Collecting a Swarm in a Grid Environment Using Shared, Global Inputs

Arun V. Mahadev, Dominik Krupke, Jan-Marc Reinhardt, Sándor P. Fekete, Aaron T. Becker

Abstract— **This paper investigates efficient techniques to collect and concentrate an under-actuated particle swarm despite obstacles. Concentrating a swarm of particles is of critical importance in health-care for targeted drug delivery, where micro-scale particles must be steered to a goal location. Individual particles must be small in order to navigate through micro-vasculature, but decreasing size brings new challenges. Individual particles are too small to contain on-board power or computation and are instead controlled by a global input, such as an applied fluidic flow or electric field.**

To make progress, this paper considers a swarm of robots initialized in a grid world in which each position is either free-space or obstacle. This paper provides algorithms that collect all the robots to one position and compares these algorithms on the basis of efficiency and implementation time.

I. Introduction

For many therapeutic treatments it is important to concentrate a drug at a particular site. The old adage *toxicity is a function of concentration* explains that often we can flow a diluted drug through the body without ill-effect, and then kill cells at a targeted location by collecting drug particles. Targeted drug therapy is a goal for many interventions, including treating cancers, delivering pain-killers, and stopping internal bleeding. Treatment often uses the patient's vasculature to deliver the therapy. This drug therapy is challenging due to the complicated geometry of vasculature, as shown in Fig. 1.

This paper builds on the techniques for controlling many simple robots with uniform control inputs presented in [1], [2], and outlines new research problems; see video and abstract [3] for a visualizing overview.

a) Definitions: The 'robots' in this paper are simple particles without autonomy. A planar grid *workspace* W is filled with a number of unit-square robots (each occupying one cell of the grid) and some fixed unit-square blocks. Each unit square in the workspace is either *free*, which a robot may occupy or *obstacle* which a robot may not occupy. Each square in the grid can be referenced by its Cartesian coordinates $\boldsymbol{x} = (x, y)$. All robots are commanded in unison: the valid commands are "Go Up" (u), "Go Right" (r), "Go Down" (d), or "Go Left" (l).

We consider two classes of commands, discrete and maximal moves. *Discrete moves*: robots all move in the commanded direction one unit unless they are prevented from moving by an obstacle or a stationary robot.

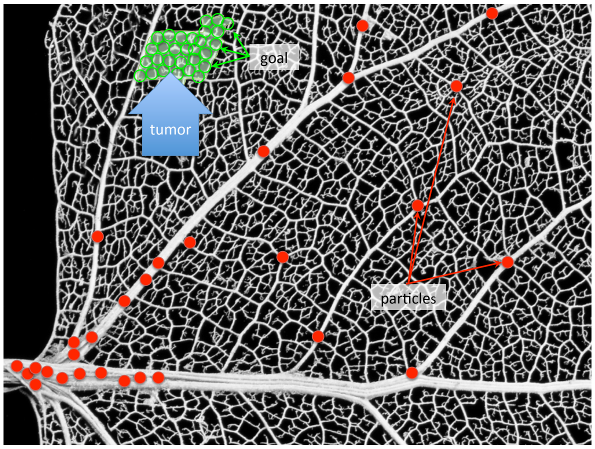

Fig. 1. Vascular networks are common in biology such as the circulatory system and cerebrospinal spaces, as well as in porous media including sponges and pumice stone. Navigating a swarm using global inputs, where each member receives the same control inputs, is challenging due to the many obstacles. This paper focuses on using boundary walls to break the symmetry and collect the swarm at a desired location. See simulation at [4].

Maximal moves: robots all move in the commanded direction until they hit an obstacle or a stationary robot. For maximal moves, we assume the area of W is finite and issue each command long enough for the robots to reach their maximum extent. A command sequence $\boldsymbol{m}$ consists of an ordered sequence of moves m_k, where each $m_k \in \{u, d, r, l\}$ A representative command sequence is $\langle u, r, d, l, d, r, u, \ldots \rangle$.

We consider two types of particles, small and large, as depicted in Fig. 2. If particles are much smaller than the workspace geometry, we call them small. We represent each grid cell as filled if it contains at least one particle and empty otherwise. A cell filled with small particles can combine with another filled cell. If particles are the same size as workspace gridcells, we call the particles large. Large particles cannot combine. The presence of a large particle in a cell prevents another particle from entering.

We study two notions of collecting a swarm, corresponding with particle size: for small particles the swarm is collected when all robots share the same (x, y) coordinates. If the particles are large, the swarm is collected when it forms one connected component. 2D cells are neighbors if they share an edge, 3D cells are neighbors if they share a face. A *connected component* is a set of particles P such that for any two particles in P, there is a sequence of neighboring particles that connect them.

