



Geometric Aspects of Robot Navigation: From Individual Robots to Massive Particle Swarms

Sándor P. Fekete^(✉)

Department of Computer Science, TU Braunschweig, Braunschweig, Germany
s.fekete@tu-bs.de

Abstract. We describe a spectrum of challenges and results related to geometric aspects of robot navigation, advancing from centralized methods for difficult offline problems (such as the *Art Gallery Problem*), to online tasks for many robots (as in *online exploration by a swarm of robots*), locally managing the connectivity and shape of a large swarm (i.e., *cohesive control*), all the way to *controlling massive swarms of particles by global forces*.

1 Introduction

Ever since the first work on autonomous robots, algorithmic aspects of robot navigation have played an important role, with new theoretical insights making it possible to expand the practical possibilities, and new real-world challenges motivate algorithmic innovation. Particularly important roles in these developments were played by geometry and by the advances of distributed models and methods. In the following, we provide a number of highlights: The *Art Gallery Problem* described in Sect. 2 deals with localizing stationary viewpoints for mapping all of a given, known region. Section 3 describes how to *explore and triangulate an unknown region* by a swarm of simple robots with only weak sensor and navigation capabilities. Section 4 deals with the challenge of *cohesive control*, i.e., organizing a swarm of simple agents by local interaction, such that connectivity is maintained, even in the presence of external forces and agent failures. The final Sect. 5 describes how to *control a massive swarm of particles by using uniform external forces*.

2 Art Gallery Problems

2.1 Motivation

Consider a robot platform that can produce high-resolution, virtual environments, based on a limited number of laser scans. For mapping all of a given region in the presence of obstacles, we need to compute an optimal set of scan positions. This is closely related to one of the classic problems of computational geometry: The Art Gallery Problem (AGP) asks for illuminating or surveying all of a given polygonal region P from as few positions (“guards”) as possible.

2.2 Formal Aspects

Consider a given polygonal region P . For any point $g \in P$, the *visibility region* $\mathcal{V}(g)$ is the set of all positions $p \in P$ for which there is a straightline connection between G and p that lies completely in P . The Art Gallery Problem (AGP) asks for a minimum cardinality guard set $G \subset P$ that sees all of P , i.e., such that $\bigcup_{g \in G} \mathcal{V}(g) = P$. An important distinction arises from the possible positions of guards: a *vertex guard* must be placed at a vertex of P , while a *point guard* can be located anywhere in P .

2.3 Context

As first proven by Chvátal [15] and shown by Fisk [32] in a beautiful and concise proof, $\lfloor \frac{n}{3} \rfloor$ guards are sometimes necessary and always sufficient for guarding a simple polygon P with n vertices. See O'Rourke [64] for an early overview.

Algorithmically, the AGP is NP-hard, even for a simply connected polygonal region P [51]. Eidenbenz et al. [20] showed that for a region with holes, finding an optimal set of vertex guards is at least as hard as the problem *Set Cover*, so there is little hope of achieving a better approximation guarantee than $\Omega(\log n)$. It seems unlikely that this gets any easier when allowing general point guards, as there is no known simple characterization of a discrete candidate set of guard locations. Furthermore, recent work by Abrahamsen et al. [1] proves that the AGP is complete for the existential theory of the reals, implying that it is unlikely to even belong to the class NP.

All this shows the difficulty of the AGP, but it does not rule out methods that combine structural insights with powerful mathematical tools to achieve provably optimal solutions for instances of interesting size.

Computing optimal solutions for general AGP instances is not only relevant from a theoretical point of view, but has also gained in practical importance in the context of modeling, mapping, and surveying complex environments, such as in the fields of architecture, robotics and medicine.

2.4 Application Scenario

One particular real-world platform giving rise to instances of the Art Gallery Problem is Irma3D (Intelligent Robot for Mapping Applications in 3D), an autonomous robot; see Fig. 1. Its main sensor is a Riegl VZ-400 laser scanner. A typical 3D laser scan needs 3 min, producing up to 20 million highly precise 3D measurements of the surrounding. A globally consistent scan matching is used to merge the 3D scans to a single scene [11]. Irma3D is built on a Volksbot RT-3 chassis; it uses the Xsens MTi IMU and odometry to sense its own position. See Fig. 2 for a real-world image (Top) and the schematic view with an optimal set of scan points (Bottom). The algorithmic challenge is to plan number and positioning of these scans.



Fig. 1. IRMA3D in front of the town hall of Bremen, scanning the city square.

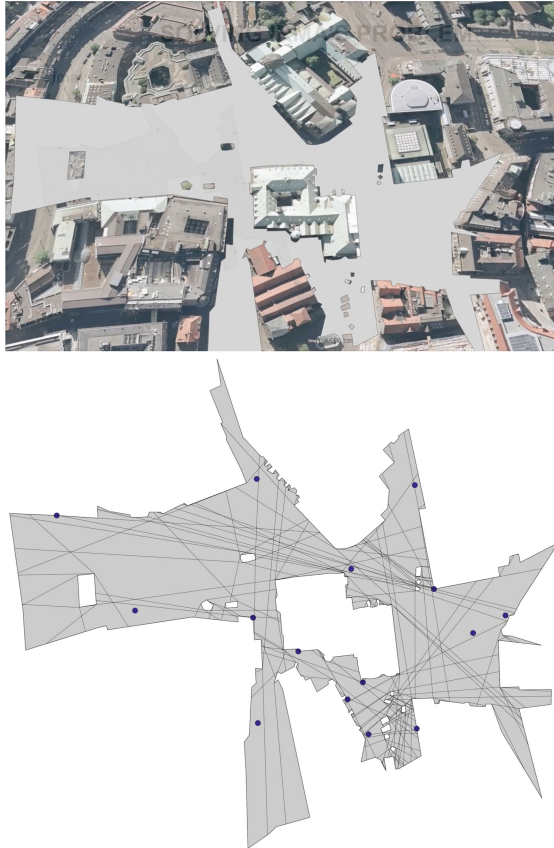


Fig. 2. (Top) part of a real-life AGP instance with 15 holes and 332 vertices: a square in the city center of Bremen. (Bottom) a corresponding extracted polygonal region, with an optimal set of scan positions, shown as 15 black dots. The white holes correspond to obstacles formed by buildings in the square.

2.5 Algorithmic Insights

Independently, different groups (Braunschweig and Campinas) have combined methods from integer (IP) and linear programming (LP) with non-discrete geometry in order to obtain optimal solutions; first for the discrete case of vertex guards [17], but later also for general point guards [46,73].

The algorithm in [73] computes lower and upper bounds for the AGP, based on computing finite set cover instances with the help of a state-of-the-art IP solver. To generate a lower bound, a finite set of witness candidates is chosen and a restricted AGP is solved, in which only the witnesses have to be covered. For this, it suffices to extract a finite set of potential guard positions from the visibility arrangement of the witness set to ensure optimality. Similarly, finite sets of potential witness positions for a given finite guard set can be extracted from the visibility arrangement of the guards. This allows it to compute upper and lower bounds for the optimal AGP value by solving discrete set cover instances. The algorithm in [73] iterates between generating tighter lower and upper bounds by refining the witness and guard candidate sets along the iterations. It stops when lower and upper bounds coincide. Although no proof of theoretical convergence is known (and the work by Abrahamsen et al. [1] strongly suggests that no such convergence can exist for all classes of instances), in tests, the approach is able to yield optimal solutions for a large variety of instance classes, even for polygons with up to a thousand vertices.

An approach presented in [46] considers a similar primal-dual scheme, but focuses on the linear relaxation of the primal guard cover with guard set G , from which a small subset has to be selected to cover all points from a witness set W : For each point $w \in W$, a guard in its visibility region $\mathcal{V}(w)$ must be chosen. Allowing fractional guards corresponds to admitting guard variables $0 \leq x_g \leq 1$ for any guard $g \in G$. This yields the following linear program.

$$\min \sum_{g \in G} x_g \tag{1}$$

$$\text{s. t. } \sum_{g \in G \cap \mathcal{V}(w)} x_g \geq 1 \quad \forall w \in W \tag{2}$$

$$0 \leq x_g \leq 1 \quad \forall g \in G \tag{3}$$

Its dual is the witness packing problem, in which the objective is to find as many independent witness positions $w \in W$ as possible, such that no two of them can be seen from the same guard position $g \in G$.

$$\max \sum_{w \in W} y_w \tag{4}$$

$$\text{s. t. } \sum_{g \in G \cap \mathcal{V}(w)} y_w \leq 1 \quad \forall g \in G \tag{5}$$

$$0 \leq y_w \leq 1 \quad \forall w \in W \tag{6}$$

Because of strong duality of linear programming, considering these fractional guard and witness values leads to optimal primal and dual solutions with identical objective values. To eliminate fractional solutions, we can apply appropriate cutting planes derived from the set cover polytope, based on specific subsets $J_1 \cap G$ and $J_2 \cap G$.

$$\sum_{g \in J_2 \cap G} 2x_g + \sum_{g \in J_1 \cap G} x_g \geq 2 \quad (7)$$

As it turns out [24], only a small subset of these inequalities matter in the context of AGP instances. Together with a similar primal-dual iteration scheme such as the one in [73], we can find optimal integral solutions for a large range of benchmark instances, including the one shown in the scenario above; see Fig. 3 for a pair of primal and dual solutions.

