the configuration space can be limited to a lower-dimensional subspace. However, their result only holds when the configuration space is contractible to a point. In a later paper Fortune et al. [9] considered the case of two planar robot arms, each having one extendible and rotatable link. For this case, where the configuration space is not contractible to a point, they gave a proof (tailored to this special case) that it is sufficient to consider only motions in contact; in addition, they developed an algorithm to find such motions. In their second paper [10], Hopcroft

and Wilfong applied the result from their first paper to translating axis-parallel polygons in $\mathbb{R}^2$. In particular, they showed that planning the motions of n rectangles in a rectangular workspace is in PSPACE.

We use essentially the same approach, that is, we also show that the search of the configuration space can be limited to lower-dimensional subspaces, corresponding to configurations with specific properties. However, we do not require the configuration space to be contractible, thus generalizing the results of Hopcroft and Wilfong to arbitrary robots. For instance, our result is directly applicable to the case of two planar robot arms studied by Fortune et al. Furthermore, we not only prove the existence of a certain restricted type of multi-path, we also present a general algorithm for finding such a path efficiently. Unfortunately, we can only prove our result for two and three robots, so in this respect our results are less general than those of Hopcroft and Wilfong. More precisely, our results are as follows.

In Section 3 we prove that in order to plan the motions of two robots R_1 and R_2, one does not have to consider the entire $(d_1 + d_2)$-dimensional configuration space $\mathcal{CS}_1 \times \mathcal{CS}_2$. Instead, it is sufficient to consider a collection of five suitably linked configuration-space slices (corresponding to the constrained type of configurations mentioned above) whose dimensions are at most $d_1 + d_2 - 1$. Combining this with the method of Basu et al. we obtain a general method to solve the motion-planning problem for a pair of robots in $O(n^{d_1+d_2})$ time.

In Section 4 we extend our ideas to triples of robots. Here we show that it is sufficient to consider a constant number of configuration-space slices of maximum dimension $d_1 + d_2 + d_3 - 2$, which leads to an $O(n^{d_1+d_2+d_3-1})$ time algorithm. Note that one can view a collection of four or more robots as a collection of three robots, one of which is a multi-robot consisting of $m - 2$ robots. Hence, the improvement (as compared to the standard method of viewing the collection of robots as one multi-robot) of two orders of magnitude that we obtain for three robots carries over to the case of four or more robots.

Our approach becomes especially effective when the robots under consideration have bounded reach and the workspace has low density [22], as we show in Section 5. Bounded-reach robots are robots that are not too large compared to the obstacles, and a workspace has low density if any region is intersected by only a constant number of obstacles that are larger than that region. These notions were introduced to exclude unrealistic inputs—contrived workspaces and robots that induce very high free-space complexities. It is expected that in most practical situations the robots have bounded reach and the workspace they operate in has low density. Van der Stappen et al. [22] demonstrate that the complexity of the free space of a single bounded-reach robot in a low-density workspace is $O(n)$, irrespective of the number of degrees of freedom of the robot and of the dimension of its workspace. Van der Stappen et al. also give an algorithm to compute a linear-size cell decomposition of the free space, leading to an $O(n \log n)$ time algorithm to plan the motion of a single robot. Unfortunately, if we consider a pair or triple of bounded-reach robots operating in a low-density workspace, then the free-space complexity of the corresponding multi-robot can be as high as $\Theta(n^2)$ for a pair and $\Theta(n^3)$ for a triple. Nevertheless, we show how to apply our method to this case to obtain the following surprising result: for bounded-reach robots in low-density workspaces one can solve the motion-planning problem for pairs or triples of robots in the same time as for a single robot, namely in $O(n \log n)$ time. Note that the method of Sharir and Sifrony [19] can be used in this case, because a cell decomposition of the free space of the individual robots is available. This, however, leads to algorithms with considerably higher running times, namely $O(n^2)$ for a pair of robots and $O(n^3)$ for a triple.

2 Preliminaries

Although we assume some familiarity with the basic concepts in motion planning, we briefly introduce the terminology and notation used throughout the paper. A general introduction to motion planning can be found in Latombe's book [13].

Let $\mathcal{R} = \{R_1, \ldots, R_m\}$ denote a collection of m robots. All robots operate in the same workspace $\mathcal{W}$, which contains a set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of obstacles. We assume that each robot
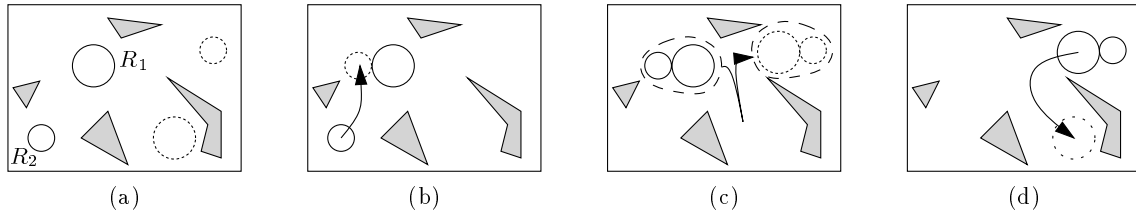
Figure 1: A feasible multi-path using permissible multi-configurations.

and each obstacle has constant complexity, and that they are semi-algebraically defined, that is, bounded by surfaces of low algebraic degree. We also assume they are open sets; this means that the robot is allowed to 'slide along' an obstacle.

The *configuration space* of robot R_i is denoted by $\mathcal{CS}_i$. The dimension of $\mathcal{CS}_i$ equals d_i, the number of degrees of freedom of R_i. We assume that $d_i > 0$ for every i. Points in $\mathcal{CS}_i$—or, *configurations*—correspond to placements of R_i in the workspace; we denote the portion of the workspace occupied by R_i at configuration $p \in \mathcal{CS}_i$ by $R_i[p]$. The points in $\mathcal{CS}_i$ representing the start and goal configuration of R_i are denoted by s_i and g_i respectively. A path for R_i from its start configuration to its goal configuration corresponds to a curve in $\mathcal{CS}_i$ from s_i to g_i. We parameterize the curve by a time parameter t, with $t \in [0,1]$, so a path from start to goal configuration is a continuous mapping $\pi_i : [0,1] \to \mathcal{CS}_i$ with $\pi_i(0) = s_i$ and $\pi_i(1) = g_i$.

Each obstacle defines a subset—its *configuration-space obstacle*—in a configuration space consisting of all configurations in which the robot intersects that obstacle. The portion of the configuration space covered by the configuration-space obstacles is called the *forbidden space*, and its complement is called the *free space*. We call a path π_i for R_i *feasible* if R_i does not intersect any obstacle during its motion or, in other words, if the curve π_i lies entirely in the free space.

As stated in the introduction, we can view the collection of robots as one composite robot, or *multi-robot*, with $d := \sum_{i=1}^{m} d_i$ degrees of freedom. We refer to configurations of the multi-robot as *multi-configurations*, and we call a path for the multi-robot (which is in fact a collection of paths for the individual robots) a *multi-path*. We want to find a *feasible multi-path* for the robots $R_1, \ldots, R_m$ and their given start and goal configurations, that is, a collection of paths that brings each robot from its start configuration to its goal configuration without colliding with either the obstacles or the other robots.

3 Pairs of robots

To explain the idea of our method, we start by studying the case of a pair of robots. One way of planning the motion of a pair of robots is to view the pair as one robot with $d := d_1 + d_2$ degrees of freedom. Thus the problem can be solved in the d-dimensional configuration space $\mathcal{CS}_1 \times \mathcal{CS}_2$. Our goal is to reduce the dimension of the space we have to consider. To this end we limit the possible multi-configurations—combinations of configurations for the two robots—that we allow. Of course, we have to guarantee that a feasible multi-path continues to exist.

