Controlling Distributed Particle Swarms with only Global Signals

Aaron T. Becker[1], Erik D. Demaine[2], and Sándor P. Fekete[3]

1 **Department of Electrical and Computer Engineering, University of Houston Houston, TX 77004, USA**
atbecker@uh.edu

2 **CSAIL, MIT Cambridge, MA 02139, USA**
edemaine@mit.edu

3 **Department of Computer Science, TU Braunschweig 38106 Braunschweig, Germany**
s.fekete@tu-bs.de

Abstract

We present fundamental progress on the computational universality of swarms of micro- or nano-scale particles in complex environments such as the vascular system of a biological organism. Components of the swarm are controlled not by individual navigation, but by a uniform global, external force. More specifically, we consider a 2D grid world, in which all obstacles and particles are unit squares, and for each actuation, particles move maximally until they collide with an obstacle or another particle. The objective is to control particle motion within obstacles, design obstacles in order to achieve desired permutation of particles, and establish controlled interaction that is complex enough to allow arbitrary computations. In this short paper, we summarize progress on all these challenges: we demonstrate NP-hardness of parallel navigation, we describe how to construct obstacles that allow arbitrary permutations, and we establish the necessary logic gates for performing arbitrary in-system computations.

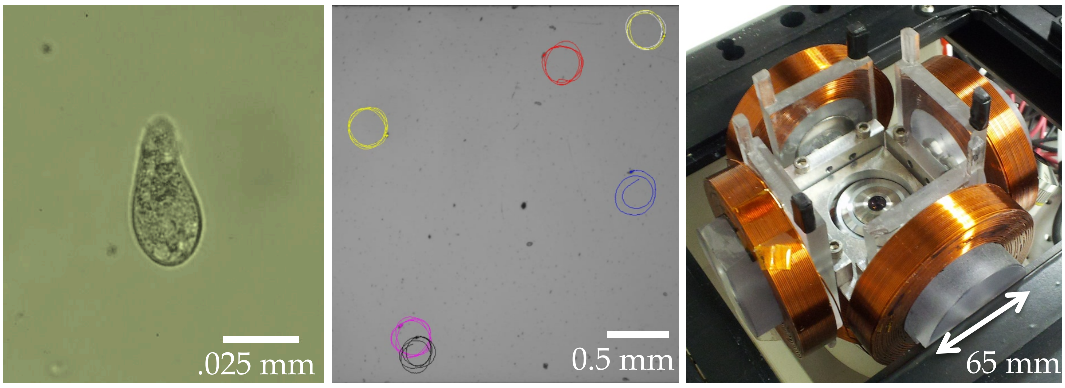
1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems–Geometrical problems and computations; F.1.1 Models of Computation–Bounded-action devices.

Keywords and phrases Particle swarms; global control; complexity; geometric computation.