2.6 Extensions

There are various extensions and related questions. In the work described above, the robot has unlimited viewing distance, only the number of scans is to be minimized, and the given region is known in advance. We have also studied this problem for the case of limited viewing distance and an objective function that is a linear combination of the number of scans and distance traveled by the robot. See [29] and the cited related work. We have also studied the context of exploring an unknown region by a robot with discrete vision; see [31].

2.7 Acknowledgments

The content of this section is based on the abstract [10] and paraphrases the joint work with Dorit Borrmann, Pedro de Rezende, Cid de Souza, Stephan Friedrichs, Alexander Kröller, Andreas Nüchter and Christiane Schmidt contained in the papers [24, 46]. For a visualization, see the video that accompanies [10], to be found at the website <http://www.computational-geometry.org/SoCG-videos/socg13video/#Borrmann-etal>.

3 Online Exploration and Triangulation by a Swarm of Simple Robots

3.1 Motivation

Consider a swarm of inexpensive robots without explicit mapping capabilities in an unknown area. Each robot has a limited visibility range, but can move around to get a more complete picture of the environment. Once the region has been fully covered, the robots can also stay around so that we can get live updates. How can we organize this exploration and make sure that we can continue to observe all parts of the environment after they are discovered?

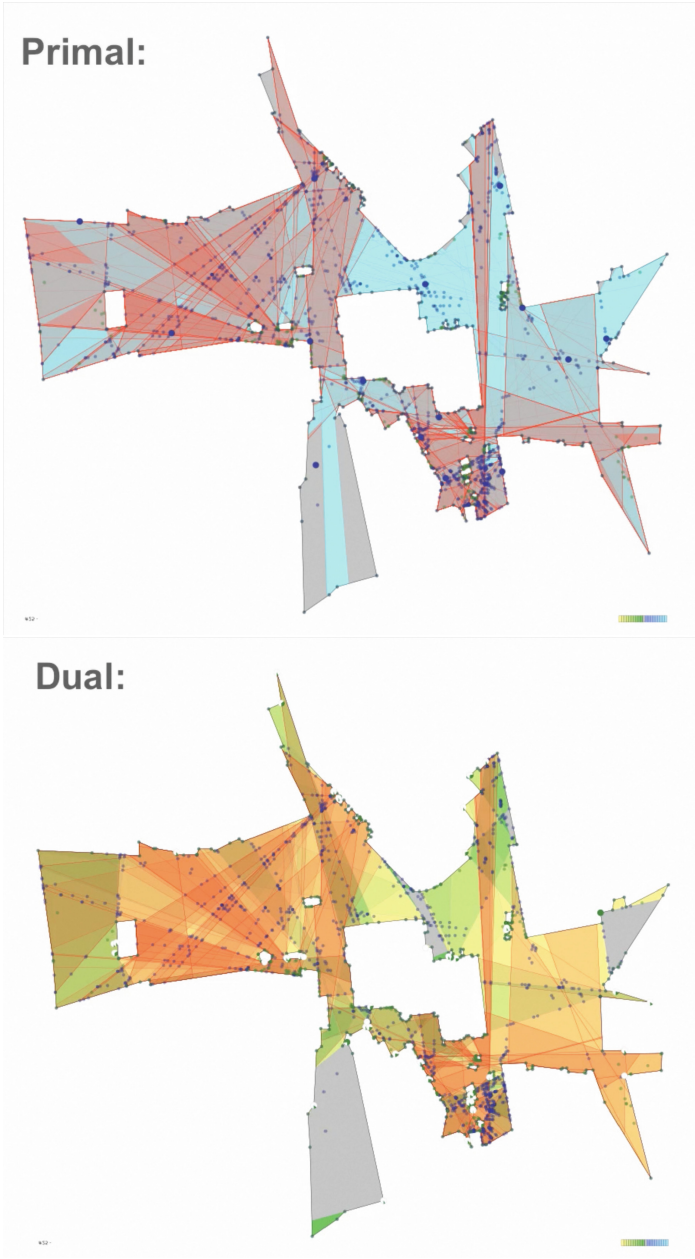


Fig. 3. A pair of primal and dual solutions to the fractional linear programming relaxation.

3.2 Formal Aspects

We are given a polygonal region P , and a point $z \in P$ on the boundary of P . In addition, we are given a supply of robots with limited (circular) communication range r ; for ease of description, we normalize to $r = 1$. Within this range, perception of and communication with other robots is possible. In the Minimum Relay Triangulation Problem (MRTP), the goal is to compute a set R of robot positions within P (with $z \in R$ and $V \subseteq R$ for the vertex set V of P), such that there is a (unit) triangulation of P whose vertex set is exactly the set R and whose edges stay within P and have length at most 1. The objective is to minimize the number of robots. In the Maximum Area Triangulation Problem (MATP), the number of available robots is bounded by a number k ; the goal is to determine a set R of at most k robot positions, with a unit triangulation covering a maximum possible area. For the online versions (OMRTP and OMATP), the polygon P is unknown. Each robot may move through the area, and has to decide on a new location for a triangulation vertex, while still being within reach of the previously placed relays. Once it has stopped, it becomes part of the static triangulation, allowing other relays to extend the exploration.

3.3 Context

In recent years, the field of robotics has seen two diverging trends. One has been to achieve progress by increasing the capabilities of individual robots, keeping the cost of state-of-the-art machines relatively high. An opposite direction has been to develop simpler and cheaper platforms, at the expense of reducing the capabilities per robot. The latter raises new challenges for developing new principles and algorithms, such as coordinating many robots with limited capabilities into a swarm that can carry out difficult tasks, such as exploration, surveillance, and guidance.

3.4 Application Scenario

A real-life example of an advanced, low-cost, swarm robot design with limited sensor capabilities is the r-one [61], shown in Fig. 4. Its estimated unit cost is about US \$250. Measuring only 11 cm in diameter, it has a 32-bit ARM-based microcontroller, running at 50 MHz with no floating point unit. The local infrared (IR) communication system is used for inter-robot communication and localization. Each robot has eight IR transmitters and eight receivers. The transmitters broadcast in unison and emit a radially uniform energy pattern. The robot's eight IR receivers are radially spaced to produce 16 distinct detection regions (shown in Fig. 4 (Right)). By monitoring the overlapping regions, the bearing of neighbors can be estimated to within $\approx \pi/8$. Thus, it has limited capabilities for measurement, which is intertwined with local communication. The IR receivers have a maximum bit rate of 1250 bits per second. Each robot transmits $(\Delta + 1)$ 4-byte messages during each round, one being a system announce message, the others containing the bearing measurements to that robot's neighbors.

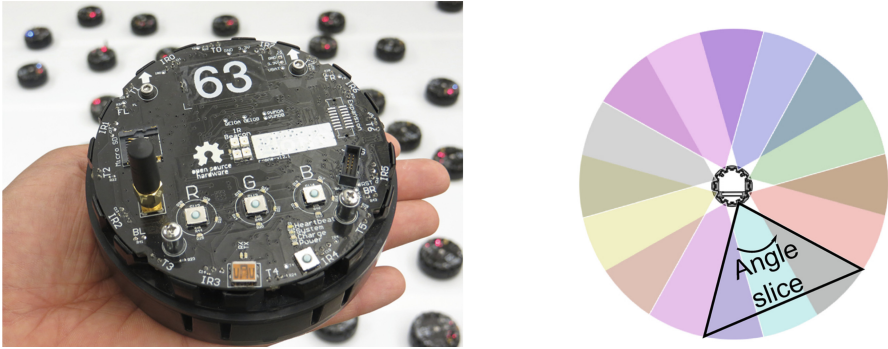


Fig. 4. (Left) the r-one for multi-robot research, designed by the MRSL group at Rice University. (Right) IR receiver detection regions. Each receiver detects an overlapping 68° , allowing to determine angles within about 22.5° .

The system supports a maximum of $\Delta = 10$. For more on experimental work on coordination and navigation of r-ones, see [61, 62].

The algorithmic challenge is to exploit the capabilities of a swarm to overcome the limitations of the individual robots, and achieve overall behavior with provable performance guarantees that are rooted in solid algorithmic theory.

