FINDING TOPOLOGY IN A FACTORY: CONFIGURATION SPACES

A. ABRAMS AND R. GHRIST

It is perhaps not universally acknowledged that an outstanding place to find interesting topological objects is within the walls of an automated warehouse or factory.

The examples of topological spaces that we construct in this exposition arose simultaneously from two seemingly disparate fields: the first author, in his thesis [1], discovered these spaces after working with H. Landau, Z. Landau, J. Pommersheim, and E. Zaslow on problems about random walks on graphs [2]. The second author discovered these same spaces while collaborating with D. Koditschek in the Artificial Intelligence Lab at the University of Michigan; see [7] and [8].

Sections 1 and 2 give some motivations arising from robotics, as well as a little background on configuration spaces. For the remainder of the paper we focus on a fascinating class of topological spaces related to motion-planning on graphs.

1. Robotics and topological motion planning

Consider an automated factory equipped with a cadre of Automated Guided Vehicles (AGVs), or mobile robots, which transport items from place to place. A common goal is to place several, say N, of these AGVs in motion simultaneously, controlled by an algorithm that either guides the AGVs from initial positions to goal positions (in a warehousing application), or executes a cyclic pattern (in manufacturing applications). These robots are costly and cannot tolerate collisions (with obstacles or with each other) without a loss of performance.

Anyone who shops at a large supermarket with wide aisles is familiar with this problem and a solution. If two carts are headed toward each other, a slight swerve is sufficient to avoid a collision, assuming the other does not move in the same direction! *The resolution of collisions on $\mathbb{R}^2$ is a local phenomenon.*

This does not imply that planning coordinated motions is a simple task: it requires an extraordinary effort to coordinate air traffic at a busy airport, a scenario in which a collision is to be avoided at all costs. A nontrivial planar example appears in Figure 1: how does one coordinate the exchange of two lines of AGVs without collisions?

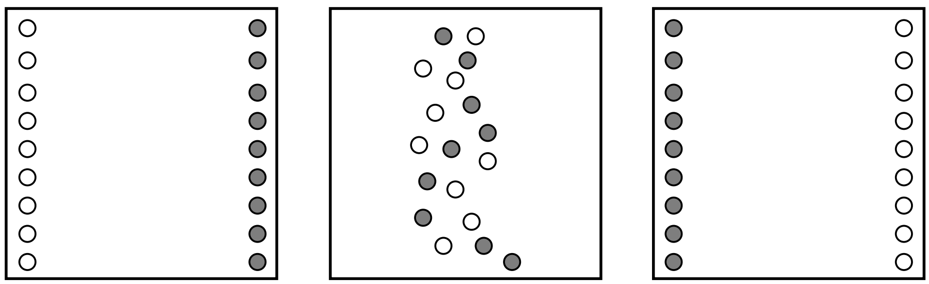

FIGURE 1. Coordinating the safe, optimal exchange of AGVs from one side to the other is a nontrivial planning problem.

One method used by the robotics community to solve coordinated motion problems is topological in nature. Loosely speaking, one constructs the space of all possible arrangements of the robots, and then one removes all arrangements that are at or near a collision. What remains is a space that collates all of the "safe" configurations of the robots. Paths on this space yield safe coordinated motions.

2. Configuration spaces

The formal construction yields a classical topological object: the configuration space of N distinct labeled points on the plane $\mathbb{R}^2$. That is, we consider all ordered N-tuples of points in $\mathbb{R}^2$ with the property that no two of the points coincide. To each such N-tuple, we assign a point in the configuration space. Two points in the configuration space are close if the N-tuples are close as measured in Euclidean $\mathbb{R}^{2N}$. Formally, the configuration space is

$$\mathcal{C}^N(\mathbb{R}^2) := \left(\mathbb{R}^2 \times \mathbb{R}^2 \times \cdots \times \mathbb{R}^2\right) - \Delta,$$

where Δ denotes the "pairwise diagonal"

$$\Delta := \{(x_1, x_2, \ldots, x_N) \in (\mathbb{R}^2)^N : x_i = x_j \text{ for some } i \neq j\}.$$

This configuration space is not easy to visualize, in no small part because the dimension of the space is $2N$.

EXERCISE: Show that $\mathcal{C}^2(\mathbb{R}^2)$ is homeomorphic to $\mathbb{R}^2 \times (\mathbb{R}^2 - (0,0))$, or, equivalently, to $\mathbb{R}^3 \times S^1$, where S^1 denotes the unit circle in $\mathbb{R}^2$. Hint: think about placing tokens on the table one at a time. Does your method of proof give a simple presentation for $\mathcal{C}^3(\mathbb{R}^2)$?