The multi-configurations that we allow—we call them *permissible multi-configurations*—are as follows.

- When R_1 is at its start or goal configuration, we allow any configuration of R_2.

- When R_2 is at its start or goal configuration, we allow any configuration of R_1.

- When neither R_1 nor R_2 is at its start or goal configuration, we only allow configurations where R_1 and R_2 touch each other.

We give an example. Consider the situation depicted in Fig. 1, where we have two disk-shaped robots moving amidst polygonal obstacles in the plane. The start and goal configurations of the

robots are indicated in Fig. 1(a); start configurations are solid and goal configurations are dotted. A feasible multi-path for this problem that uses permissible multi-configurations is indicated in Fig. 1(b)–(d): first R_2 moves towards R_1 until it touches it, then R_2 and R_1 together move until R_2 is at its end configuration, and finally R_1 breaks off its contact with R_2 and moves to its own goal configuration.

At first sight, it seems quite severe to restrict oneself to permissible multi-configurations. Nevertheless, it turns out that solutions using only permissible multi-configurations always exist, provided a solution exists at all.

Lemma 1 *Let R_1 and R_2 be two robots operating in the same workspace. If there is a feasible multi-path for given start and goal configurations, there is also a feasible multi-path for those start and goal configurations that only uses permissible multi-configurations.*

Proof: Let $\Pi = \{\pi_1, \pi_2\}$ be a feasible multi-path. We define the *coordination diagram* for Π as follows. Let U be the unit square. We call the edges of U incident to the origin the *axes* of U. The horizontal axis, or t_1-*axis*, of U represents the configuration of R_1 along π_1; the vertical axis, or t_2-*axis*, represents the configuration of R_2 along π_2. Thus a point $(t_1, t_2) \in U$ corresponds to placing R_1 and R_2 at configurations $\pi_1(t_1)$ and $\pi_2(t_2)$ respectively. Observe that the left edge of U corresponds to multi-configurations where R_1 is at its start configuration, the top edge of U corresponds to configurations where R_2 is at its goal configuration, and so on. A point $(t_1, t_2) \in U$ is called *forbidden* if $R_1[\pi_1(t_1)]$ intersects $R_2[\pi_2(t_2)]$; otherwise it is called *free*—see Fig. 2. The coordination diagram for Π is the subdivision of U into free and forbidden regions.
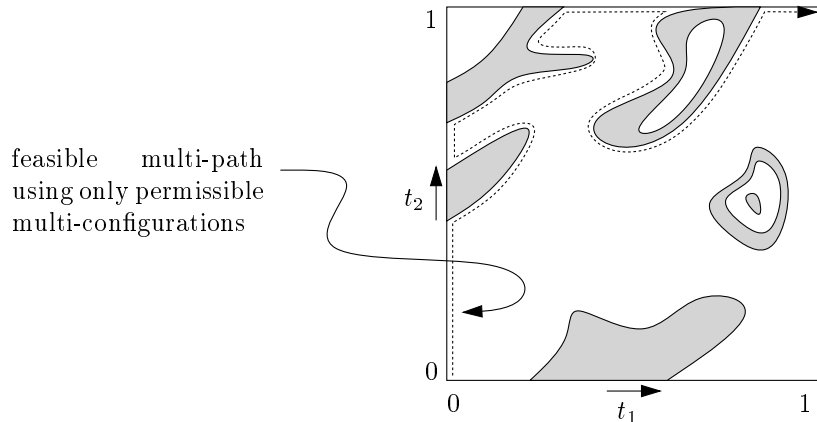

Figure 2: The coordination diagram; forbidden regions are shaded.

Define $\mathbf{0} := (0, 0)$ and $\mathbf{1} := (1, 1)$ to be the lower left and top right vertex of U, respectively. We call a path in U from $\mathbf{0}$ to $\mathbf{1}$ a $\mathbf{0}$-$\mathbf{1}$ *path*. Since $\{\pi_1, \pi_2\}$ is a feasible multi-path, R_1 does not intersect any obstacle along π_1 and R_1 does not intersect any obstacle along π_2. Hence, a $\mathbf{0}$-$\mathbf{1}$ path that lies in the free region corresponds to a feasible multi-path; we call this a *feasible* $\mathbf{0}$-$\mathbf{1}$ *path*. Notice that the diagonal from $\mathbf{0}$ to $\mathbf{1}$ is by definition a feasible $\mathbf{0}$-$\mathbf{1}$ path. This means that $\mathbf{0}$ and $\mathbf{1}$ lie in the same component of the free region. Since they both lie on the boundary of U, they must lie in the same component of the boundary of the free region as well. Hence, there is a feasible $\mathbf{0}$-$\mathbf{1}$ path along the boundary of the free region, as illustrated in Fig. 2. Any point on such a $\mathbf{0}$-$\mathbf{1}$ path corresponds to a permissible multi-configuration: the point either lies on the boundary of U, in which case one of the robots is at its start or goal configuration, or it lies on the boundary of a forbidden region, in which case the robots touch each other. $\square$

Before we continue we make two remarks about the proof. First, the $\mathbf{0}$-$\mathbf{1}$ path that we find is not necessarily monotone in either t_1 or t_2. This may seem like a problem, because it appears to mean that we move back in time. But what it really means it that we move back along the path π_1

or π_2, which is allowed. Second, it is important to realize that we do not have a feasible multi-path available at the start of the algorithm—otherwise we would already be done—so we cannot compute the coordination diagram used in the proof. But we can use it to prove the correctness of our approach.

We now know that we can solve the motion-planning problem by looking at only a subspace of the composite configuration space $\mathcal{CS}_1 \times \mathcal{CS}_2$. This subspace consists of five *configuration-space slices*, or *slices* for short.

- In the first slice, R_2 is free to move and R_1 is stationary at its start configuration; here we can simply consider R_1 as an additional obstacle. We denote this configuration-space slice by $\mathcal{CS}_{1,s}$; its dimension is d_2, the number of degrees of freedom of R_2.

- In $\mathcal{CS}_{1,g}$, the second slice, R_2 is again free to move and R_1 is stationary, but this time R_1 is at its goal configuration. Again, R_1 is an additional obstacle, and the dimension of the slice is d_2.

- The third slice, $\mathcal{CS}_{2,s}$, is defined analogously to $\mathcal{CS}_{1,s}$, with the roles of R_1 and R_2 reversed. Its dimension is d_1.

- The fourth slice, $\mathcal{CS}_{2,g}$, is defined analogously to $\mathcal{CS}_{1,g}$, with the roles of R_1 and R_2 reversed. Its dimension is d_1.

- The fifth slice, $\mathcal{CS}_{\text{contact}}$, is a configuration space for the *contact robot* $\langle R_1 R_2 \rangle$, which is the robot composed of R_1 and R_2 where R_1 and R_2 are required to touch each other.

In the sequel we make the following assumption:

DOF-Reduction Assumption (for pairs of robots): The number of degrees of freedom of the contact robot $\langle R_1 R_2 \rangle$ is at most $d_1 + d_2 - 1$, where d_1 and d_2 are the numbers of degrees of freedom of the robots R_1 and R_2, respectively.

There are certain degenerate situations where this condition is not fulfilled. For instance, when two blocks are confined to move on parallel tracks whose distance is such that the blocks touch when they pass each other, then requiring them to touch will not reduce the total number of degrees of freedom.