3.5 Algorithmic Insights

The problems MRTP and MATP were introduced in [26]; the currently best results for the online versions OMRTP and OMATP were presented in [25, 68]. Both problems share their decision problem, which is known to be NP-hard. For the OMRTP, there is a lower bound of $6/5$ on the competitive factor of any deterministic strategy, as well as a 3-competitive algorithm for general polygons. This strategy is shown in Fig. 5: We place robots at unit intervals along the boundary (green) and fill the interior with a regular triangular grid (blue). The space between the two is patched together using a third class (red). One can prove that the size of each of the three classes is bounded by the number of robots in an optimal solution. For polyominoes, algorithms with better competitive factors exist [30].

On the other hand, the OMATP does not admit a deterministic strategy with a constant competitive factor, if the polygon may have small corridors. If these can be excluded, greedy strategies perform well [30].

These strategies have been used on the real robots described above; see Fig. 6 for a snapshot.

3.6 Extensions

Once a well-formed triangulation (with lower bounds on minimum edge length and minimum internal angle) is established, it can be employed for a number

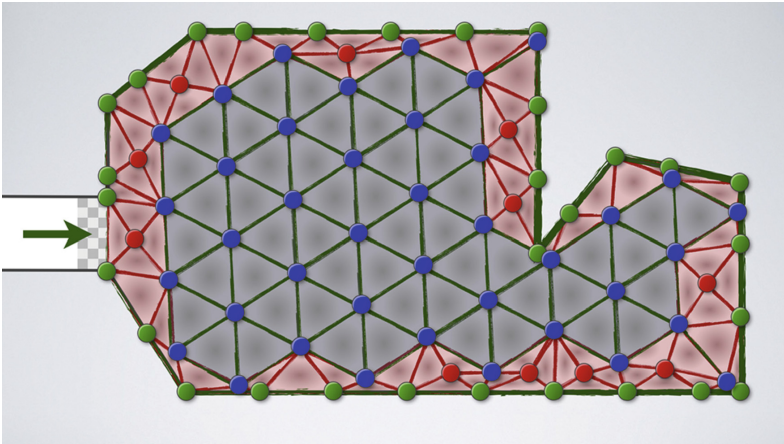


Fig. 5. The 3-competitive strategy for the OM RTP, consisting of three sets of robots: green robots are placed along the boundary at vertices and at unit distance along edges; blue robots fill the interior by a grid; red robots complete the triangulation by connecting boundary interior robots. (Color figure online)

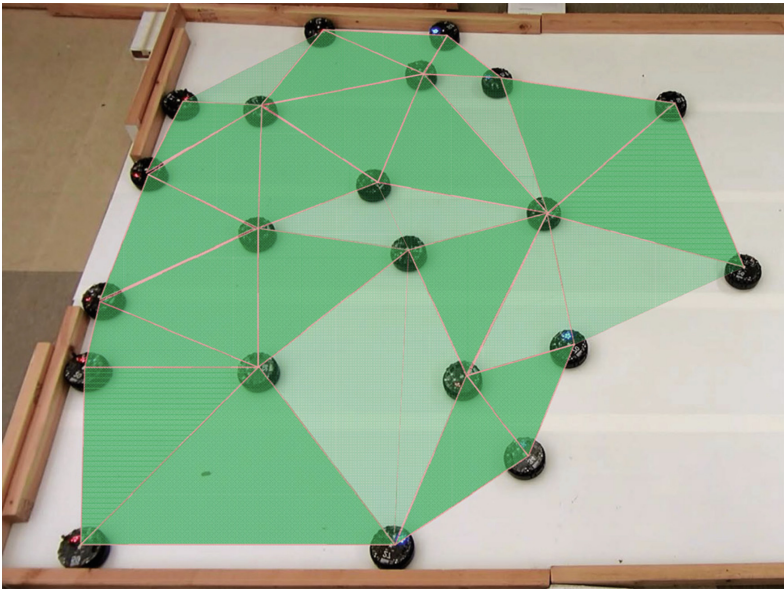


Fig. 6. A swarm of r-ones executing the online algorithm for the OM ATP in the real world.

of different purposes. In [53], we show how the underlying connectivity graph of the robots forming the triangulation can be used for maintaining the corresponding dual graph (in which vertices correspond to triangles, and edges represent triangles sharing a common boundary), and how a minimum hop count in the unweighted dual graph achieves a constant stretch factor compared to the shortest geometric distances, i.e., manages to stay within a constant factor of the shortest achievable distance with global information; see Fig. 7.

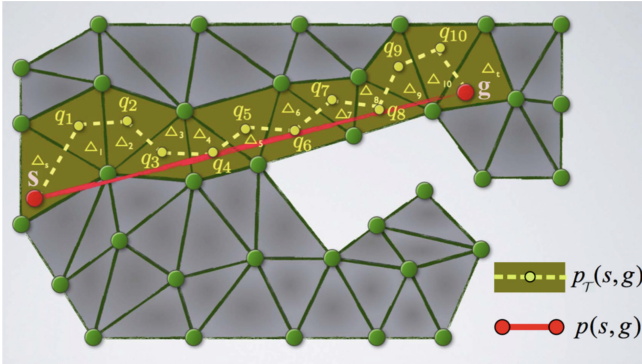


Fig. 7. Using a dual path for routing in a triangulated environment: a shortest path (shown in red) is approximated by a minimum-hop path (shown in yellow), achieving constant stretch. (Color figure online)

Another application of a triangulation is to use it for surveying the underlying region by additional, mobile robots, again based on the dual graph of the stationary triangulation. In [2, 54, 57], we discuss various aspects of local policies for patrolling the vertices of such a graph.

3.7 Acknowledgments

The content of this section is based on the abstract [28] and paraphrases the joint work with Aaron Becker, Tom Kamphans, Alexander Krölller, Seoung Kyou Lee, James McLurkin, Joe Mitchell, Christiane Schmidt, described in the papers [25, 52, 54]. For a visualization, see the video <http://www.computational-geometry.org/SoCG-videos/socg13video/#Becker-et-al> that accompanies [28]. See the journal paper [54] for full technical details of the robotics side and its extensions.

4 Distributed Cohesive Control

4.1 Motivation

Consider a swarm of robots that needs to remain connected. There is no central control and no knowledge of the overall environment. This environment is

hostile: The swarm is being pulled apart by external forces, stretching it into a number of different directions, so it is in danger of breaking up. Individual robots are weak, with limited sensing, limited communication, and limited connectivity; even worse, each robot’s expected lifetime is limited by random, permanent failures, which may destroy connectedness and functioning of the swarm as a whole. How can we achieve coordinated dynamic swarm behavior without centralized coordination? How can we employ each robot as much as possible, without depending on it if it fails? How can we balance overall flexibility and robustness to deal with the hostile environment?

The challenge is to develop local self-stabilizing mechanisms that allow the swarm to stay locally well connected (forcing swarm members to stay close to each other), even when it is being pulled apart by several distant and mobile sites (forcing swarm members to spread out).

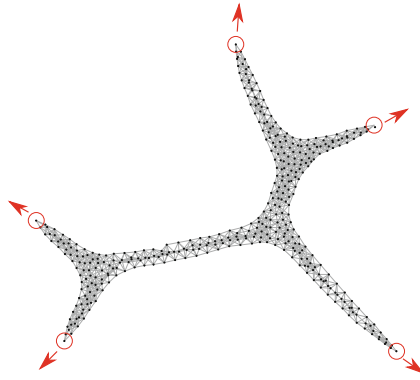


Fig. 8. A robust robot swarm emulating a Steiner tree between five diverging leader robots.

4.2 Formal Aspects

We consider a finite set of robots \mathcal{R} with an externally controlled subset of *leader robots* $\mathcal{L} \subsetneq \mathcal{R}$, $|\mathcal{L}| \ll |\mathcal{R}|$. We want the remaining robots $\mathcal{R} \setminus \mathcal{L}$ to maintain a dynamic and robust network that keeps the swarm connected, even in the presence of random robot failures and arbitrary leader movements. Thus, the overall shape of the swarm should form a “thick” Steiner tree among the leaders with the robots $\mathcal{R} \setminus \mathcal{L}$ evenly distributed along the edges, as shown in Fig. 8.

Robots have the shape of circles; two of them are connected when within a maximum distance and with an unobstructed line of sight. Robots know the relative positions and orientations of their neighbors and can communicate asynchronously. Each robot has a unique ID; leader IDs are known by all others. Robot’s translations and rotations are limited in velocity and acceleration. Communication is possible by broadcasting to immediate neighbors.

The perception of all robots is local; however, due to the known position and orientation difference, each robot can transform vectors of its neighbors to its own coordinate system. We avoid multi-hop transformations to keep errors small.

4.3 Context

One of the earliest works on flocking is Reynold's pioneering work [66]. In recent years, a considerable number of aspects and objectives have extended this perspective. We highlight only some of the ensuing papers, showing how they differ from our perspective.