A. Mahadev and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA aviswanathanmahadev@uh.edu, atbecker@uh.edu
S. Fekete, D. Krupke, and J.M. Reinhardt are with the Dept. of Computer Science, TU Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany, s.fekete@tu-bs.de, j-m.reinhardt@tu-bs.de, d.krupke@tu-bs.de

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 28, 2016.

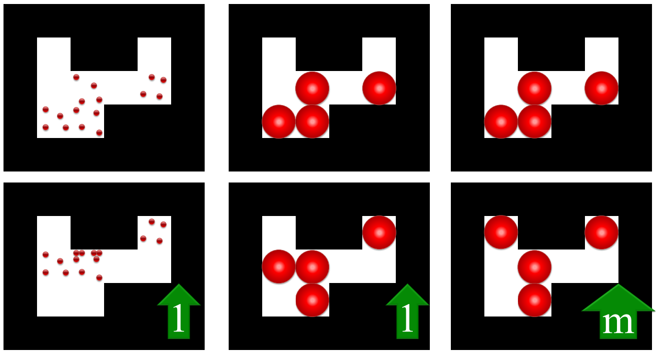

Fig. 2. If particles are much smaller than the workspace geometry, we call them *small*. We represent each grid cell as filled or empty, and allow a filled cell to combine with another filled cell. If particles are the same size as workspace gridcells, we call the particles *large*. Large particles cannot combine. The presence of a large particle in a cell prevents another particle from entering.

II. Related work

Collecting particles in an obstacle filled workspace has strong parallels with robot localization, rendezvous, and has applications to drug delivery and manufacturing.

A. Micro-Robots with Global Control

For a recent survey on challenges related to controlling multiple micro robots (less than 64 robots at a time), see [5]. This paper seeks to understand control for large numbers of microrobots, and uses a generalized model that could apply to a variety of drug-carrying micro-particles. One example are particles with a magnetic core and a catalytic surface for carrying medicinal payloads [6], [7]. An alternative are aggregates of *superparamagnetic iron oxide microparticles*, 9 μm particles that are used as a contrast agent in MRI studies [8]. Real-time MRI scanning could allow feedback control using the location of a swarm of these particles.

Steering magnetic particles using the magnetic gradient coils in an MRI scanner was implemented in [6], [9]. 3D Maxwell-Helmholtz coils are often used for precise magnetic field control [8].

B. Localization

The strongest parallel in the literature is the field of *Almost sensorless localization* or "localizing a blind robot in a known map", where a mobile robot with a map of the workspace must localize itself [10], [11], using only a compass and a bump-sensor that detects when the robot contacts a wall. This has been extended to robots with bounded uncertainty in their inputs [12]. The basic methodology is to design a locating sequence of movements. First: actions are selected that reduce position uncertainty to a finite set of possibilities. This is accomplished by driving the robot until it collides with a workspace boundary or obstacle. Second: actions are selected to reduce the cardinality of this finite set until it reduces to a single point [11].

Given an environment, finding a localizing sequence is framed as a planning problem with an unknown initial state and an unobservable current state. The solution in [10] was to transform the problem from an unobservable planning problem in state space to an observable problem in a more complex information space. They provided a complete algorithm, but generating an optimal localizing sequences remains an open problem. [10] assumes there is only one robot, while our paper gathers a large number of robots.

Also related is work on sensorless part orientation, where a flat tray is tilted in a series of directions to bring a polygonal part, initially placed at random orientation and position in the tray, to a known position and orientation [13]. This is similar to localizing a robot with minimum travel; however, it moves the robot and has it take additional measurements [14].

C. Robot Rendezvous

The concept of robot rendezvous usually concerns two or more independent, intelligent agents that must meet at a predetermined time: Starting at unknown locations, without any communication with each other, the goal is to meet in minimum time such that the particles interact with each other in order to communicate information that can only be shared when in close proximity. See the survey book by Alpern and Gal [15] for a wide range of models and methods; in a two-dimensional geometric setting, Anderson and Fekete [16] have studied various scenarios.

In the context of robotics, key assumptions include a bounded topological environment and robots. This is of interest to us because we have a key focus on maneuvering particles through worlds with obstacles and implementation of strategies to reduce computational burden while calculating distances in such complex worlds [17]. In a setting with autonomous robots, these can move independent of each other, i.e., follow different movement protocols, which is called *asymmetric* rendezvous in the mathematical literature [15]. If the agents are required to follow the same protocol, this is called *symmetric* rendezvous. This corresponds to our model in which particles are bound by the uniform motion constraint; symmetry is broken by interaction with the obstacles.