Linking the configuration-space slices. Of course we cannot treat the five slices completely separately; a feasible multi-path using only permissible multi-configurations will in general switch between slices a number of times. In the example of Fig. 1, for instance, the first part of the path lies in $\mathcal{CS}_{1,s}$, then (when R_2 reaches R_1) a switch is made to $\mathcal{CS}_{\text{contact}}$, and finally a switch is made to $\mathcal{CS}_{2,g}$. We have to connect the slices to make such switches possible. We do this by identifying certain *transition points* in each slice. These points correspond to configurations that are represented by a point in one of the other slices as well. For a given transition point in a slice, we call the point in another slice that corresponds to the same configuration its *twin* in the other slice. Thus if we travel along a curve in one slice and reach a transition point, we may continue in the other slice from its twin. Next we explain which transition points we use.

First we observe that no switches can occur between $\mathcal{CS}_{1,s}$ and $\mathcal{CS}_{1,g}$, because R_1 cannot go from its start to its goal configuration instantaneously. (If $s_1 = g_1$ then $\mathcal{CS}_{1,s} = \mathcal{CS}_{1,g}$, so we can discard one of these configuration-space slices.) Similarly, no switches can occur between $\mathcal{CS}_{2,s}$ and $\mathcal{CS}_{2,g}$.

There is only one point in $\mathcal{CS}_{1,s}$ where we can step to $\mathcal{CS}_{2,s}$, namely at s_2. Its twin in $\mathcal{CS}_{2,s}$ is s_1. This transition is not needed, however, because there is no reason to ever come back to the initial configuration and make a switch there. Similarly, we need not add transition points to step from $\mathcal{CS}_{1,g}$ to $\mathcal{CS}_{2,g}$, because when we arrive at that point we have solved the problem.

We do need to add transition points from $\mathcal{CS}_{1,s}$ to $\mathcal{CS}_{2,g}$, namely the point $g_2 \in \mathcal{CS}_{1,s}$ and its twin $s_1 \in \mathcal{CS}_{2,g}$. Similarly, we need to add $s_2 \in \mathcal{CS}_{1,g}$ and its twin $g_1 \in \mathcal{CS}_{2,s}$ to the collection of transition points.

The difficulty lies in defining transition points to step from $\mathcal{CS}_{\text{contact}}$ to one of the other slices. The problem is that in general there is a continuum of configurations common to $\mathcal{CS}_{\text{contact}}$ and, say, $\mathcal{CS}_{1,s}$. We want to add only a limited number of transition points, while at the same time ensuring that no essential connectivity is lost. To achieve this we add the following transition points.

The free part of the slice $\mathcal{CS}_{1,s}$ consists of a number of *cells* (d_2-dimensional features), which are bounded by parts of constraint hypersurfaces. We call the $(d_2 - 1)$-dimensional features of a cell the *patches* of the cell. A patch is a part of the boundary of some configuration-space obstacle (recall that R_1 is now regarded as an additional obstacle); patches are by definition path-connected. Each patch corresponds to a maximal path-connected set of configurations where R_2, the robot which is free to move in the slice we are considering, either touches a specific obstacle or R_1. For each patch on the boundary of a cell in the free space that corresponds to configurations where R_1 and R_2 touch, we take an arbitrary point on the patch as a transition point (together with its twin in $\mathcal{CS}_{\text{contact}}$).

The following lemma shows that the transition points we defined are sufficient to capture the connectivity between the slices.

Lemma 2 *If there is a feasible multi-path for given start and goal configurations for R_1 and R_2, then there is a feasible multi-path whose corresponding curve in the configuration space $\mathcal{CS}_1 \times \mathcal{CS}_2$ lies entirely in the union of the five slices defined above and switches between slices at transition points.*

Proof: Let s_1, g_1, s_2, and g_2 be start and goal configurations such that there is a feasible multi-path for R_1 and R_2. Lemma 1 states that there is a feasible multi-path that uses permissible multi-configurations only, that is, lies in the union of the five slices. Consider such a multi-path $\Pi = \{\pi_1, \pi_2\}$. It is possible that Π switches between configuration-space slices at points other than transition points. We modify Π so that it only switches between slices at transition points.

Switches that are not at transition points can only occur between $\mathcal{CS}_{\text{contact}}$ and one of the other four configuration-space slices. Assume without loss of generality that Π switches to $\mathcal{CS}_{1,s}$. By definition, the switch must occur at a configuration where R_1 and R_2 touch and, moreover, R_1 is at its start configuration. Hence, this configuration is represented by a point p on a patch of some free-space cell in $\mathcal{CS}_{1,s}$. Let q be the transition point chosen for that patch. Furthermore, let p' and q' be the twins in $\mathcal{CS}_{\text{contact}}$ of p and q. Because p and q are on the same patch, there is a curve on that patch connecting them. Such a curve corresponds to a motion of R_2 that keeps it in contact with R_1. Hence, this motion is also represented by some curve connecting p' to q' in $\mathcal{CS}_{\text{contact}}$. This means that instead of stepping from $\mathcal{CS}_{\text{contact}}$ to $\mathcal{CS}_{1,s}$ at the point p', we can first follow the curve from p' to q' in $\mathcal{CS}_{\text{contact}}$, then follow the link between q' and q, and finally move back from q to p. This proves that the transition points provide all the necessary connectivity. □

The algorithm. We combine the ideas above with an algorithm of Basu et al. [2]. This algorithm computes a roadmap of a given semi-algebraic set—of the free configuration space in a motion-planning problem, for instance. A *roadmap* is a one-dimensional subspace—a graph embedded in $\mathcal{CS}$—that captures the connectivity of the free configuration space. If the number of obstacles is n and the dimension of the configuration space is d, then the algorithm constructs the roadmap in $O(n^{d+1})$ time. The algorithm by Basu et al. allows to connect a given point in the free space to the part of the roadmap that lies in the same connected component of the free space in $O(n)$ time. We will use this feature of the algorithm to include the transition points in the roadmap. In both time bounds, the constant of proportionality depends on the algebraic degree of the constraints.

Now consider the setting where we have two robots, R_1 and R_2. We use the method of Basu et al. to construct a roadmap in each of the five configuration-space slices defined above. Since the five configuration-space slices have dimensions at most $d_1 + d_2 - 1$, we can compute these roadmaps in time $O(n^{d_1+d_2})$. It remains to link the roadmaps in the five configuration-space slices. To this end, we first include in the roadmap of each configuration-space slice the set of

transition points defined for that slice, taking $O(n)$ time per point. (We explain below how to find the transition points.) This gives us five graphs, each capturing the connectivity of one of the free spaces. Finally, we add links between transition points and their twins. We thus obtain one graph, which captures the connectivity of the free space in $\mathcal{CS}_1 \times \mathcal{CS}_2$.

The computation of the transition points is only interesting for transition points between $\mathcal{CS}_{\text{contact}}$ and the other configuration-space slices. We focus on the the transition points between $\mathcal{CS}_{\text{contact}}$ and $\mathcal{CS}_{1,s}$. Consider the surface of the configuration-space obstacle in $\mathcal{CS}_{1,s}$ corresponding to R_2 touching R_1. This surface is an algebraic variety of dimension $d_2 - 1$. Consider the arrangement induced on it by all the configuration-space obstacles in $\mathcal{CS}_{1,s}$. We place a transition point in those faces of this arrangement that correspond to free configurations of the contact robot. This is done by invoking the algorithm of Basu et al. [3], which yields in $O(n^{d_2})$ time a representative point in each face (of any dimension) of the arrangement. For each of the resulting $O(n^{d_2-1})$ representative points we determine in a brute-force manner whether the point is in the free space; the points that are in the free space are added to the set of transition points. The time needed for this is $O(n^{d_2})$. Hence, the total time to compute the transition points between $\mathcal{CS}_{\text{contact}}$ and $\mathcal{CS}_{1,s}$ is $O(n^{d_2})$.