A basic component of flocking is volumetric control, as it was presented by Spears [71]: Robots use local potential field controllers (with attractive and repulsive forces) for constructing a regular lattice with a corresponding base density [40, 63]. This does not necessarily preserve connectivity [3, 36, 71]. While the latter can be side-stepped by simply assuming that robots are always connected [70], we aim for connectivity as a requirement, which is vital in a fully distributed setting in which deterministic recovery from disconnectedness may be impossible.

Some of the ideas of Olfati-Saber [63] form the basis of our work; however, in that model, robots do utilize global information, e.g., the position of a guide robot in a shared coordinate frame [14, 49, 50, 63] or environmental potential [34]. Instead of the potentials, Cortes et al. [16] and Lindhé et al. [56] used Voronoi tessellation. This is based on a density function, requiring global information for covering a region. Overall, this differs from our objective of developing methods that are fully distributed, aiming for collective mechanisms for complex group behavior that go beyond relatively simple objectives [9], but also for systems that are robust against partial hardware failures [43].

The final property is *cohesiveness* of the overall swarm: all robots should maintain a unified state, such as desired distance or orientation; see [63] for a formal definition. As described in [60], detecting and maintaining a swarm boundary is of particular importance for maintaining swarm cohesiveness and connectedness. This is based on and related to work in the field of wireless sensor networks (WSNs), which has considered many geometric settings in which a large swarm of stationary nodes is faced with the task of achieving a large-scale overall goal, while the individual components can only operate locally, based on limited individual capabilities and information; refer to Fekete et al. [27, 47] for a detailed description. In addition to the work on swarm robotics described above, there is a large body of theoretical work on geometric swarm behavior; here we only mention Chazelle [13] for flocking behavior, and Fekete et al. [27, 47] for geometric algorithms for static sensor networks, including distributed boundary detection.

Beyond the involved properties and paradigm, the overall goal for the swarm can also be described as a distributed optimization problem: Maintain a generalized Steiner tree with limited edge lengths that connects a moving set of terminals. To the best of our knowledge, only Hamann and Wörn [35] have

explicitly considered the construction of Steiner trees by a robot swarm. For static terminals, they start with an exploratory network; as soon as all terminals are connected, only best paths are kept and locally optimized.

Even in a centralized and static setting with full information, we must deal with a generalization of the well-known NP-hard problem of finding a good Steiner tree [33]; see the books by Hwang et al. [41] and Prömel and Steger [65] for further introduction. More specifically, we are faced with the relay placement problem: The input is a set of sensors and a number $r \geq 1$, the communication range of a relay. The objective is to place a minimum number of relays so that between every pair of sensors is connected by a path through sensors and/or relays. The best known theoretical performance bound for this NP-hard problem was given by Efrat et al. [19], who presented a 3.11-approximation algorithm; they also showed a worst-case lower bound of 3 for a large class of approximation algorithms. For a fixed number of available relays, this turns into our problem of maximizing the achievable networks size, with matching approximation factor.

4.4 Algorithmic Insights

A key insight is that achieving complex overall behavior can be based not only on local interaction that resembled physical forces, but also on principles of distributed algorithms that build more complex structures. To this end, we have developed a number of powerful local mechanisms for maintaining a dynamic swarm of robots with limited capabilities and information, in the presence of external forces and permanent node failures. These mechanisms consist of a set of local, self-stabilizing, *continuous* algorithms that together produce a generalization of a Euclidean Steiner tree, maintain a dynamic and robust network between leader robots. At any stage, the resulting overall shape achieves a good compromise between local thickness, global connectivity, and flexibility to further continuous motion of the terminals, adopting the directions of multiple leaders, while preserving a uniform thickness along the edges of the Steiner tree. The resulting swarm behavior scales well, is robust against node failures, and performs close to the best known approximation bound for a corresponding centralized static optimization problem.

We first sketch the base behavior of the robots, inducing an almost convex swarm shape. This is subsequently improved by leader forces, and stability improvement and thickness contraction.

4.4.1 Base Behavior

Our base behavior consists of three components that result in a swarm shape of a droplet. (i) The *flocking algorithm* of Olfati-Saber [63] considers regular distribution and movement consensus. The algorithm is a stateless equation based on potential fields and is proven to converge. It uses three rules: Attraction to neighbors, repulsion from too close neighbors, and adaption to the velocity of neighbors. We slightly modified the algorithm for better response to additional forces. (ii) An extended version of the *boundary detection* algorithm of McLurkin

and Demaine [60], which determines if a robot lies on the boundary and also identifies small holes by using the average angle. (iii) The *boundary tension* of Lee and McLurkin [55], which straightens and minimizes the boundary of the swarm. This is done by simply pushing boundary robots to the middle of its two boundary neighbors.

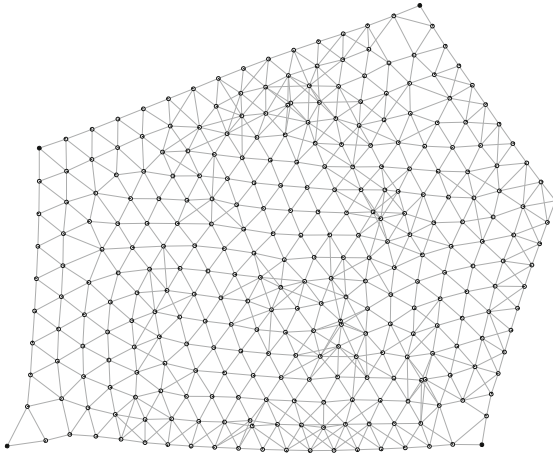


Fig. 9. A swarm configuration in which a purely physics-based mechanism lead to disconnection.

The base behavior without any other forces results in at most convex shapes before losing connectivity. Figure 9 shows a situation in which the swarm is about to lose connectivity. For stronger control and more variable shapes, leader forces are introduced.

4.4.2 Leader Forces

A single leader constitutes the simplest form of swarm control. In this case the swarm motion is determined by the leader's velocity. With multiple (possibly antagonistic) leaders, the swarm is not just steered, but may be stretched to the limit until connectivity is lost. To this end, each robot needs to find an appropriate balance between the influence of different leaders. See top of Fig. 10 for an illustration. We therefore combine both methods by a smooth transition between velocity matching close to the leaders and leader pursuit when further away; see the bottom of Fig. 10.

4.4.3 Stability Improvement and Thickness

Near Steiner points, connections along concave swarm boundaries may be stretched by boundary forces. When the involved edges approach the upper bound for communication, connections may be disrupted, to the point where the

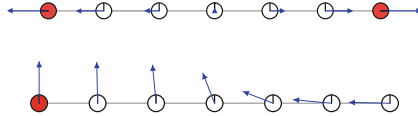


Fig. 10. (Top) A one-dimensional scenario with two leaders (red) moving in opposite directions. (Bottom) With increasing distance to the leader, the effect shifts from velocity matching to leader pursuit. (Color figure online)

swarm loses connectivity. By adding a thickness-dependent compression force, we reduce neighbor distances without influencing the Steiner-tree shape of the swarm; in effect, this works similar to compression stockings. Algorithmically, the involved mechanisms resemble methods that have been studied in the context of sensor networks, such as local methods for boundary detection and hop distance from the boundary. This gives rise to notions such as the hop circle of radius h with robot r as circle center: This is the set of all robots with a hop count $\leq h$ to c ; only robots with hop distance at least h may be on the boundary, so a hop circle of maximal radius around a given robot gives an indication of the local thickness in its neighborhood. (For an example, see Fig. 11.)

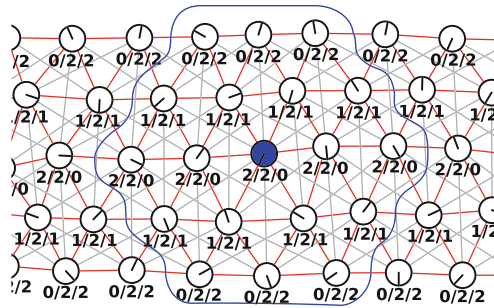


Fig. 11. Thickness determination for a limb part of a swarm. The indicated triples of numbers at each node r correspond to $(b(r)/t(r)/h(r))$, where $b(r)$ is the hop count from the boundary, $t(r)$ is the local thickness, and $h(r)$ is the circle center distance. A largest hop circle is marked in blue. (Color figure online)

This allows local approaches to keep track of and maintain local thickness and connectivity, even in the presence of external forces and robot failures. See our paper [48] for more technical details.

4.5 Extensions

There are numerous possible extensions, most notably for dealing with a cohesive swarm in the presence of obstacles. At this time, these are still under development.