Thus, modeling the factory floor as $\mathbb{R}^2$ and the AGVs as points, one often wishes to find paths or cycles in $\mathcal{C}^N(\mathbb{R}^2)$ to enact specific behaviors. Obstacles can easily be incorporated into these models — there is a vast literature on this subject [12]. Executing cyclic motions is more complex but can at first be approximated by composed point-to-point motions. Several assumptions are required for the simple construction we present, and various kinematic issues (e.g., steering geometry) must be addressed in general. Of course, since the robots are not truly points, and since no control algorithm implementation is of infinite precision, we require that the control path reside outside of a neighborhood of the diagonal Δ in $(\mathbb{R}^2)^N$.

It is possible to construct safe control schemes using configuration spaces. The work of Koditschek and Rimon [11] provides one example of a concrete solution: they write out explicit vector fields on these configuration spaces that can be used to flow from initial to goal positions in the presence of certain types of obstacles. By arranging these vector fields so that they strongly push away from the vestiges of the diagonal Δ on the boundary of $\mathcal{C}^N(\mathbb{R}^2)$, the control scheme is *provably* safe from collisions (as opposed to being *statistically* safe via computer simulations): no path can ever intersect the diagonal. Furthermore, since a neighborhood of the diagonal is repelling, the control scheme is *stable* with respect to perturbations to the system. This is quite important, as mechanical systems have an annoying tendency to malfunction occasionally. Drawing an appropriate vector field on a configuration space yields an excellent method of self-correction.

This is a clean, direct application of topological and dynamical ideas to a matter of great practical relevance, and it is currently used in various industrial settings.

3. Graphs

The robotics community, largely independently of the topology community, has enjoyed great success at identifying and manipulating configuration spaces to their advantage in control problems. There is, however, a class of simple, physically relevant scenarios whose configuration spaces have been untapped: the configuration spaces of points on a *graph*, or a network of edges and vertices.

Suppose the AGVs must move about on a collection of tracks embedded in the floor, or via a path of electrified guide-wires from the ceiling; see [6] for examples. Such a restricted network is quite common, mainly because it is more cost-effective than a full two degree-of-freedom steering system for AGVs. In this setting, the state of the system at any instant of time is a point in the configuration space of the graph Γ:

$$\mathcal{C}^N(\Gamma) := (\Gamma \times \cdots \times \Gamma) - \Delta.$$

As before, to navigate safely on a graph, one must construct appropriate paths that remain strictly within $\mathcal{C}^N(\Gamma)$ and are repulsed by any boundaries near Δ. But several problems seem to prevent an analogous solution, including the following:

1. What do these spaces look like?
2. How does one resolve an impending collision?

The first difference between this problem and the problem of $\mathcal{C}^N(\mathbb{R}^2)$ is that $\mathcal{C}^N(\Gamma)$ is *not* a manifold: that is, you cannot hope that every point has a neighborhood that is locally homeomorphic to a Euclidean space. Indeed, if we ignore trivial graphs that are homeomorphic to a line segment or a circle, then the graph itself is not locally Euclidean and products of the graph still share this feature. A second difference is that collisions within the interior of an edge are no longer locally resolvable. Imagine that the aisles of a grocery store are only as wide as the shopping carts, so that passing another person is impossible. A store full of shoppers (using carts) would pose a difficult coordinated control problem. How can carts avoid a collision in the interior of an aisle? Clearly, at least one of the participants must make a large-scale change in plans and back up to the end of the aisle. *The resolution of a collision on a graph is a non-local phenomenon.*

The remainder of this paper explores the topological features of these interesting spaces. For more in-depth applications of configuration spaces to problems in motion planning, see [12] and the references therein.

4. Examples: two robots

Since a graph is a one-dimensional object, the configuration space $\mathcal{C}^N(\Gamma)$ is N-dimensional. Thus configuration spaces of two robots are two-dimensional objects, which (at least in simple cases) one should be able to visualize. Three simple examples follow.