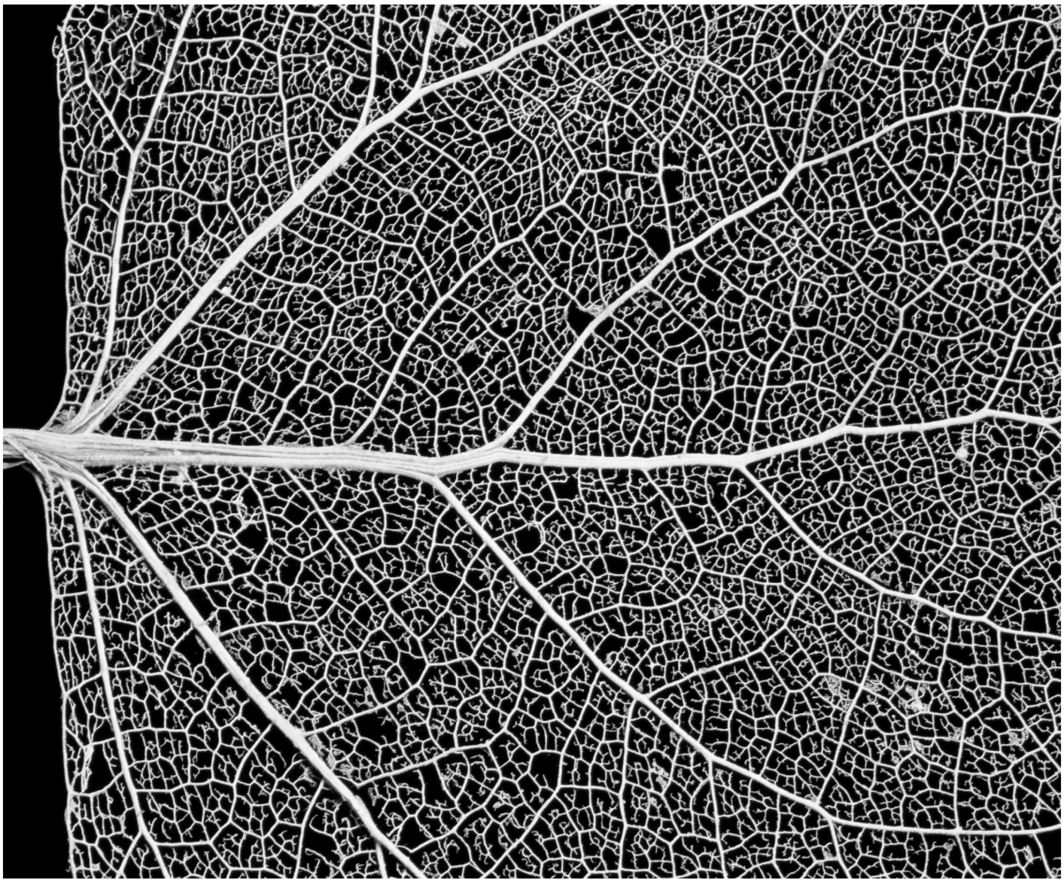
1 Introduction: Global Motion Control

One of the exciting new directions of robotics is the design and development of micro- and nanoparticle systems, with the goal of letting a massive swarm of particles perform complex operations in a complicated environment. Due to scaling issues, individual control of the involved particles becomes physically impossible: while energy storage capacity drops with the third power of particle 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 nanoparticle systems with many particles do: the whole swarm is steered and directed by an external force that acts as a common control signal; see Figure 1 for a visual impression in the context of biological systems, and our paper [3] for detailed references. These common control signals include global magnetic or electric fields, chemical gradients, and turning a light source on and off.

Clearly, having only one global signal that uniformly affects all particles 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 particle swarm and obstacles



(a) (Left, center) 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. (Right) Using two orthogonal Helmholz electromagnets (left), Becker et al. demonstrated steering many living magnetized *T. pyriformis* cells [5]. 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 is a graph structure with loops. MATLAB code for driving n robots through this network available at http://www.mathworks.com/matlabcentral/fileexchange/42892.

Figure 1 (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.

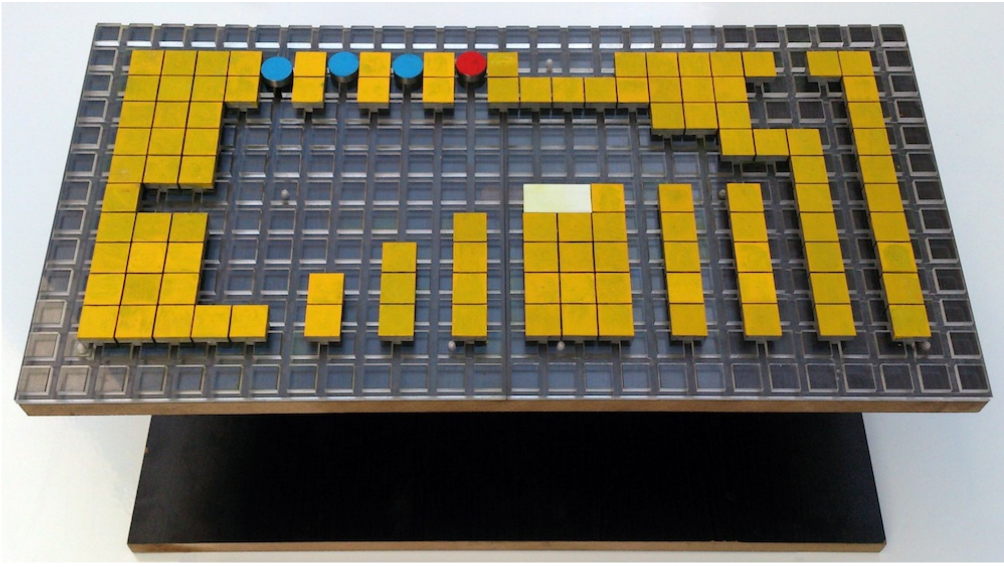


Figure 2 Gravity-fed hardware implementation of particle computation. The reconfigurable prototype is set up as a FAN-OUT gate using a 2×1 particle (white)

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.