Fig. 12. A comparison of strategies for the same example, for a swarm with $n = 400$ and failure rate 0. As indicated, columns correspond to strategies BASE, LEADER, and ALL. Rows show the swarms at times $T = 200$, $T = 2000$, $T = 3000$, $T = 7600$, $T = 12,000$, with 60 steps per simulated second. When a swarm is no longer shown, it has become disconnected right after the previous time step.

4.6 Acknowledgments

The content of this section is based on the abstract [22] and paraphrases the joint work with Maximilian Ernestus, Michael Hemmer and Dominik Krupke contained in the paper [48] and the thesis [23] of Maximilian Ernestus and Dominik Krupke.

5 Controlling Swarms by Global Forces

5.1 Motivation

One of the exciting new directions of robotics is the design and development of micro- and nanorobot systems, with the goal of letting a huge population of robots perform complex operations in a complicated environment. Due to scaling issues, individual control of the involved robots becomes physically impossible: While energy storage capacity drops with the third power of robot size, medium resistance decreases much slower. A possible answer lies in applying a global, external force to all particles in the swarm. This is what many current micro- and nanorobot systems with many robots do: The whole swarm is steered and directed by an external force that acts as a common control signal. These common control signals include global magnetic or electric fields, chemical gradients, and turning a light source on and off.

5.2 Formal Aspects

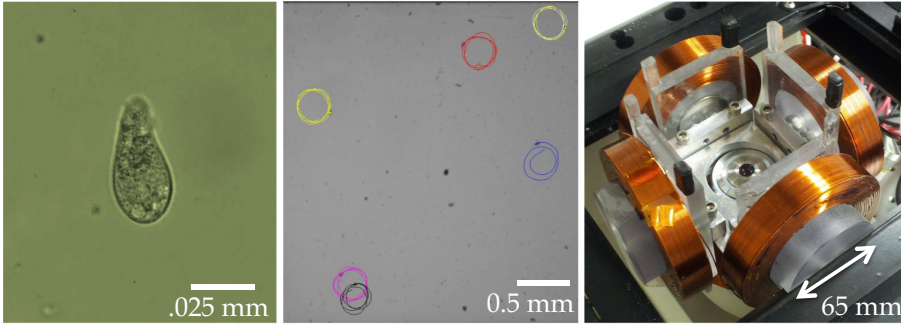
We consider a two-dimensional grid world, with some cells occupied and others free. Initially, the planar square grid is filled with some unit-square particles (each occupying a cell of the grid) and some fixed unit-square blocks. All particles are commanded in unison: a valid command is “Go Up” (u), “Go Right” (r), “Go Down” (d), or “Go Left” (l). All particles move in the commanded direction until they hit an obstacle or another particle. A representative command sequence is $\langle u, r, d, l, d, r, u, \dots \rangle$. We call these global commands *force-field moves*. We assume that we can bound the minimum particle speed and that we can guarantee that all particles have moved to their maximum extent.

5.3 Application Scenario

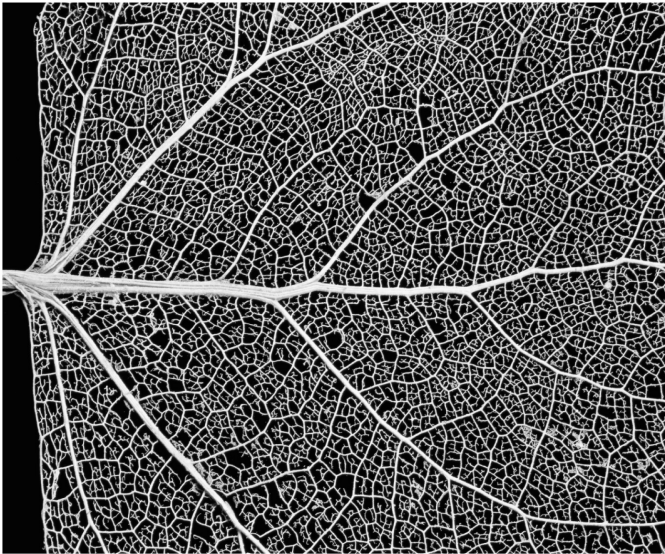
Becker et al. [8] demonstrate how to apply a magnetic field to simultaneously move cells containing iron particles in a specific direction within a fabricated workspace; see Fig. 13a. Other recent examples include using the global magnetic field from an MRI to guide magneto-tactic bacteria through a vascular network to deliver payloads at specific locations [12], and using electromagnets to steer a magneto-tactic bacterium through a micro-fabricated maze [45]; however, this still involves only individual particles at a time, not the parallel motion of a whole, massive swarm. How can we manipulate the overall swarm with coarse global control, such that individual particles arrive at multiple different destinations in a (known) complex vascular network such as the one in Fig. 13b?

5.4 Context

The problem resembles the logic puzzle *Tilt* [72], and dexterity ball-in-a-maze puzzles such as *Pigs in Clover* and *Labyrinth*, which involve tilting a board to cause all mobile pieces to roll or slide in a desired direction. Problems of this type



(a) (Left) After feeding iron particles to ciliate eukaryon (*Tetrahymena pyriformis*) and magnetizing the particles with a permanent magnet, the cells can be turned by changing the orientation of an external magnetic field (see colored paths in the center image). (Right) Using two orthogonal Helmholtz electromagnets, Becker et al. [8] demonstrated steering many living magnetized *T. pyriformis* cells. All cells are steered by the same global field.



(b) Biological vascular network (cottonwood leaf). (Photo: Royce Bair/Flickr/Getty Images.) Given such a network along with initial and goal positions of N particles, is it possible to bring each particle to its goal position using a global control signal? Note that this arrangement is *not* a tree, but a graph structure with many cycles. MATLAB code for driving N particles through this network is available at <http://www.mathworks.com/matlabcentral/fileexchange/42892>.

Fig. 13. (Top) State of the art in controlling small objects by force fields. (Bottom) A complex vascular network, forming a typical environment for the parallel navigation of small objects. This section investigates parallel navigation in discretized 2D environments. (Color figure online)

are also similar to sliding-block puzzles with fixed obstacles [18, 37–39], except that all particles receive the same control inputs, as in the Tilt puzzle. Another connection is to *Randolph’s Ricochet Robots* [21], a game that allows individual and independent control of the involved particles.

In the real world, driving ferromagnetic particles with a magnetic resonance imaging (MRI) scanner gives examples of this challenge, from nano- to micro-scales; see [74].

5.5 Algorithmic Insights

Clearly, having only one global signal that uniformly affects all robots at once poses a strong restriction on the ability of the swarm to perform complex operations. The only hope for breaking symmetry is to use interactions between the robot swarm and obstacles in the environment. The key challenge is to establish if interactions with obstacles are sufficient to perform complex operations, ideally by analyzing the complexity of possible logical operations.

It is important to note that there are two fundamentally different classes of algorithmic problems, which we denote by *External Computation* and *Internal Computation*.

5.5.1 External Computation

Considering the particle swarm as input for a given algorithmic problem, we are faced with a number of questions that need to be resolved by a computing device “outside” of the particle system, such as the following.

1. Given a map of an environment, such as the vascular network shown in Fig. 13b, along with initial and goal positions for each particle, does there *exist* a sequence of inputs that will bring each particle to its goal position?
2. Given a map of an environment, such as the vascular network shown in Fig. 13b, along with initial and goal positions for each particle, what is the *shortest* sequence of moves that will bring each particle to its goal position?
3. Given initial and goal positions for each particle in a swarm, how can we *design* a set of obstacles and a sequence of moves, such that each particle reaches its goal position?

Deliberate use of existing stationary obstacles leads to a wide range of possible particle configurations. In our work [4–6, 69], we address all these issues. For the first two questions, we show that they may lead to computationally difficult situations. We also develop several positive results for the third question. The underlying idea is to construct artificial obstacles (such as walls) that allow arbitrary rearrangements of a given two-dimensional particle swarm.

Theorem 1. *Given a specified goal location and an initial configuration of movable particles and fixed obstacles, it is NP-hard to decide if a move sequence exists that ends with some particle at the goal location.*

The proof relies on a reduction from 3SAT. Suppose we are given n Boolean variables x_1, x_2, \dots, x_n , and m disjunctive clauses $C_j = U_j \vee V_j \vee W_j$, where each literal U_j, V_j, W_j is of the form x_i or $\neg x_i$. We construct a problem instance that has a solution if and only if all clauses can be satisfied by a truth assignment to the variables. This instance is composed of *variable gadgets* for setting individual variables TRUE or FALSE, *clause gadgets* that construct the logical OR of groupings of three variables, and a *check gadget* that constructs the logical AND of all the clauses. A particle is only delivered to the goal location if the variables have been set in such a way that the formula evaluates to TRUE. See Fig. 14 for an overview of the whole construction.