EXAMPLE 1: $\mathcal{C}^2(\top)$

Let $\top$ denote the graph of three edges attached at a central vertex. It is straighforward to determine the cellular structure of $\mathcal{C}^2(\top)$. The product graph $\top \times \top$ consists of nine squares or "2-cells" glued together. Of these, six correspond to configurations in which the two robots are on distinct edges of $\top$. Since there are three edges in $\top$, the remaining configurations, in which both robots are on the same edge, yield three square cells, each of which is divided by the diagonal Δ into a pair of triangular cells. Thus there are six triangular 2-cells corresponding to the configurations in which both robots are on the same edge, but at distinct locations. By enumerating the behaviors of each of these 2-cells, one can make the identifications to arrive at the space given in Figure 2; one can also use the slightly more sophisticated argument in [9].

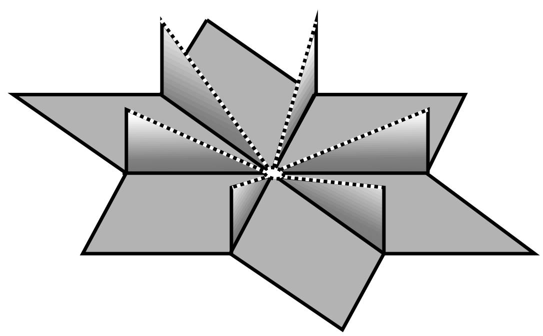

FIGURE 2. The configuration space $\mathcal{C}^2(\top)$ embedded in $\mathbb{R}^3$. Dotted lines refer to edges that lie on the diagonal Δ. Note that the central vertex is deleted.

EXERCISE: Place two coins on a piece of paper with a large $\top$ drawn on it so that both coins are on the same edge of the graph. Using two fingers, execute a path that exchanges the coins' positions without collisions. Draw this motion as a path on Figure 2.

EXAMPLE 2: $\mathcal{C}^2(⟜)$

Let ⟜ denote the graph with three edges obtained from $\top$ by gluing two boundary vertices together. One method of constructing $\mathcal{C}^2(⟜)$ would be to first remove the configurations in which both robots are on the vertices to be glued. Then identify those portions of the boundary of $\mathcal{C}^2(\top)$ that have a robot at the vertices to be glued in ⟜, and glue these portions of $\mathcal{C}^2(\top)$ together. The result, although a very simple configuration space, is already somewhat difficult to visualize: we illustrate the space, embedded in $\mathbb{R}^3$, in Figure 3 [left]. Each of the three "punctures" corresponds to a collision of the robots at one of the three vertices. The six dotted edges are the images of the diagonal curves from Figure 2.

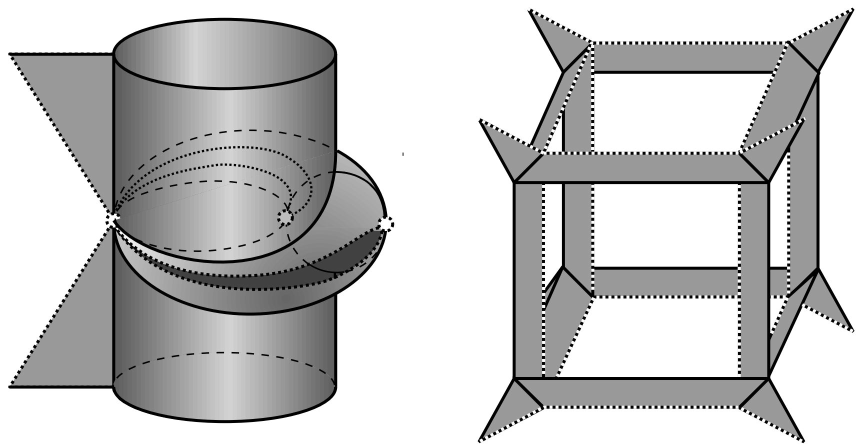

FIGURE 3. The configuration spaces $\mathcal{C}^2(\multimap)$ [left] and $\mathcal{C}^2(\times)$ [right]. Dotted lines refer to vestiges of the diagonal Δ.

EXAMPLE 3: $\mathcal{C}^2(\times)$

Increasing the incidence number of the central vertex complicates the configuration space. Consider $\times$, a radial tree of four edges emanating from a central vertex. The visualization of $\mathcal{C}^2(\times)$ is a bit more involved and requires some work to obtain. For the purpose of stimulating curiosity, we include this configuration space as Figure 3 [right].

5. Simplification: discretization

To visualize more complicated configuration spaces of graphs, some simplification into a more manageable form is necessary. We use two principal methods: the first is a way of removing the "unsafe" cells of the space near the diagonal Δ. The second, deformation retraction, is a more drastic crushing of the space down to a lower dimensional "skeleton"; we discuss it in Section 6.