It follows from the above discussion that the total time to compute all the transition points is $O(n^{\max(d_1,d_2)})$. The connection of all these $O(n^{\max(d_1,d_2)-1})$ transition points to their respective roadmaps takes $O(n^{\max(d_1,d_2)})$ time. As a result, the computation of the final graph, capturing the connectivity of the free space in $\mathcal{CS}_1 \times \mathcal{CS}_2$, takes $O(n^{d_1+d_2})$ time. The graph search needed to finally solve the motion-planning problem takes linear time in the size of the roadmap. As a result, the motion planning problem for a pair of robots can be solved in $O(n^{d_1+d_2})$ time. This leads to the following theorem.

Theorem 3 *Let R_1 and R_2 be two robots, with d_1 and d_2 degrees of freedom respectively, that satisfy the DOF-Reduction Assumption, operating in a workspace with n obstacles. Then we can compute a feasible multi-path for a given pair of start and goal configurations for R_1 and R_2 in $O(n^{d_1+d_2})$ time, if it exists, and otherwise report failure.*

4 Three robots

To extend the results of the previous section to the case of three robots, we have to find a suitable definition of permissible multi-configurations. We first generalize the notion of a coordination diagram, a concept from the proof of Lemma 1. We then prove the existence of a certain type of **0-1** path in the coordination diagram—such a curve represents a feasible motion—from which the definition of permissible multi-configurations follows. As in the previous section, the permissible multi-configurations induce a constant number of slices of the configuration space for the multi-robot; the motion-planning problem can then be solved in the union of these slices, suitably linked.

Let $\mathcal{R} = \{R_1, R_2, R_3\}$ denote a collection of three robots operating in a workspace $\mathcal{W}$, which contains a set $\mathcal{C} = \{C_1, \ldots, C_n\}$ of obstacles. As in Section 3, we assume that robots are open and semi-algebraically defined. Suppose that a feasible multi-path exists between the given start and goal configurations of the robots. In addition, we make a stronger assumption:

Positive-Clearance Assumption: There is a feasible **0-1** path that stays in the interior of the free configuration space of the multi-robot, except at the endpoints.

We suspect that this assumption is not necessary, but we face some technical difficulties if we drop it.

Let $\Pi = \{\pi_1, \pi_2, \pi_3\}$ denote a feasible multi-path guaranteed by the positive-clearance assumption. As it lies in the interior of the free configuration space, which is a semi-algebraic set, we can assume Π is semi-algebraic; connected semi-algebraic sets are semi-algebraically connected (see for example Schwartz and Sharir [16]). The coordination diagram for Π is defined as follows. Let U denote the three-dimensional unit cube $[0,1]^3$. The edges of U incident to the origin are called

the *axes* of U. Each axis represents the configuration of one of the robots; the t_1-axis represents the configuration of R_1 along π_1, the t_2-axis represents the configuration of R_2 along π_2, and the t_3-axis represents the configuration of R_3 along π_3. Thus a point $(t_1, t_2, t_3) \in U$ corresponds to placing R_i at $\pi_i(t_i)$ for every $1 \leqslant i \leqslant 3$.

A pair of values $(\alpha, \beta) \in [0, 1]^2$ is called *ij-forbidden*, for some $i \neq j$, if $R_i[\pi_i(\alpha)]$ intersects $R_j[\pi_j(\beta)]$. For indices i, j, k with $\{i, j, k\} = \{1, 2, 3\}$, define $U_{ij} := \{(t_1, t_2, t_3) | t_i, t_j \in [0, 1], t_k \in \mathbb{R}\}$. Define B_{ij} to be the set of all points $(t_1, t_2, t_3) \in U_{ij}$ so that (t_i, t_j) is *ij*-forbidden. We call B_{ij} a *coordination-diagram obstacle*, or *cd-obstacle* for short. (We do not constrain the remaining coordinate to lie in the interval $[0, 1]$ for technical reasons.) Since the robots and workspace obstacles are open and the robot position is a continuous function of t, cd-obstacles are open. Each obstacle is a cylinder that is the Cartesian product of the forbidden region on some $t_i t_j$-face of U with the remaining axis. (The $t_i t_j$-*face* of U is the 2-dimensional face of U spanned by the t_i- and t_j-axes.) Note that cd-obstacles have nothing to do with the obstacles in the workspace; they are defined using Π, the positive-clearance feasible multi-path, and reflect possible interferences between robots if they follow the paths of Π independently.

A point $(t_1, t_2, t_3) \in U$ is called *forbidden* if there is a pair i, j of distinct indices such that (t_i, t_j) is *ij*-forbidden; otherwise it is called *free*. In other words, a free point corresponds to a placement of each robot at some point along its path so that the robots do not overlap. The *coordination diagram* for Π is the subdivision of U into free and forbidden regions. Fig. 3 shows schematically a coordination diagram for three disk-shaped robots in the plane. By definition, the
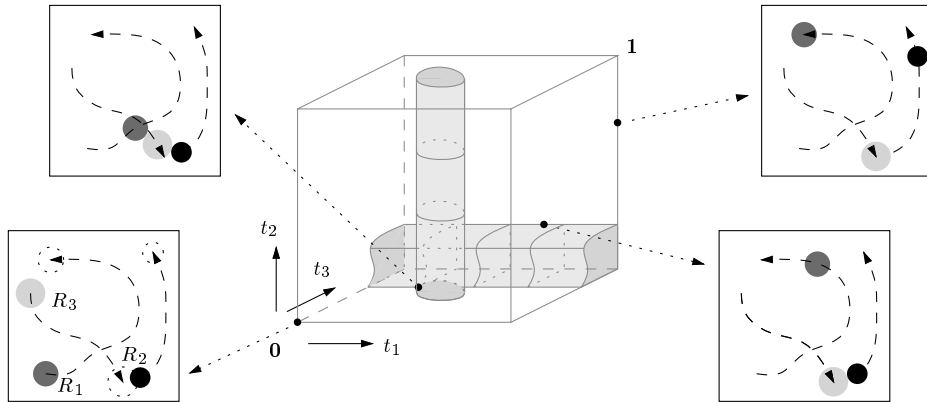

Figure 3: The coordination diagram for three robots.

forbidden region is the union of the cd-obstacles truncated to within U.

Define $\mathbf{0} := (0, 0, 0)$ and $\mathbf{1} := (1, 1, 1)$ to be the lexicographically smallest and largest vertex of U, respectively, and let Δ be the diagonal connecting $\mathbf{0}$ to $\mathbf{1}$.

We will now slightly modify the notion of a cd-obstacle and free point. Namely, positive clearance implies that Δ in fact does not meet the *closure* of the cd-obstacles, except at its endpoints. Consider a specific cd-obstacle B_{ij}. Take its closure $\mathrm{Cl}(B_{ij})$. If $\mathrm{Cl}(B_{ij})$ is not simply connected, Δ cannot meet any of its holes, because Δ avoids $\mathrm{Cl}(B_{ij})$ except at its endpoints. (There is one case in which this is not true, namely if $\mathrm{Cl}(B_{ij})$ has a hole containing Δ. This can only happen if $\mathrm{Cl}(B_{ij})$ touches Δ at $\mathbf{0}$ and at $\mathbf{1}$; we then consider $\mathrm{Cl}(B_{ij})$ to be two seperate obstacles—one 'above' and one 'below' Δ—that meet along the two edges of the cube incident to $\mathbf{0}$ and $\mathbf{1}$, respectively, and perpendicular to the $t_i t_j$-facet. The volume in between these two obstacles is not considered a hole. It is easy to verify that this does not cause any problems in the remainder of the proof.) So we can safely "fill in" the holes if there are any. The resulting set $\bar{B}_{ij}$ is a *closed cd-obstacle* and its interior $B_{ij} := \mathrm{Int}(\bar{B}_{ij})$ relative to U_{ij} is a *(modified open) cd-obstacle*. In the sequel, we refer to it simply as a cd-obstacle.