On the positive side, we can show that for a given labeled arrangement of particles, arbitrary permutations can be achieved with appropriate sets of obstacles. To this end, we consider a 2D array of particles, as shown in Fig. 15. For an $a_r \times a_c$ matrix A and a $b_r \times b_c$ matrix B , of equal total size $N = a_r a_c = b_r b_c$, a *matrix permutation* assigns each element in A a unique position in B .

Theorem 2. *Let A and B be matrices with dimensions as above. Any matrix permutation that transforms A into B can be executed by a set of obstacles in just four moves. For N particles, the constructed arrangement of obstacles requires $(3N + 1)^2$ space and $4N + 1$ obstacles. If particles move with a speed of v , the required time for those four moves is $12N/v$.*

This can be employed to realize larger sets of permutations all at once, as shown in Fig. 16.

The previous construction is efficient with respect to the number of required moves, at the expense of a possibly higher number of obstacles. By allowing a larger number of moves, we can limit the number of obstacles for achieving *any* permutation, as shown in Fig. 17.

Theorem 3. *We can construct a set of $O(N)$ obstacles such that any $a_r \times a_c$ arrangement of N particles can be rearranged into any other $a_r \times a_c$ arrangement π of the same particles, using at most $O(N^2)$ force-field moves.*

Proof. See Fig. 17. Use Theorem 2 to build two sets of obstacles, one each for p and q , such that p is realized by the sequence $\langle u, r, d, \ell \rangle$ (clockwise) and q is realized by $\langle r, u, \ell, d \rangle$ (counterclockwise). Then we use the appropriate sequence for generating π in $O(N^2)$ moves.

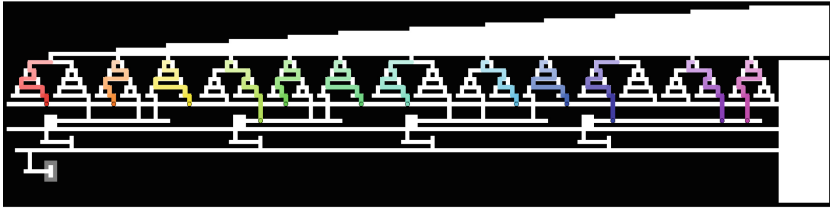
On the other hand, minimizing the number of moves for achieving a desired goal configuration for *all* particles turns out to be PSPACE-complete.

Theorem 4. *Given an initial configuration of (labeled) movable particles and fixed obstacles, it is PSPACE-complete to compute a shortest sequence of force-field moves to achieve another (labeled) configuration.*

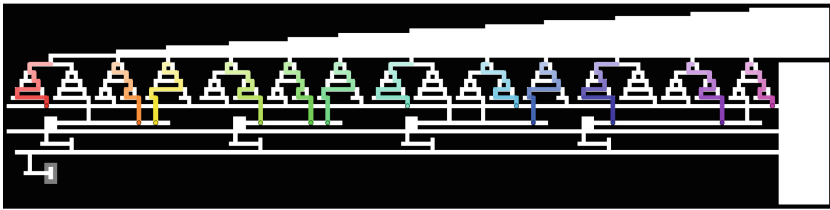
The proof is largely based on a complexity result by Jerrum [42], who considered the following problem: Given a permutation group specified by a set of generators and a single target permutation π , which is a member of the group,



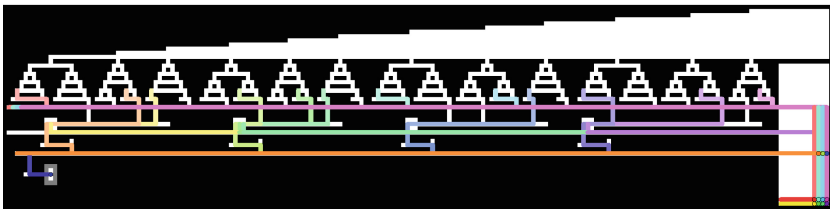
(a) Initial state with particles (colored) on the upper right. The objective is to move one particle into the grey target rectangle at lower left.



(b) Setting variables to (FALSE, TRUE, FALSE, TRUE) does not satisfy this 3SAT instance.



(c) Setting the variables (TRUE, FALSE, FALSE, TRUE) satisfies this 3SAT instance.



(d) Successful outcome. (TRUE, FALSE, FALSE, TRUE) moves a single particle into the target region.

Fig. 14. Combining twelve variable gadgets, four 3-input OR gadgets, and a 4-input AND gadget to realize the 3SAT expression $(\neg x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_3)$. (Color figure online)

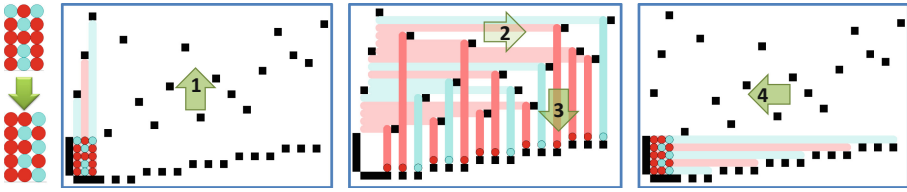


Fig. 15. In this image for $N = 15$, black cells are obstacles, white cells are free, and colored discs are individual particles. The world has been designed to permute the particles between ‘A’ into ‘B’ every four steps: $\langle u, r, d, \ell \rangle$. See the video at <http://youtu.be/3tJdRrNShXM>. Visually, the distinction between particles of the same color does not matter; however, the arrangement of obstacles induces a specific permutation of individual particles. (Color figure online)

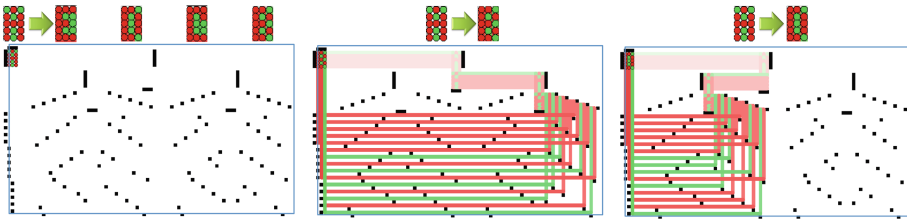


Fig. 16. For any set of k fixed, but arbitrary permutations of N particles, we can construct a set of $O(kN)$ obstacles, such that we can switch from a start arrangement into any of the k permutations using at most $O(\log k)$ force-field moves. Here $k = 4$ and ‘A’ is transformed into ‘B’, ‘C’, ‘D’, or ‘E’ in eight moves: $\langle r, d, (r/\ell), d, (r/\ell), d, \ell, u \rangle$.

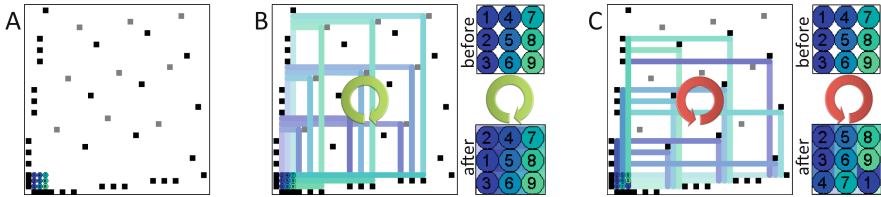


Fig. 17. Repeated application of two base permutations can generate any permutation, when used in a manner similar to BUBBLE SORT. The obstacles in (A) generate the base permutation $p = (1, 2)$ in the clockwise direction $\langle u, r, d, \ell \rangle$ (B) and $q = (1, 2, \dots, N)$ in the counterclockwise direction $\langle r, u, \ell, d \rangle$ (C).

what is the shortest expression for the target permutation in terms of the generator? This problem was shown to be PSPACE-complete in [42], even when the generator set consists of only two permutations π_1 and π_2 . Combining this with the idea of our construction of Theorem 3 yields the claimed result: Use two sets of obstacles for realizing π_1 by a sequence of four clockwise moves and π_2 by a

sequence of four counterclockwise moves; then a shortest sequence of force-field moves for achieving a desired target permutation π_3 corresponds to a minimum generation of π_3 by π_1 and π_2 .

5.5.2 Internal Computation

Considering the particle swarm as a complex system that can be reconfigured in various ways, we are faced with issues of the computational power of the swarm itself (as opposed to that of an external device), such as the following.

1. Can the complexity of particle interaction be exploited to model logical operations?
2. Are there limits to the computational power of the particle swarm?
3. How can we achieve computational universality with particle computation?

In [4, 5, 69], we give precise answers to all of these questions. In particular, we show that the logical operations AND, NAND, NOR, and OR can be implemented in our model using dual-rail logic. Using terminology from electrical engineering, we call these components that calculate logical operations *gates*. We establish a fundamental limitation for particle interactions: We cannot duplicate the output of a chain of gates without also duplicating the chain of gates. This means that a so-called FAN-OUT gate cannot be generated. We resolve this missing component with the help of 2×1 particles, which can be used to create FAN-OUT gates that produce multiple copies of the inputs without needing duplicate gates, as shown in Fig. 18 for a physical prototype. Using these FAN-OUT gates, we provide rules for replicating arbitrary digital circuits, allowing us to establish the full range of computational universality as presented by complex digital circuits.