Any graph Γ comes equipped with a cellular structure: 0-cells (vertices) and 1-cells (edges). The N-fold cross product of Γ with itself inherits a cell structure, each cell being a product of N (not necessarily distinct) cells in Γ. However, as in Example 1, the configuration space does not quite have a natural cell structure, since the diagonal Δ slices through all product cells with repeated factors. Notice, however, that in several of the previous examples, these partial cells dangle "inessentially" and could be collapsed onto a more "essential" skeleton of the configuration space.

This skeleton can be specified as follows [1]. Consider the *discretized configuration space* of Γ, denoted $\mathcal{D}^N(\Gamma)$, defined as $(\Gamma \times \cdots \times \Gamma) - \tilde{\Delta}$, where $\tilde{\Delta}$ denotes the set of all product cells in $\Gamma \times \cdots \times \Gamma$ whose closures intersect the diagonal Δ. Equivalently, we can describe $\mathcal{D}^N(\Gamma)$ as the set of configurations for which, given any two robots on Γ and any path in Γ connecting them, the path contains at least one entire edge. Thus, instead of restricting robots to be at least some intrinsic distance ϵ apart (i.e., removing an ϵ neighborhood of Δ), one now restricts robots on Γ to be "at least one full edge apart". This is a natural kind of configuration space in the context of random walks on graphs [2]. Note that $\mathcal{D}^N(\Gamma)$ is a subcomplex of $\mathcal{C}^N(\Gamma)$ (it does not contain "partial cells" that arise when cutting along the diagonal), and is, in fact, the largest subcomplex of $\mathcal{C}^N(\Gamma)$ that does not intersect Δ.

With this natural cell structure, one can think of the vertices (0-cells) of $\mathcal{D}^N(\Gamma)$ as "discretized" configurations — arrangements of labeled tokens at the vertices of the graph. The edges of $\mathcal{D}^N(\Gamma)$, or 1-cells, tell us which discrete configurations can be connected by moving one token along an edge of Γ. Each 2-cell in $\mathcal{D}^N(\Gamma)$ represents two independent (or "commuting") edges: one can move a pair of tokens independently along disjoint edges. A k-cell in $\mathcal{D}^N(\Gamma)$ likewise represents the ability to move k tokens along k disjoint edges in Γ.

Returning to Figure 2, discretizing $\mathcal{C}^2(\top)$ removes much of the space. For example, the triangular two-dimensional cells represent configurations in which both robots are on the interior of the same edge. Since they are not "one full edge apart", these cells are deleted. The same is true of all the other two-dimensional cells that represent robots in the interior of separate edges. Which configurations of two robots on $\top$ are separated by a full edge?

EXERCISE: Show that discretizing the configuration spaces of Examples 1 through 3 yields the configuration spaces of Figure 4. How well do these spaces "approximate" the configuration spaces?

One could compute the discretization of Example 1 in a less direct manner that generalizes to some lovely examples to follow. Recall that the discretized

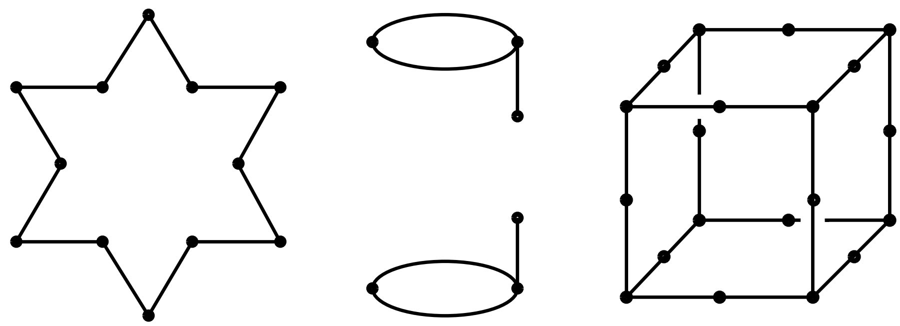

FIGURE 4. The discretizations of the configuration spaces in Examples 1-3 (left to right).