This resembles the logic puzzle Tilt [11], 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 are also similar to sliding-block puzzles with fixed obstacles [6, 8, 9, 10], except that all particles receive the same control inputs, as in the Tilt puzzle. Another connection is Randolph's Ricochet Robots [7], a game that allows individual and independent control of the involved particles.

2 The Problems

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 $\langle u, r, d, l, d, r, u, \ldots\rangle$. We call these global commands *force-field moves*. We assume we can bound the minimum particle speed and can guarantee all particles have moved to their maximum extent.

Three of the most basic problems are as follows.

1. *Given a map of an environment, 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 an initial matrix arrangement of particles, how can we design a set of obstacles, such that any permutation can be realized with a relatively simple sequence of moves?*
3. *Can we establish sets of obstacles, particles, and moves, such that the resulting motion can be used for carrying out arbitrary computation strictly within the system, i.e., without an intelligent observer?*

3 The Results

We have provided answers for the above problems in our previous papers [1, 3, 2]. A it turns out, motion planning for fixed obstacles is difficult.

▶ **Theorem 1.** *(1) Given an initial configuration of movable particles and fixed obstacles, it is NP-hard to decide whether any particle can be moved to a specified location.*

(2) Given an initial configuration of labeled movable particles and fixed obstacles, it is PSPACE-complete to determine the minimum number of force-field moves to a given final configuration.

On the other hand, we were able to show that designing obstacles appropriately can be used to achieve arbitrary permutations.

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

Furthermore, using an extra particle of size 2×1 suffices to achieve universal computation.

▶ **Theorem 3.** *Using particles of size 1×1 and 2×1, motion planning with uniform external force can encode universal computation. This is not possible by only using 1×1 particles.*

In [4] we present a compact visual demonstration, in part based on a real-world realization, showing that further applications and extensions are possible.

References

1. A. T. Becker, E. D. Demaine, S. P. Fekete, and J. McLurkin. Particle computation: Designing worlds to control robot swarms with only global signals. In *ICRA*, pages 6751–6756, 2014.
2. A. T. Becker, E. D. Demaine, S. P. Fekete, S. H. Mohtasham Shad, and R. Morris-Wright. Reconfiguring massive particle swarms with limited, global control. In *ALGOSENSORS*, volume 8343 of *Springer LNCS*, pages 51–66, 2013.
3. A. T. Becker, E. D. Demaine, S. P. Fekete, S. H. Mohtasham Shad, and R. Morris-Wright. Particle computation: Device fan-out and binary memory. In *ICRA*, to appear, 2015.
4. A. T. Becker, E. D. Demaine, S. P. Fekete, S. H. Mohtasham Shad, and R. Morris-Wright. Tilt: The video. designing worlds to control robot swarms with only global signals. In *SoCG*, to appear, 2015. Video at https://youtu.be/H6o9Dlfkn0.
5. A. T. Becker, Y. Ou, P. Kim, M. J. Kim, and A. A. Julius. Feedback control of many magnetized tetrahymena pyriformis cells by exploiting phase inhomogeneity. In *IROS*, pages 3317–3323, November 2013.
6. E. D. Demaine, M. L. Demaine, and J. O'Rourke. PushPush and Push-1 are NP-hard in 2D. In *CCCG*, pages 211–219, August 2000.
7. B. Engels and T. Kamphans. Randolphs robot game is NP-hard! *Electronic Notes in Discrete Mathematics*, 25:49–53, 2006.
8. R. A. Hearn and E. D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *arXiv:cs/0205005*, cs.CC/0205005, 2002.
9. M. Hoffmann. Motion planning amidst movable square blocks: Push-* is NP-hard. In *CCCG*, pages 205–210, June 2000.
10. M. Holzer and S. Schwoon. Assembling molecules in ATOMIX is hard. *Theoretical Computer Science*, 313(3):447–462, 2004.
11. ThinkFun. Tilt: Gravity fed logic maze. `http://www.thinkfun.com/tilt`.