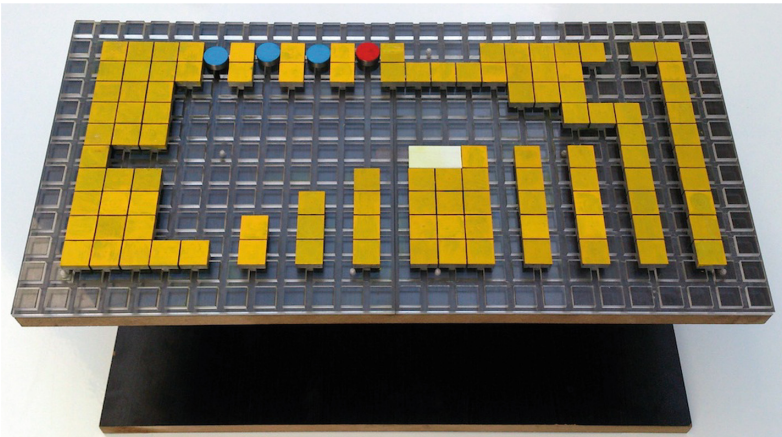


Fig. 18. Gravity-fed hardware implementation of particle computation. The reconfigurable prototype is set up as a FAN-OUT gate using a 2×1 robot (white)

5.6 Extensions

There are numerous extensions to the basic framework of control by uniform global forces. In [58] we develop methods for mapping, foraging, and coverage with a particle swarm. In [59] we show how to collect a particle swarm. In [44] we show how to use uniform global forces in combination with appropriate obstacles to efficiently sort and classify polyomino shapes.

In combination with “sticky” particles that bind together when brought into contact, we can use the basic setup to build production lines for assembling given shapes; see our paper [7] for a basic algorithmic and complexity analysis, and [67] for more efficient methods that proceed in a hierarchical fashion.

5.7 Acknowledgments

The content of this section is based on the abstract [6] and paraphrases the joint work with Aaron Becker, Erik Demaine, Golnaz Habibi, Jarrett Lonsford, James McLurkin, Hahmed Mohtasham Shad, Rose Morris-Wright contained in the papers [4, 5, 69]. For an animated visualization, see the video at <https://youtu.be/H6o9DTIfkn0>.

References

1. Abrahamsen, M., Adamaszek, A., Miltzow, T.: The art gallery problem is $\exists\mathbb{R}$ -complete. In: Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC), pp. 65–73 (2018)
2. Akash, A.K., Fekete, S.P., Lee, S.K., López-Ortiz, A., Maftuleac, D., McLurkin, J.: Lower bounds for graph exploration using local policies. *J. Graph Algorithms Appl.* **21**(3), 371–387 (2017)
3. Balch, T., Hybinette, M.: Social potentials for scalable multi-robot formations. In: Proceedings of the 17th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 73–80 (2000)
4. Becker, A., Demaine, E.D., Fekete, S.P., Habibi, G., McLurkin, J.: Reconfiguring massive particle swarms with limited, global control. In: Flocchini, P., Gao, J., Kranakis, E., Meyer auf der Heide, F. (eds.) *ALGOSENSORS 2013*. LNCS, vol. 8243, pp. 51–66. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-45346-5_5
5. Becker, A.T., Demaine, E.D., Fekete, S.P., Lonsford, J., Morris-Wright, R.: Particle computation: complexity, algorithms, and logic. *Natural Computing* (to appear)
6. Becker, A.T., Demaine, E.D., Fekete, S.P., Shad, S.H.M., Morris-Wright, R.: Tilt: the video. Designing worlds to control robot swarms with only global signals. In: Proceedings of the 31st International Symposium on Computational Geometry (SoCG), pp. 16–18 (2015). <https://youtu.be/H6o9DTIfkn0>
7. Becker, A.T., et al.: Tilt assembly: algorithms for micro-factories that build objects with uniform external forces. In: Proceedings of the 28th International Symposium on Algorithms and Computation (ISAAC 2017), pp. 11:1–11:13 (2017). Full version to appear in: *Algorithmica*

8. Becker, A.T., Ou, Y., Kim, P., Kim, M.J., Julius, A.: Feedback control of many magnetized: tetrahymena pyriformis cells by exploiting phase inhomogeneity. In: Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3317–3323 (2013)
9. Bonabeau, E., Meyer, C.: Swarm intelligence: a whole new way to think about business. *Harvard Bus. Rev.* **79**, 106–114 (2001)
10. Borrmann, D., et al.: Point guards and point clouds: solving general art gallery problems. In: Proceedings of the 29th Symposium on Computational Geometry (SoCG), pp. 347–348 (2013)
11. Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., Hertzberg, J.: Globally consistent 3D mapping with scan matching. *Robot. Auton. Syst.* **56**(2), 130–142 (2008)
12. Chanu, A., Felfoul, O., Beaudoin, G., Martel, S.: Adapting the clinical MRI software environment for real-time navigation of an endovascular untethered ferromagnetic bead for future endovascular interventions. *Magn. Reson. Med.* **59**(6), 1287–1297 (2008)
13. Chazelle, B.: The convergence of bird flocking. *J. ACM* **61**(4), 21 (2014)
14. Chuang, Y.-L., Huang, Y.R., D’Orsogna, M.R., Bertozzi, A.L.: Multi-vehicle flocking: scalability of cooperative control algorithms using pairwise potentials. In: Proceedings of the 24th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 2292–2299 (2007)
15. Chvátal, V.: A combinatorial theorem in plane geometry. *J. Comb. Theory Ser. B* **18**, 39–41 (1974)
16. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* **20**(2), 243–255 (2004)
17. Couto, M.C., de Souza, C.C., de Rezende, P.J.: An exact and efficient algorithm for the orthogonal art gallery problem. In: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), pp. 87–94 (2007)
18. Demaine, E.D., Demaine, M.L., O’Rourke, J.: PushPush and Push-1 are NP-hard in 2D. In: Proceedings of the 12th Canadian Conference on Computational Geometry (CCCG), pp. 211–219 (2000)
19. Efrat, A., Fekete, S.P., Mitchell, J.S., Polishchuk, V., Suomela, J.: Improved approximation algorithms for relay placement. *ACM Trans. Algorithms* **12**, 20:1–20:28 (2016)
20. Eidenbenz, S., Stamm, C., Widmayer, P.: Inapproximability results for guarding polygons and terrains. *Algorithmica* **31**(1), 79–113 (2001)
21. Engels, B., Kamphans, T.: Randolphys robot game is NP-hard. *Electron. Notes Discrete Math.* **25**, 49–53 (2006)
22. Ernestus, M., Fekete, S.P., Hemmer, M., Krupke, D.: Continuous geometric algorithms for robot swarms with multiple leaders. In: Proceedings of the 31st European Workshop on Computational Geometry (EuroCG), pp. 69–72 (2015)
23. Ernestus, M., Krupke, D.: Distributed, scalable algorithmic methods for swarms with multiple leader robots. Bachelor thesis, TU Braunschweig (2014)
24. Fekete, S.P., Friedrichs, S., Kröller, A., Schmidt, C.: Facets for art gallery problems. *Algorithmica* **73**, 411–440 (2015)
25. Fekete, S.P., Kamphans, T., Kröller, A., Mitchell, J.S.B., Schmidt, C.: Exploring and triangulating a region by a swarm of robots. In: Goldberg, L.A., Jansen, K., Ravi, R., Rolim, J.D.P. (eds.) APPROX/RANDOM -2011. LNCS, vol. 6845, pp. 206–217. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22935-0_18