configuration spaces inherit a cell structure from Γ: all the product cells of Γ^N not entirely in $\mathcal{C}^N(\Gamma)$ are removed by the discretization. Thus, in Example 1, simple counting reveals that the space $\mathcal{D}^2(\top)$ possesses twelve 0-cells (both robots are at distinct vertices of $\top$), twelve 1-cells (one robot is at a vertex and the other is on an edge whose closure does not contain said vertex), and zero 2-cells (since any two edges intersect at the central vertex). With a little thought, one can see that $\mathcal{D}^2(\top)$ is a connected manifold: each zero-cell connects to exactly two 1-cells, and all of the 1-cells are joined end-to-end cyclically. Thus, $\mathcal{D}^2(\top)$ is a topological circle, precisely as obtained by deleting all the near-diagonal cells from $\mathcal{C}^2(\top)$ in Figure 2. The discretization operation yields a subcomplex of $\mathcal{C}^2(\top)$ that appears to contain all the "essential" topology (more specifically, the spaces $\mathcal{C}^N(\top)$ and $\mathcal{D}^N(\top)$ are of the same *homotopy type* — see Section 6 for definitions); however, this is certainly not the case for the discretization of $\mathcal{C}^2(\circ\!\!-)$, which becomes disconnected! In Section 6, we state the criteria under which discretization is topologically faithful.

Counting arguments like those above can often determine the discretized configuration space, even when the full configuration space is unknown. The following are some surprising examples of interesting spaces that arise as the discretized configuration space of non-planar graphs [1].

EXAMPLE 4: $\mathcal{D}^2(K_5)$

Consider the complete graph K_5 pictured in Figure 5 [left]. The discretized configuration space of two robots on this graph is a two-dimensional complex. A simple counting argument reveals the cell-structure:

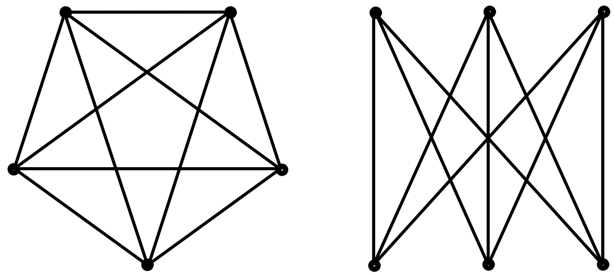

FIGURE 5. The non-planar graphs K_5 [left] and $K_{3,3}$ [right]. This notation comes from graph theory, where these are fundamental examples of non-planar graphs.

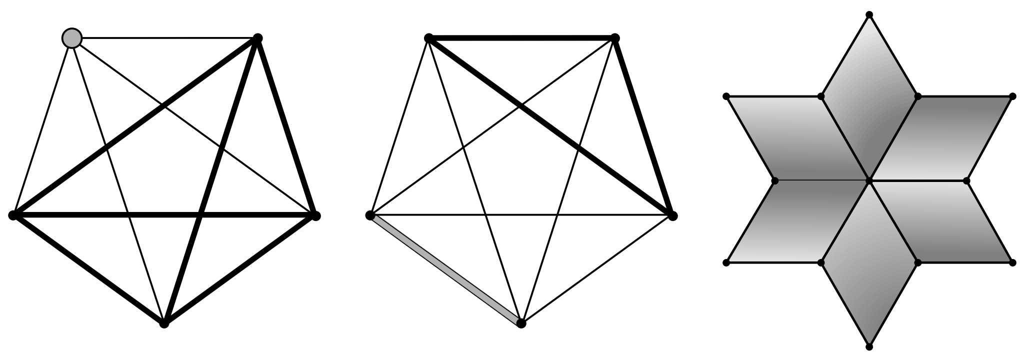

FIGURE 6. [left] For every vertex in the space K_5 there are six disjoint edges. Likewise [middle] for each edge there are three totally disjoint edges. In $\mathcal{D}^2(K_5)$, these cells fit together to form a locally Euclidean two-dimensional complex [right].

- 0-CELLS Each 0-cell corresponds to a configuration in which the two robots are at distinct vertices. Since K_5 has five vertices, there are exactly $(5)(5-1) = 20$ such 0-cells. (There is no vertex where two edges cross in the picture; there are vertices only at the corners of the pentagon.)
- 1-CELLS Each 1-cell corresponds to a configuration in which one robot is at a vertex and the other is on an edge whose endpoints do not include the vertex already occupied. From the diagram of K_5 one counts that there are $(2)(5)(6) = 60$ such 1-cells, as in Figure 6 [left]. The factor of two is present because we label the two robots on K_5.
- 2-CELLS Each 2-cell corresponds to a configuration in which the two robots occupy edges whose closures are disjoint. Again, from the diagram (and

Figure 6 [middle]) one counts that there are $(10)(3) = 30$ such 2-cells in the complex.