We write $\partial \bar{B}_{ij}$ for the boundary of $\bar{B}_{ij}$ relative to U_{ij}. Note that each connected component of $\bar{B}_{ij}$ is contractible by construction, since we removed all holes from it. Put $\mathcal{B} := \{\bar{B}_{12}, \bar{B}_{13}, \bar{B}_{23}\}$.

The *free region* $\mathcal{FR}$ is defined as
$$\mathcal{FR} := U \smallsetminus \bigcup_{i \neq j} B_{ij}.$$

A **0-1** *path* is a curve in U from **0** to **1**. We call a **0-1** path *feasible* if it lies in $\mathcal{FR}$. By construction, Δ is feasible.

We want to prove the existence of a feasible **0-1** path along $\partial \mathcal{FR}$, because this will restrict the configurations for the collection of robots. In fact, we want to find a **0-1** path along the so-called *skeleton* of $\mathcal{FR}$, which intuitively is a graph-like structure embedded in $\partial \mathcal{FR}$. We give a formal definition below.

In the remainder of this section we will assume that all subsets of the configuration diagram that we consider are semi-algebraic. Indeed, since the robots and obstacles are semi-algebraically defined, and Π is semi-algebraic, B_{ij}, $\bar{B}_{ij}$, $\partial \bar{B}_{ij}$ are semi-algebraic.

Let $h_1, \ldots, h_6$ be the six planes supporting facets of U. For $1 \leqslant i \leqslant 6$, let H_i be the open halfspace bounded by h_i and not containing U. Put $\mathcal{H} := \{H_1, \ldots, H_6\}$.

Let $\mathcal{G}$ be the following set of "surfaces": h_i, for $1 \leqslant i \leqslant 6$, and for each connected component of each $\bar{B}_{ij} \in \mathcal{B}$, its boundary relative to U_{ij}. Note that the latter "surfaces" need not be manifolds; for instance, their cross section can look like a figure eight. We define the *1-skeleton* (or simply *skeleton*) S of $\mathcal{FR}$ as follows:
$$S := \mathcal{FR} \cap \bigcup \left\{ \sigma \cap \sigma' \ : \ \sigma, \sigma' \in \mathcal{G} \right\}.$$

If the surfaces of $\mathcal{G}$ are manifolds and intersect among themselves transversally, then $\partial \mathcal{FR}$ is a manifold bounded by 2-faces ("facets")—connected portions of boundary of sets from $\mathcal{B} \cup \mathcal{H}$—, 1-faces ("edges")—portions of pairwise intersections of these surfaces—, and 0-faces ("vertices")— points of triple intersections. In this case S is the traditional 1-skeleton of $\partial \mathcal{FR}$—the union of edges and vertices of $\mathcal{FR}$. In more general situations, $\partial \mathcal{FR}$ need not be a manifold. Moreover, S need not be a variety of dimension 1, since it may contain 2-dimensional portions.

We want to prove that there is a **0-1** path along the skeleton S or, in other words, that **0** and **1** are connected by a path in S.

Proposition 4 *Let $\mathcal{B}$ be a finite set of closed simply-connected cylinders as above in the three-dimensional unit cube U such that the **0-1** diagonal Δ lies in the complement $U \smallsetminus \bigcup \mathcal{B}$ of the cylinders. Then there is a **0-1** path along the skeleton S of $\mathcal{FR}$.*

We first sketch the line of attack. We argue by contradiction. If there is no **0-1** path along the skeleton, there must be a collection of curves (in the proof we actually use more general sets) on $\partial \mathcal{FR}$ that avoids S and separates the surface, so that **0** and **1** lie in different components when the curves are removed. Then we view $\mathcal{FR}$ and its complement as two topological spaces glued together along the boundary of $\mathcal{FR}$ *except for these curves*. Since the boundary of $\mathcal{FR}$ is separated by the curves, a path connecting the two points outside the cube and another one (namely, Δ) connecting them in $\mathcal{FR}$ together form a loop that cannot be contracted to a point in the topological space obtained by removal of these curves from $\mathbb{R}^3$. We continue by proving that the curves are not "tangled" in the sense that each of them can be contracted to a point in a cd-obstacle without meeting the other curves or the loop, and then removed. Thus an incremental process removes all curves, yielding $\mathbb{R}^3$. We argue that the non-contractibility of the loop we constructed above is not affected by this process—it should remain non-contractible. We then have a contradiction, since $\mathbb{R}^3$, being simply connected, admits no non-contractible loops.

We now give a more formal version of this argument. We will need the following topological facts. Recall that we have assumed that all sets we consider are semi-algebraic—which, for a compact set, is a stronger assumption than assuming it corresponds to a finite CW-complex. In the sequel we will not distinguish between a complex and the corresponding topological space.

Fact 5 *Let V and W be finite CW-complexes with V path-connected and $V \cup W$ contractible to a point. Then the number of path-connected components of $V \cap W$ equals the number of path-connected components of W.*

This is easily shown by considering the Mayer-Vietoris sequence. For a complete argument, see Theorem 3.9 of [11].

Fact 6 *Let D be (a topological space homeomorphic to) a closed two-dimensional disk. Let V, W be two finite CW-complexes and $x, y \in \partial D$ be distinct points such that:*

- *$V \cup W = D$,*

- *$x, y \in V \cap W$, and*

- *x and y divide ∂D into two curves α and β, with $\alpha \subset V$ and $\beta \subset W$.*

Then x and y are connected by a path in $V \cap W$.

Proof: In this proof, we write "component" to mean "path-connected component." By replacing V with its component containing α and adjoining the rest of V to W, we maintain the assumptions of the lemma while not increasing $V \cap W$. So in the remainder of the proof we assume that V is path-connected.

Consider a component X of W. In finite CW-complexes the notion of path-connectedness coincides with the notion of connectedness, so X, being a connected component of a closed set must be a closed subset of W and thus of D. If $X \cap V = \varnothing$, then X and $V \cup (W \smallsetminus X)$ are two disjoint closed sets whose union is D, contradicting connectedness of D. Hence X must intersect V. Thus each component of W meets $V \cap W$. Trivially a component of $V \cap W$ is contained in a single component of W. By Fact 5 the number of components of $V \cap W$ is equal to the number of components of W. Hence each component of W contains exactly one component of $V \cap W$. As x and y lie on β, and thus in the same component of W, and also in $V \cap W$, they must lie in the same component of $V \cap W$, completing the proof. □

Given a finite family $\mathcal{F}$ of disjoint, compact, not necessarily connected semi-algebraic sets in $\mathbb{R}^3$, we say that a set $V \in \mathcal{F}$ is *trivial (with respect to $\mathcal{F}$)* if there exists a contractible compact set W containing V and avoiding the rest of the sets of the family. Intuitively, this means that V is contractible in the complement of $\bigcup \mathcal{F} \smallsetminus \{V\}$.