26. Fekete, S.P., Kamphans, T., Kröller, A., Schmidt, C.: Robot swarms for exploration and triangulation of unknown environments. In: Proceedings of the 25th European Workshop on Computational Geometry, pp. 153–156 (2010)
27. Fekete, S.P., Kröller, A.: Geometry-based reasoning for a large sensor network. In: Proceedings of the 22nd Symposium on Computational Geometry (SoCG), pp. 475–476 (2006)
28. Fekete, S.P., Kröller, A., Lee, S., McLurkin, J., Schmidt, C.: Triangulating unknown environments using robot swarms. In: Proceedings of the 29th Symposium on Computational Geometry (SoCG), pp. 345–346 (2013)
29. Fekete, S.P., Mitchell, J.S.B., Schmidt, C.: Minimum covering with travel cost. *J. Comb. Optim.* **24**, 32–51 (2012)
30. Fekete, S.P., Rex, S., Schmidt, C.: Online exploration and triangulation in orthogonal polygonal regions. In: Ghosh, S.K., Tokuyama, T. (eds.) WALCOM 2013. LNCS, vol. 7748, pp. 29–40. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36065-7_5
31. Fekete, S.P., Schmidt, C.: Polygon exploration with time-discrete vision. *Comput. Geom.: Theory Appl.* **43**(2), 148–168 (2010)
32. Fisk, S.: A short proof of Chvátal’s watchman theorem. *J. Comb. Theory Ser. B* **24**, 374 (1978)
33. Garey, M.R., Graham, R.L., Johnson, D.S.: The complexity of computing Steiner minimal trees. *SIAM J. Appl. Math.* **32**(4), 835–859 (1977)
34. Gazi, V.: Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Trans. Robot.* **21**(6), 1208–1214 (2005)
35. Hamann, H., Wörn, H.: Aggregating robots compute: an adaptive heuristic for the euclidean steiner tree problem. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 447–456. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69134-1_44
36. Hayes, A.T., Dormiani-Tabatabaei, P.: Self-organized flocking with agent failure: off-line optimization and demonstration with real robots. In: Proceedings of the 19th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 3900–3905 (2002)
37. Hearn, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.* **343**(1–2), 72–96 (2005)
38. Hoffmann, M.: Motion planning amidst movable square blocks: Push-* is NP-hard. In: Proceedings of the 12th Canadian Conference on Computational Geometry (CCCG), pp. 205–210 (2000)
39. Holzer, M., Schwoon, S.: Assembling molecules in ATOMIX is hard. *Theor. Comput. Sci.* **313**(3), 447–462 (2004)
40. Howard, A., Mataric, M.J., Sukhatme, G.S.: Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In: Asama, H., Arai, T., Fukuda, T., Hasegawa, T. (eds.) Proceedings 6th International Symposium on Distributed Autonomous Robotics Systems (DARS), pp. 299–308. Springer, Tokyo (2002). https://doi.org/10.1007/978-4-431-65941-9_30
41. Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. *Annals of Discrete Mathematics*, vol. 53. Elsevier, Amsterdam (1992)
42. Jerrum, M.R.: The complexity of finding minimum-length generator sequences. *Theor. Comput. Sci.* **36**, 265–289 (1985)

43. Kamimura, A., Murata, S., Yoshida, E., Kurokawa, H., Tomita, K., Kokaji, S.: Self-reconfigurable modular robot-experiments on reconfiguration and locomotion. In: Proceedings of the 14th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 606–612 (2001)
44. Keldenich, P., et al.: On designing 2D discrete workspaces to sort or classify 2D polyominoes. In: Proceedings of the 35th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA) (2018, to appear)
45. Khalil, I.S.M., Pichel, M.P., Reefman, B.A., Sukas, O.S., Abelmann, L., Misra, S.: Control of magnetotactic bacterium in a micro-fabricated maze. In: Proceedings of the 30th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 5488–5493 (2013)
46. Kröllner, A., Baumgartner, T., Fekete, S.P., Schmidt, C.: Exact solutions and bounds for general art gallery problems. *J. Exp. Algorithms* **17**, 2–3 (2012)
47. Kröllner, A., Fekete, S.P., Pfisterer, D., Fischer, S.: Deterministic boundary recognition and topology extraction for large sensor networks. In: Proceedings of the ACM/SIAM Symposium on Discrete Algorithms (SODA), pp. 1000–1009 (2006)
48. Krupke, D.M., Ernestus, M., Hemmer, M., Fekete, S.P.: Distributed cohesive control for robot swarms: maintaining good connectivity in the presence of exterior forces. In: Proceedings of the 28th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 413–420 (2015)
49. La, H.M., Sheng, W.: Adaptive flocking control for dynamic target tracking in mobile sensor networks. In: Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4843–4848 (2009)
50. La, H.M., Sheng, W.: Flocking control of a mobile sensor network to track and observe a moving target. In: Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3129–3134 (2009)
51. Lee, D.-T., Lin, A.K.: Computational complexity of art gallery problems. *IEEE Trans. Inf. Theory* **32**(2), 276–282 (1986)
52. Lee, S.K., Becker, A.T., Fekete, S.P., Kröllner, A., McLurkin, J.: Exploration via structured triangulation by a multi-robot system with bearing-only low-resolution sensors. In: Proceedings of the 31st IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 2150–2157 (2014)
53. Lee, S.K., Fekete, S.P., McLurkin, J.: Virtual-agent coverage control in triangulated environments using geodesic Voronoi tessellation. In: Proceedings of the 27th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3858–3865 (2014)
54. Lee, S.K., Fekete, S.P., McLurkin, J.: Structured triangulation in multi-robot systems: coverage, patrolling, Voronoi partitions, and geodesic centers. *Int. J. Robot. Res.* **9**(35), 1234–1260 (2016)
55. Lee, S.K., McLurkin, J.: Distributed cohesive configuration control for swarm robots with boundary information and network sensing. In: Proceedings of the 27th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1161–1167. IEEE (2014)
56. Lindhé, M., Ögren, P., Johansson, K.H.: Flocking with obstacle avoidance: a new distributed coordination algorithm a new distributed coordination algorithm based on Voronoi partitions. In: Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 1785–1790 (2005)
57. Maftulec, D., Lee, S.K., Fekete, S.P., Akash, A.K., López-Ortiz, A., McLurkin, J.: Local policies for efficiently patrolling a triangulated region by a robot swarm. In: Proceedings of the 32nd IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 1809–1815 (2015)

58. Mahadev, A.V., Krupke, D., Fekete, S.P., Becker, A.T.: Mapping, foraging, and coverage with a particle swarm controlled by uniform inputs. In: Proceedings of the 30th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1097–1104 (2017)
59. Mahadev, A.V., Krupke, D., Reinhardt, J.-M., Fekete, S.P., Becker, A.T.: Collecting a swarm in a 2D environment using shared, global inputs. In: Proceedings of the 13th Conference on Automation Science and Engineering (CASE 2016), pp. 1231–1236 (2016)
60. McLurkin, J., Demaine, E.D.: A distributed boundary detection algorithm for multi-robot systems. In: Proceedings of the 22th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4791–4798 (2009)
61. McLurkin, J., et al.: A low-cost multi-robot system for research, teaching, and outreach. In: Martinoli, A., et al. (eds.) Proceedings 10th International Symposium on Distributed Autonomous Robotics Systems (DARS), vol. 83, pp. 597–609. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-32723-0_43
62. McLurkin, J., et al.: A robot system design for low-cost multi-robot manipulation. In: Proceedings of the 27th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 912–918 (2014)
63. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Autom. Control* **51**, 401–420 (2006)
64. O'Rourke, J.: *Art Gallery Theorems and Algorithms*. International Series of Monographs on Computer Science. Oxford University Press, New York (1987)
65. Prömel, H.J., Steger, A.: *The Steiner Tree Problem: A Tour Through Graphs, Algorithms, and Complexity*. Vieweg, Braunschweig (2002)
66. Reynolds, C.W.: Flocking, herds, and schools: a distributed behavioral model. *Comput. Graph.* **21**(4), 25–34 (1987)
67. Schmidt, A., Manzoor, S., Huang, L., Becker, A.T., Fekete, S.P.: Efficient parallel self-assembly under uniform control inputs. *IEEE Robot. Autom. Lett.* **3**, 3521–3528 (2018)
68. Schmidt, C.: *Algorithms for mobile agents with limited capabilities*. Ph.D. thesis, TU Braunschweig (2011)
69. Shad, H.M., Moris-Wright, R., Demaine, E.D., Fekete, S.P., Becker, A.T.: Particle computation: device fan-out and binary memory. In: Proceedings of the 32nd IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 5384–5389 (2015)
70. Shi, H., Wang, L., Chu, T., Xu, M.: Flocking coordination of multiple mobile autonomous agents with asymmetric interactions and switching topology. In: Proceedings of the 18th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 935–940 (2005)
71. Spears, W.M., Spears, D.F., Hammann, J.C., Heil, R.: Distributed, physics-based control of swarms of vehicles. *Auton. Robots* **17**(2–3), 137–162 (2004)
72. ThinkFun: Tilt: gravity fed logic maze. <http://www.thinkfun.com/tilt>
73. Tozoni, D.C., de Rezende, P.J., de Souza, C.C.: The quest for optimal solutions for the art gallery problem: a practical iterative algorithm. In: Bonifaci, V., Demetrescu, C., Marchetti-Spaccamela, A. (eds.) SEA 2013. LNCS, vol. 7933, pp. 320–336. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38527-8_29
74. Vartholomeos, P., Akhavan-Sharif, M., Dupont, P.E.: Motion planning for multiple millimeter-scale magnetic capsules in a fluid environment. In: Proceedings of the 29th IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA), pp. 1927–1932 (2012)