One then demonstrates that each edge borders a pair of 2-cells preserving an orientation and that each vertex is incident to six edges, as in Figure 6 [right]. Also, the space $\mathcal{D}^2(K_5)$ is connected: one can move from any configuration to any other. Thus $\mathcal{D}^2(K_5)$ is a connected orientable surface, and the classification theorem for surfaces implies that the space is determined uniquely up to homeomorphism by the *Euler characteristic*:

$$\chi(\mathcal{D}^2(K_5)) := \#\text{faces} - \#\text{edges} + \#\text{vertices} = 30 - 60 + 20 = -10. \tag{5.1}$$

Thus, $\mathcal{D}^2(K_5)$ is a closed orientable surface of genus $g := 1 - \frac{1}{2}\chi = 6$. It is not at all obvious that the motion of two robots on this graph should produce a genus six surface. Obtaining a manifold is surprising enough, but a manifold with genus larger than one is at odds with the notion that all of the interesting topology in these spaces is "localized" in configurations about a vertex.

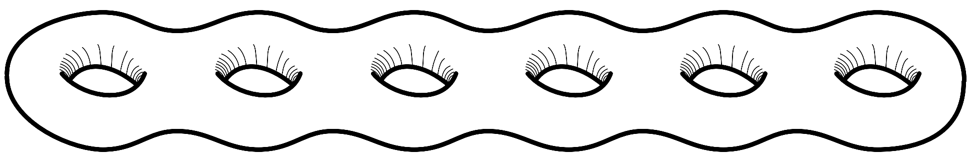

FIGURE 7. The space $\mathcal{D}^2(K_5)$ is homeomorphic to a closed orientable surface of genus six.

EXAMPLE 5: $\mathcal{D}^2(K_{3,3})$

A near-identical analysis on the graph $K_{3,3}$ of Figure 5 [right] reveals that $\mathcal{D}^2(K_{3,3})$ is also a connected closed orientable surface. The natural cell structure on $\mathcal{D}^2(K_{3,3})$ possesses exactly 36 faces, 72 edges, and 30 vertices. Thus,

$$\chi(\mathcal{D}^2(K_5)) = 36 - 72 + 30 = -6, \tag{5.2}$$

and we conclude that the genus of this surface is four.

In order to understand discretizations of higher-dimensional configuration spaces, one can employ an appropriate version of duality [1]: instead of tracking how distinct robots on Γ move about, one considers configurations of "holes" — regions on Γ that have no robots on them. As the holes have no natural labeling, the duality applies directly to the *unlabeled* configuration spaces. This duality implies a relationship between $\mathcal{D}^N(\Gamma)$ and $\mathcal{D}^{V-N}(\Gamma)$, where V is the number of vertices of Γ. These arguments establish the following examples.

EXAMPLE 6: The space $\mathcal{D}^3(K_5)$ is homeomorphic to a connected closed orientable surface of genus 16.

EXAMPLE 7: Likewise, $\mathcal{D}^4(K_{3,3})$ is homeomorphic to a connected closed orientable surface of genus 37.

6. Simplification: deformation

We are naturally led to the question of how well the discretized configuration spaces approximate the true configuration spaces. Heuristically, the spaces $\mathcal{C}^N$ and $\mathcal{D}^N$ should be similar, since the latter is a subset of the former obtained by collapsing out those cells that border the remains of the diagonal Δ. However, the discretization is not always faithful: $\mathcal{C}^2$(⟜) is connected while $\mathcal{D}^2$(⟜) is not. Indeed, by definition, $\mathcal{D}^N(\Gamma)$ is the empty set whenever N is greater than the number of vertices of Γ, i.e., when the discretization of Γ is too "coarse".

The notion of "sameness" appropriate here is that of *deformation retraction*. A subspace A of a space X is a (strong) deformation retraction of X if there exists a continuous family of continuous maps $f_t : X \to X$ such that f_0 is the identity map on X and f_1 is a map that sends X onto A, and such that f_t fixes A pointwise for all t. The images of the f_t can be seen as frames in a movie that exhibits a continuous shrinking of X onto A. Deformation retractions are an excellent way to simplify a space without changing any essential topological properties (except perhaps dimension, but for the applications at hand, reducing the dimension of the configuration space is a boon). The most fundamental notion of topological equivalence, *homotopy type*, can be defined in terms of deformation retractions: two spaces X and Y are of the same homotopy type if and only if they are both deformation retractions of a "larger" space Z.