Fact 7 *Consider a family $\mathcal{F}$ as above containing, among other sets, a loop C. If all sets besides C are trivial with respect to $\mathcal{F}$, then C is also trivial.*

Proof: We proceed by induction on the size of the family. If it consists of only C the claim trivially holds.

Now suppose that the claim holds for families of size $k > 0$, and consider a family $\mathcal{F}$ with $k + 1$ sets. All sets besides $C \in \mathcal{F}$ are trivial. Let $V \in \mathcal{F}$ be different from C. By inductive assumption, C is trivial with respect to $\mathcal{F} \smallsetminus \{V\}$. For a contradiction, suppose that C is *not* trivial with respect to $\mathcal{F}$. By assumption, V is trivial, so there is a compact contractible set $W \supset V$ avoiding the rest of the sets of $\mathcal{F}$. Consider the collection $(\mathcal{F} \smallsetminus \{V, C\}) \cup \{W\}$. Using the techniques of Milnor [14], one can construct an open neighborhood of each set of this collection such that the neighborhoods of different sets are disjoint and avoid C and each set is a strong deformation retract of its neighborhood and of the closure of its neighborhood. Let $\widehat{\mathcal{F}}$ be the resulting collection of neighborhoods. Non-triviality of C with respect to $\mathcal{F}$ implies that it cannot be contracted to a point in $\mathbb{R}^3 \smallsetminus \bigcup \widehat{\mathcal{F}}$, as this set is smaller than $\mathbb{R}^3 \smallsetminus \bigcup(\mathcal{F} \smallsetminus \{C\})$. Because the neighborhoods can be made arbitrarily close to the sets they surround, C is contractible in $\mathbb{R}^3 \smallsetminus \bigcup(\widehat{\mathcal{F}} \smallsetminus \{\widehat{W}\})$, where $\widehat{W}$ is the neighborhood of W. Hence $H_1(\mathbb{R}^3 \smallsetminus \bigcup \widehat{\mathcal{F}})$ cannot be isomorphic to $H_1(\mathbb{R}^3 \smallsetminus \bigcup(\widehat{\mathcal{F}} \smallsetminus \{\widehat{W}\}))$; here H_1 denotes the first singular homology group. Using Alexander Duality this is equivalent to $H_1(\mathrm{Cl}(\bigcup \widehat{\mathcal{F}}))$ being non-isomorphic to $H_1(\mathrm{Cl}(\bigcup(\widehat{\mathcal{F}} \smallsetminus \{\widehat{W}\})))$. However, $\mathrm{Cl}(\bigcup \widehat{\mathcal{F}})$ consists of several connected components, and $\mathrm{Cl}(\bigcup(\widehat{\mathcal{F}} \smallsetminus \{\widehat{W}\}))$ consists of the same connected components plus the closure of $\widehat{W}$, which can be retracted to W, which is contractible and thus has trivial first singular homology group, contradicting the claim and proving the fact. □

Proof of Proposition 4: Put $\mathcal{FR}' := \mathrm{Cl}(\mathbb{R}^3 \smallsetminus \mathcal{FR})$. Since $\mathbf{0}, \mathbf{1}$ lie on ∂U, there exists a path σ connecting them in $\mathrm{Cl}(\mathbb{R}^3 \smallsetminus U) \subset \mathcal{FR}'$. Later it will be convenient to assume that σ lies outside U except at its endpoints. On the other hand, Δ lies in $\mathrm{Int}(\mathcal{FR})$ except at $\mathbf{0}$ and $\mathbf{1}$. Let η be the loop formed by joining σ and Δ.

For a contradiction, suppose that $\mathbf{0}, \mathbf{1}$ do not lie in the same path-connected component of S. Let S_1 be the component of S containing $\mathbf{0}$ and S_2 be the rest of S. Consider the set Σ of all points in $\partial \mathcal{FR}$ at equal distance from S_1 and S_2 in the Euclidean metric. It is compact. In absence of degeneracies it would be a one-dimensional variety, but in general it may contain two-dimensional portions.

Suppose we were to continuously deform σ to Δ in $\mathbb{R}^3$; this is possible since $\mathbb{R}^3$ is contractible. This deformation can be viewed as a continuous function γ mapping a two-dimensional disk D into $\mathbb{R}^3 = \mathcal{FR} \cup \mathcal{FR}'$ and ∂D onto η. Putting $V := \gamma^{-1}(\mathcal{FR})$ and $W := \gamma^{-1}(\mathcal{FR}')$, we notice that $V \cup W = D$, $x := \gamma^{-1}(\mathbf{0})$ and $y := \gamma^{-1}(\mathbf{1})$ both lie in $V \cap W$, and the paths Δ and σ connecting $\mathbf{0}$ to $\mathbf{1}$ correspond to the two portions of ∂D connecting x to y in V and W, respectively. Applying Fact 6, we conclude that there is a path δ from x to y in $V \cap W$, and hence a path $\gamma(\delta)$ from $\mathbf{0} = \gamma(x)$ to $\mathbf{1} = \gamma(y)$ in $\gamma(V \cap W) = \mathcal{FR} \cap \mathcal{FR}' = \partial \mathcal{FR}$. As the difference of the Euclidean distances from a point on this path to S_1 and to S_2 is a continuous function of position of the point along the path, and since it is negative at $\mathbf{0}$ and positive at $\mathbf{1}$, there is a point $z \in \Sigma$ on the path. In other words, we have shown that during any continuous deformation of σ to Δ there is an intermediate path that meets the bisector Σ.[1] In other words, in any family of sets that includes η and Σ, η is not trivial.

Now consider a path-connected component of Σ. We know that Σ lies in $\partial \mathcal{FR} \subset \bigcup \mathcal{G}$ and avoids S, which is the union of pairwise intersections of surfaces from $\mathcal{G}$. Hence, each component of Σ must lie on a single such surface. Partition Σ into sets according to the surface of $\mathcal{G}$ they lie on. Each consists of a number of connected components of Σ. Add η to the resulting family $\mathcal{F}$ of compact disjoint sets. We claim that all sets but η are trivial with respect to $\mathcal{F}$. Then, by Fact 7, so is η. However, we just observed that this is not the case, leading to a contradiction.

It remains to argue the claim that the intersection of Σ with the boundary of a path-connected component X of, say, $\bar{B}_{12}$ is trivial in $\mathcal{F}$. It lies in a compact contractible set $\bar{B}_{ij} \cap U$, which avoids all other components of Σ, as they lie on the other surfaces, outside of $\bar{B}_{ij}$. Moreover, it avoids η since Δ lies in the interior of free space and σ lies strictly outside U, except for $\mathbf{0}$ and $\mathbf{1}$. What if $\mathbf{0} \in X$ and, as before, $X \subset \bar{B}_{12}$? (The other cases are handled symmetrically.) Then the t_3-axis of U lies in X and in S. Note that $\Sigma \cap \partial \bar{B}_{12}$ avoids it. We remove a small open neighborhood of the t_3-axis from $\bar{B}_{12}$, possibly splitting X into several components and increasing the number of path-connected components of $\bar{B}_{12}$. However, each resulting connected component of X is still contractible and the set $\Sigma \cap \partial X$ is not affected by this transformation. If the removal of the neighborhood of the t_3-axis from X splits it into several path-connected components, we intersect the boundary of each component with Σ and use the resulting sets to replace $\Sigma \cap \partial X$ in $\mathcal{F}$.