The key result is that $\mathcal{D}^N(\Gamma)$ is a deformation retraction of $\mathcal{C}^N(\Gamma)$ as long as the discretization of the graph Γ is not too coarse. More specifically,

Theorem 1: [1] *For any $N > 1$ and any graph Γ with at least N vertices, $\mathcal{C}^N(\Gamma)$ deformation retracts to $\mathcal{D}^N(\Gamma)$ if and only if*

(1) *Each path between distinct vertices of valence not equal to two passes through at least $N - 1$ edges; and*
(2) *Each path from a vertex to itself that cannot be shrunk to a point in Γ passes through at least $N + 1$ edges.*

(The valence of a vertex is the number of incident edges.)

It thus follows that the spaces $\mathcal{C}^2(K_5)$ and $\mathcal{C}^2(K_{3,3})$ deformation retract to the discretized configuration spaces computed in Examples 4 and 5. This is extremely useful information: trying to compute the configuration space $\mathcal{C}^2(K_5)$ directly would appear hopelessly complex. Note that the second condition of Theorem 1 fails for the discretization of $\mathcal{C}^2$(○–), but adding one more vertex yields a faithful discretized configuration space. The discretizations of Examples 6 and 7 are not fine enough to give an equivalence.

The dimension of the smallest subcomplex to which a configuration space deformation retracts is an important quantity in practice, since a large dimension greatly increases the complexity of the computational work needed to control the system. The following theorem reveals that the essential dimension of the configuration space is governed by properties of the graph, independent of the number of robots on the graph.

Theorem 2: [7] *Given a graph Γ having V vertices of valence greater than two, the space $\mathcal{C}^N(\Gamma)$ deformation retracts to a subcomplex of dimension at most V.*

EXAMPLE 8: $\mathcal{C}^N(\Upsilon_k)$

Consider the radial k-prong tree Υ_k having $k > 2$ edges and $k+1$ vertices, all edges being attached at a single central vertex. For example, $\Upsilon_3 =$ ⊤, and $\Upsilon_4 =$ ✕. Theorem 2 ensures that $\mathcal{C}^N(\Upsilon_k)$ deformation retracts to a 1-dimensional subcomplex — that is, a graph. Since the essential topological features of a graph are determined by its Euler characteristic, one need merely compute the number of vertices and edges to classify these spaces. Using a double-induction argument on N and k [7], one derives a two-variable recursion relation for the Euler characteristic. By solving this equation, one can prove that $\mathcal{C}^N(\Upsilon_k)$ has the homotopy type of a graph that is a “bouquet” of P distinct loops joined together like petals on a daisy, where

$$P = 1 + (Nk - 2N - k + 1)\frac{(N+k-2)!}{(k-1)!}. \tag{6.1}$$

Note, for example, that $\mathcal{C}^2(\Upsilon_3)$ has exactly one generating loop, as Figures 2 and 4 [left] confirm. Figure 4 [right] provides another confirmation of this equation for $k = 4$, as the reader should verify. The factorial growth of P in N is due to the fact that we label the N robots on Υ_k. If one considers the unlabeled configuration spaces, then the second term in (6.1) is reduced by a factor of $N!$.

It is worth emphasizing that while the control problem of robots on a graph is rather intuitive for two robots, it quickly builds in complexity. Since the dimension alone makes most configuration spaces nearly impossible to visualize, Theorem 2 is quite helpful — the "essential" dimension of the configuration space is independent of the number of robots on the graph. For the graph Υ_k, Theorem 2 implies that there is a one-dimensional *roadmap* that gives a perfect representation of the configuration space: no topological data are lost. Since the proof of Theorem 2 is constructive, one can use standard algorithms for determining shortest paths on a graph to develop efficient path planning for multiple robots on Υ_k via the roadmap.

7. Conclusions

Applications of configuration spaces to robotics are by no means novel: ideas by computer scientists, engineers, and mathematicians have been evolving since the 1960's; [5] gives an introduction. In addition, various kinds of configuration spaces arise in topology and physics rather often, as in the study of braids [3], abstract linkages ([13] and [10]), or invariants of manifolds [4].

In applications involving multiple independent robots, the global aspect of the control problem is the principal difficulty when the robots are constrained to a network. It is this global nature that hints at the efficacy of a topological viewpoint. Indeed, the determination and simplification of the configuration spaces of graphs can be used to construct practical control schemes; see [8] and [9] for some simple examples.