Finally, $\Sigma \cap h_i$ is also trivial with respect to $\mathcal{F}$, by similar reasoning. It lies in a facet of U which is contractible and avoids the remainder of Σ and η, except possibly for $\mathbf{0}$ or $\mathbf{1}$. Suppose the facet contains $\mathbf{0}$ ($\mathbf{1}$ is handled similarly). However, observe that $\mathbf{0}$ avoids $\Sigma \cap h_i$, hence the facet can be replaced by a compact contractible set, namely a square with the $\mathbf{0}$ corner cut off,

---

[1] Technically, we have shown that this occurs for any *semi-algebraic* deformation, or more generally any transformation γ for which $\gamma^{-1}(\mathcal{FR})$ and $\gamma^{-1}(\mathcal{FR}')$ are CW-complexes. However, if there were a continuous deformation of a semi-algebraic path σ to a semi-algebraic path Δ avoiding a semi-algebraic set Σ, there would be a semi-algebraic deformation with such properties. Indeed, suppose such a continuous deformation is possible. Then there is a positive minimum distance that is achieved between the path being moved and Σ. Using Milnor's perturbation technique [14] we construct an open neighborhood Σ^+ of Σ that lies within half that distance from Σ and such that Σ is a strong deformational retract of Σ^+. We are intersted in constructing a semi-algebraic deformation of σ to Δ in $\mathbb{R}^3 \smallsetminus \Sigma^+$, if we know that a continuous deformation exists. Consider the cylindrical algebraic decomposion [6] of η and Σ^+ and refine it to a triangulation. Contractibility of η can be decided by considering only the incidence structure of the triangulation. Thus the assertion that η is contractible in $\mathbb{R}^3 \smallsetminus \Sigma^+$ becomes a statement in a first-order theory of reals and therefore the deformation can be carried out semi-algebraically.

showing triviality of $\Sigma \cap h_i$ with respect to $\mathcal{F}$.

To summarize, we have proven that all sets except for η are trivial in $\mathcal{F}$, with respect to $\mathcal{F}$, contradicting Fact 7, as claimed, and completing the proof of Proposition 4. $\qquad\square$

The existence of a **0-1** path along the skeleton of the coordination diagram implies the existence of a multi-path that uses certain types of multi-configurations only, as explained next. The skeleton consists of those features of the modified free region, which are the intersections of $2 - k$ boundaries of closed cd-obstacles with k facets of U, for some k with $0 \leqslant k \leqslant 2$. This means that points on the skeleton correspond to the following type of *permissible multi-configurations*:

> There are k, for some $k = 0, 1, 2$, robots placed at either their start or goal configuration, and $2 - k$ pairs of robots that are in contact.

There are several different ways in which a permissible multi-configuration can be achieved. We mention a few of the possibilities. One type of permissible multi-configuration is that R_1 is at its start configuration and R_2 and R_3 form a contact robot. Another type is that R_1 is at its goal configuration, R_2 touches R_1, and R_3 is unconstrained. Although the number of types of permissible multi-configurations is fairly large, it is a constant. Each type of permissible multi-configuration gives rise to a configuration space slice, as in the previous section. In the sequel we make the following assumption, similar to the DOF-Reduction Assumption we had for a pair of robots.

DOF-Reduction Assumption (for triples of robots): Let R_1, R_2, and R_3 be a triple of robots. Suppose we require k, for some $k = 0, 1, 2$, robots to be placed at either their start or goal configuration, and $2 - k$ pairs of robots to be in contact. Then the number of degrees of freedom of the resulting robot system is at most $d_1 + d_2 + d_3 - 2$, where d_i is the number of degrees of freedom of R_i.

Generally, every touching pair of robots reduces the number of degrees of freedom of the multi-robot by one, and every robot fixed at its start or goal reduces this number by at least one (namely, by the number of its degrees of freedom), so this condition will be fulfilled. As we already noted, however, there are certain 'degenerate' situation where it is not.

We now proceed as in the previous section: each type of permissible multi-configuration gives a configuration-space slice in which we compute a roadmap, we connect the roadmaps at a suitable collection of transition points (when the number of configurations common to two slices is infinite, we again choose a representative point from each face of every dimension bounding the free parts in such slices) and search the resulting graph to finally find a feasible multi-path. We obtain the following result.

Theorem 8 *Let R_1, R_2, R_3 be three robots satisfying the DOF-Reduction Assumption, operating in a workspace with n obstacles. Then we can compute a feasible multi-path for a given triple of start and goal configurations for R_1, R_2, R_3 in $O(n^{d-1})$ time, where d is the sum of the degrees of freedom of the three robots, if a feasible multi-path with positive clearance exists, and otherwise report failure.*

5 Bounded-reach robots in low-density environments

Define the *size* of an object o, denoted by $\text{size}(o)$ to be the radius of the smallest ball enclosing it. We say that the workspace $\mathcal{W}$ with the obstacle set $\mathcal{C}$ is a λ-*low-density environment* [7, 18, 20, 22] if for any ball B, the number of obstacles $C \in \mathcal{C}$ with $\text{size}(C) \geqslant \text{radius}(B)$ that intersect B is at most λ. (Our definition is the one used by de Berg et al. [7], and is slightly different from the earlier definition in [20].) If λ is a (small) constant, we say that $\mathcal{W}$ has *low density*.

In this section we apply the ideas from the previous sections to obtain efficient solutions to the motion-planning problem for the so-called *bounded-reach* robots [21] moving in low-density

workspaces. Bounded-reach robots are basically robots that are not too large compared to the obstacles.

More precisely, they are defined as follows. Let p_R be an arbitrary reference point in a robot R. Then the *reach* of R, denoted by reach(R), is defined as the radius of the smallest ball centered at p_R that contains R, no matter in which configuration R is. For instance, if R consists of two links of length 1 that are both attached to the origin, and the reference point is the tip of one of the links, then the reach of R is 2. (If the reference point were the origin then the reach would be 1. For any two reference points in R, however, the two values reach(R) can be at most a factor of two apart.) A bounded-reach robot is now defined as a robot R with

$$\text{reach}(R) \leqslant c \cdot \min_{C \in \mathcal{C}}\{\text{size}(C)\},$$

where c is a (small) constant.

Van der Stappen et al. [22] have shown that the free space of a bounded reach robot moving in a low-density workspace has $O(n)$ complexity, irrespective of the number of degrees of freedom of the robot or the dimension of its workspace. They also show how to compute in $O(n \log n)$ time a decomposition of the configuration space, after which a feasible path between given start and goal configurations can be found in $O(n)$ time. Hence, the total amount of time to solve the motion-planning problem is $O(n \log n)$.

If we have two or more bounded-reach robots then we cannot use the result of van der Stappen et al. directly, because the multi-robot consisting of these robots does not have bounded reach: its bodies can be arbitrarily far apart. But when the number of robots is two or three, the multi-robots we have to consider when we restrict ourselves to permissible multi-configurations do have free configuration spaces with linear complexity.

Indeed, consider first the case of two bounded-reach robots. Clearly, when one robot is fixed at its start or goal configuration and the other robot moves, the free-space complexity of the moving robot is linear, because it has bounded reach. (The moving robot can be very large with respect to the fixed robot, which we now view as an additional obstacle, but this does not influence the asymptotic bound.) The remaining type of multi-configuration is where the two robots form a contact robot. Here the free-space complexity is also linear, because the reach of a contact robot is bounded by the reach of one of the constituent robots plus twice the reach of the other constituent robot.

For three robots, a similar argument shows that the free-space complexities of all the configuration-space slices we have to consider are linear. The only case which is slightly different from the cases we have for two robots is when one robot is fixed, another robot moves in contact with the fixed robot, and the third robot is free. In this case the multi-robot formed by the second and third robot does not have bounded reach. But it follows from results of van der Stappen [20] that the robot that moves in contact with the fixed robot has only a constant number of combinatorially distinct critical configurations. Combined with the fact that the third robot had bounded reach and, hence, a linear number of critical configurations, this shows that the multi-robot consisting of the second and third robot has a free space of linear complexity.

We have argued that in all the configuration-space slices we have to consider the complexity of the free space is linear. Moreover, we can use the algorithm of van der Stappen et al. to compute decompositions of these free spaces. Each cell in the resulting decomposition is bounded by a constant number of algebraic surfaces, and therefore has constant complexity. For each facet of such a free-space cell in one of the slices, we determine a transition point; due to the constant complexity of the cell, this takes constant time. Next, for each transition point thus found we determine its twins in the other slices as follows. The decomposition algorithm of van der Stappen et al. constructs a so-called *cc-partition* of a low-density workspace: a subdivision into constant-complexity regions such that the robot can intersect at most a constant number of obstacles as long as its reference point stays within a single region. The decomposition consists of a BSP-like structure that allows us to determine the work-space cell containing a given transition point in logarithmic time. Finally, we examine the constant number of configuration-space cells that

correspond to this cell to determine which cell contains the transition point. Summed over all $O(n)$ transition points, this takes $O(n \log n)$ time.

To see that this algorithm can also be applied in the exceptional case mentioned above—one robot R_1 is fixed, another robot R_2 moves in contact with R_1, and the third robot R_3 is free— pretend that R_2 does not exist, and compute the work-space decomposition of R_3. Each region in the decomposition corresponds to a region in the configuration space of R_3—and, hence, in that of R_3 combined with R_2—that is crossed by only a constant number of constraint surfaces. (These surfaces are reported by the algorithm of van der Stappen et al.) Now consider what happens if we place R_2 back into the scene. Since R_2 is forced to maintain contact with R_1, it defines only a constant number of additional constraint surfaces over the entire configuration space. When computing the free cells of a configuration-space cylinder corresponding to a region in the cc-partition, we determine which (if any) of these surfaces intersect the cylinder, and add them to the constraint surfaces reported by the algorithm. A similar reasoning shows that the computation of transition points and their twins can still be done in $O(n \log n)$ time.

This leads to the following result. Notice that the time bound of our algorithm is, as in the case of a single bounded-reach robot, independent of both the numbers of degrees of freedom of the robots and of the dimension of the workspace. Also note that we do not need the DOF-Reduction Assumption for the result below to hold.

Theorem 9

(i) Let R_1 and R_2 be two bounded-reach robots operating in a low-density workspace with n obstacles. Then we can compute a feasible multi-path for a given pair of start and goal configurations for R_1 and R_2 in $O(n \log n)$ time, or report that no such path exists.

(ii) Let R_1, R_2, and R_3 be three bounded-reach robots operating in a low-density workspace with n obstacles. Then we can compute a feasible multi-path for a given triple of start and goal configurations for R_1, R_2, and R_3 in $O(n \log n)$ time, if a feasible multi-path with positive clearance exists, or report that no such path exists.

6   Concluding remarks

We presented a general technique to plan the motions of pairs and triples of robots sharing the same workspace. By combining this with the roadmap algorithm of Basu et al. [2], we obtain an algorithm with $O(n^d)$ running time for a pair of robots and $O(n^{d-1})$ running time for a triple of robots, where d is the sum of the degrees of freedom of the individual robots. These bounds are one and two orders of magnitude, respectively, faster than what the standard method would give. Here we have assumed that the sum of the number of degrees of freedom of two robots reduces by at least one if the robots are required to stay in contact; for triples a similar assumption is made. Moreover, for triples we need to assume that a solution with positive clearance exists. Our approach becomes especially effective for bounded-reach robots operating in low-density workspaces. In this case our algorithm runs in $O(n \log n)$, both for two and for three robots, irrespective of their numbers of degrees of freedom.

Any collection of $m > 3$ robots can be viewed as a triple of robots, one of which is a multi-robot consisting of $m - 2$ robots. Hence, our result for triples can be applied to four or more robots. This will reduce the dimension of the configuration space one has to consider by two. Greater savings are possible if for any m there would be a **0-1** path along the 1-skeleton of the m-dimensional coordination diagram. If this is true (which we have been unable to prove or disprove), the resulting reduction in dimension would be $m - 1$ and the motion-planning problem for m robots would be solvable in $O(n^{d-m+2})$ time.

References

[1] R. Alami, F. Robert, F. Ingrand, and S. Suzuki. Multi-robot cooperation through incremental plan merging. In *Proc. IEEEE Internat. Conf. Robot. Automat.*, pages 2573–2578, 1995.

[2] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, pages 1–15. 1997.

[3] S. Basu, R. Pollack, and M.-F. Roy. On computing a set of points meeting every semi-algebraically connected component of a family of polynomials on a variety. *J. Complexity*, 13:28–37, 1997.

[4] J. Canny. *The Complexity of Robot Motion Planning*. ACM – MIT Press Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1987.

[5] J. Canny. Computing roadmaps in general semialgebraic sets. *Comput. J.*, 36:409–418, 1994.

[6] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata Theory and Formal Languages*, volume 33 of *Lecture Notes Comput. Sci.*, pages 134–183. Springer-Verlag, Berlin, West Germany, 1975.

[7] M. de Berg, M. J. Katz, A. F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 294–303, 1997.

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

[9] S. J. Fortune, G. Wilfong, and C. Yap. Coordinated motion of two robot arms. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1216–1223, 1986.

[10] J. E. Hopcroft and G. T. Wilfong. Reducing multiple object motion planning to graph searching. *SIAM J. Comput.*, 15:768–785, 1986.

[11] John Hopcroft and Gordon Wilfong. Motion of objects in contact. *Internat. J. Robot. Res.*, 4:32–46, 1986.

[12] K. Kedem, R. Livne, J. Pach, and Micha Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.

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

[14] J. Milnor. On the betti numbers of real algebraic varieties. *Proc. Amer. Math. Soc.*, 15:275–280, 1964.

[15] P. A. O'Donnell and T. Lozano-Pérez. Deadlock-free and collision-free coordination of two robotic manipulators. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 484–489, 1989.

[16] J. T. Schwartz and Micha Sharir. On the "piano movers" problem II: general techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.

[17] J. T. Schwartz and Micha Sharir. On the "piano movers" problem III: coordinating the motion of several independent bodies: the special case of circular bodies moving amidst polygonal barriers. *Internat. J. Robot. Res.*, 2(3):46–75, 1983.

[18] Otfried Schwarzkopf and Jules Vleugels. Range searching in low-density environments. *Inform. Process. Lett.*, 60:121–127, 1996.

[19] Micha Sharir and S. Sifrony. Coordinated motion planning for two independent robots. *Ann. Math. Artif. Intell.*, 3:107–130, 1991.

[20] A. F. van der Stappen. *Motion Planning amidst Fat Obstacles*. Ph.D. dissertation, Dept. Comput. Sci., Utrecht Univ., Utrecht, Netherlands, 1994.

[21] A. F. van der Stappen, D. Halperin, and M. H. Overmars. The complexity of the free space for a robot moving amidst fat obstacles. *Comput. Geom. Theory Appl.*, 3:353–373, 1993.

[22] A. F. van der Stappen, M. H. Overmars, M. de Berg, and J. Vleugels. Motion planning in environments with low obstacle density. Technical Report UU-CS-1997-19, Dept. Comput. Sci., Utrecht Univ., 1997.

[23] C. W. Warren. Multiple robot path coordination using artificial potential fields. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 500–505, 1990.