Reversing the perspective, it is remarkable that this class of topologically rich configuration spaces was virtually untouched until motivated by problems from other fields. There are many deeper properties of these spaces:

(1) For any graph Γ, $\mathcal{C}^N(\Gamma)$ is an Eilenberg-MacLane space of type $K(\pi, 1)$. That is, the image of any continuous map from a k-dimensional sphere S^k, $k > 1$, into $\mathcal{C}^N(\Gamma)$ can be shrunk to a point in $\mathcal{C}^N(\Gamma)$. Such a space is sometimes descriptively called "aspherical".
(2) The discretized configuration space has a natural structure of a cube-complex, since all the cells are products of intervals. Consequently, one can use recent fast algorithms from computational homology to determine homology groups and generators in practical settings.
(3) Using the cube complex structure of $\mathcal{D}^N(\Gamma)$, one can show that $\mathcal{C}^N(\Gamma)$ is an NPC (non-positively curved) space: there exists a metric whose curvature (defined appropriately at the non-manifold points) is never positive.

(4) The fundamental group of $\mathcal{C}^N(\Gamma)$ is always torsion-free. In other words, if a loop in $\mathcal{C}^N(\Gamma)$ cannot be shrunk to a point, then neither can any multiple of the loop. This property is also true for configuration spaces of $\mathbb{R}^2$, but *not* for $\mathcal{C}^N(S^2)$ — robots on a two-dimensional sphere.

(5) The fundamental group of $\mathcal{C}^N(\Gamma)$ always has a solvable word problem, which means that there is an algorithm for deciding whether any given loop in $\mathcal{C}^N(\Gamma)$ can be shrunk to a point in $\mathcal{C}^N(\Gamma)$.

(6) The fundamental group $\pi_1(C^N(\Gamma))$ has the algebraic structure of a *graph of groups* over the graph Γ: it can be obtained by "amalgamating" (or algebraically gluing together) a collection of simpler groups in a pattern described by Γ.

For those not familiar with these more subtle features of topological spaces and their fundamental groups, the examples presented here pose an excellent concrete manifestation of these properties. Indeed, one can easily explain what a configuration space of robots is to a high-school class (both authors have done this on several occasions with positive results). With very little background, one can readily understand the proof that two robots on K_5 has a genus six surface as its configuration space: this provides an excellent manifestation of the nontrivial topology lurking behind many physical settings.

8. Acknowledgements

The second author is supported in part by NSF Grant # DMS-9971629. The authors thank Margaret Symington for her careful reading of the manuscript.

References

[1] A. Abrams, *Configuration Spaces and Braid Groups of Graphs*, PhD thesis, UC Berkeley, 2000.

[2] A. Abrams, H. Landau, Z. Landau, J. Pommersheim, and E. Zaslow, Evasive random walks, preprint.

[3] J. Birman, *Braids, Links, and Mapping Class Groups*, Princeton University Press, Princeton, N.J., 1974.

[4] R. Bott and C. Taubes, On the self-linking of knots: topology and physics, *J. Math. Phys.* 35 (1994) 5247–5287.

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

[6] G. Castleberry, *The AGV Handbook*, Braun-Brumfield, Ann Arbor, MI, 1991.

[7] R. Ghrist, Configuration spaces of graphs in robotics, in *Braids, Links, and Mapping Class Groups: the Proceedings of Joan Birman's 70th Birthday*, AMS/IP Studies in Mathematics vol. 19, 2000, pp. 31–41. ArXiv preprint **math.GT/9905023**.

[8] R. Ghrist and D. Koditschek, Safe cooperative robot dynamics via dynamics on graphs, in Y. Nakayama, ed., *Proceedings of the Eigth International Symposium on Robotics Research*, Springer-Verlag, 1998, pp. 81–92.
[9] R. Ghrist and D. Koditschek, Safe, cooperative robot dynamics on graphs, ArXiv preprint **cs.RO/0002014**.
[10] M. Kapovich and J. J. Millson, The symplectic geometry of polygons in Euclidean space, *J. Differential Geom.* 44 (1996) 479–513.
[11] D. Koditschek and E. Rimon, Robot navigation functions on manifolds with boundary, *Adv. in Appl. Math.* 11 (1990) 412–442.
[12] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic Press, Boston, MA, 1991.
[13] K. Walker, *Configuration Spaces of Linkages*, Undergraduate thesis, Princeton University, 1985.

Department of Mathematics, University of Georgia, Athens, GA 30602, USA

E-mail address: `abrams@math.uga.edu`

School of Mathematics and Center for Dynamical Systems and Nonlinear Studies, Georgia Institute of Technology, Atlanta, GA 30332, USA

E-mail address: `ghrist@math.gatech.edu`