The main variation that occurs in this field of research lies on how a successful rendezvous is defined. It can be defined as "The time taken for successful transitive rendezvous between all agents" [17]. Or it can be considered as "aggregation of all agents to a given spot' [18].

D. Manufacturing

Consider a manufactured component with arbitrary geometry that is filled with fluid. Tipping the part in a given orientation allows the fluid to move and drain from part openings [19]. Determining a sequence of orientations that drain all the fluid is similar to gathering particles, but here the goal is to remove as much fluid as possible. When designing parts for injection molding, the challenge is the opposite – where should input ports be placed so that injected plastic will fill the entire void [20]. This can be framed as an inverse problem of gathering all the empty squares to a single point and then filling that point.

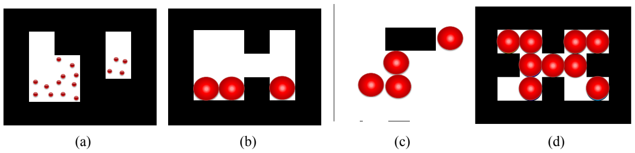

Fig. 3. Examples of workspaces for which collection is not possible. (a) The swarms are in unconnected components (b) A world for which *maximal* moves will never allow particles to meet. (c) An unbounded world with a single obstacle. In this world *discrete* moves can collect particles but *maximal* moves cannot collect particles. In a world without boundaries and without obstacles, discrete moves cannot gather all particles. (d) A world configuration where *large* particle collection is impossible. No input sequence exists that will make all the particles part of the same connected component.

III. Our problems of interest

Robots initialized in two unconnected components i and j of a free space cannot be collected. The proof is trivial, since a robot in free space i can not reach free space j. Such a configuration is depicted in Fig. 3a.

Under maximal inputs, the world can be constructed with spaces resembling bottles or fish weirs from which a single robot cannot escape, as shown in Fig. 3b. If the free space contains at least two such bottles with at least one robot in each, the swarm cannot be collected with maximal inputs.

The world must be bounded. Two initially separated robots in an unbounded world without obstacles cannot be collected; however with discrete inputs, one obstacle is sufficient as seen in Fig. 3c and can be inferred from [21].

A swarm with discrete moves and small particles can be collected on any bounded grid. However, with large particles there are configurations where the topology does not allow collection, as seen in Fig. 3d. Section IV explores collection algorithms and analyzes how the move sequence and size of particles affects swarm collection.

IV. Theory

With discrete inputs and small particles, the problem can be reduced to localizing a sensorless robot in a known workspace. This is similar to work on draining a polygon [19], or localizing a blind robot [10], [11], but with discrete inputs. For the small particle problem we present an optimal collection algorithm, Alg. 1 in Section IV-A and a greedy collection policy Alg. 2 in Section IV-B. We also give positive (Section IV-C) and negative (Section IV-D) results for large particles.

A. Collecting with the shortest move sequence

A conceptually simple strategy to collect all particles in a workspace is to construct a configuration tree that expands the tree of all possible movement sequences in a breadth-first search manner, and halts when the configuration has all robots collected at one point. Alg. 1 implements this breadth-first-search technique. It initializes a tree where each node contains the configuration of robot locations $C[p]$, the move that generated this configuration $M[p]$ and a parent configuration pointer $P[p]$. Here C, M, P are the respective complete lists. p is the current iteration pointer and e is the end of list pointer. The root node is $\{C_0, \varnothing, 0\}$, where C_0 is the initial configuration of robot locations. We then construct a breadth-first tree of possible configurations $\{u, r, d, l\}$, pruning configurations that already exist in the tree. We stop when the cardinality at a leaf is one, $|C_i|= 1$, which indicates that the swarm has been collected (equivalently, that the robot has been localized). This algorithm produces the optimal path as seen in Fig. 4, but requires $O(4^{\text{shortest path length}})$ time to learn and $O(4^{\text{shortest path length}})$ memory and the graph grows exponentially (Fig. 5). This leads us to investigate other algorithms which will solve the path with much lower computational time and data.

Algorithm 1 OptimalGathering(W, C_0)

```
p ← 1
{C[p], M[p], P[p]} ← {C_0, ∅, 0}                ▷ initialize
e ← 1
while |C[p]| > 1 do               ▷ more than 1 unique position
    for m = {u, d, r, l} do
        C_temp ← ApplyMove( C[p], m)
        if C_temp ∉ C then                   ▷ add node to list
            e ← e + 1
            {C[e], M[e], P[e]} ← {C_temp, m, p}
        end if
    end for
    p ← p + 1                        ▷ get next configuration
end while
path ← {}                            ▷ construct optimal path
while P[p] > 1 do
    Append M[p] to path
    p ← P[p]
end while
path ← Reverse[path]
```

B. Collecting small particles with a greedy strategy

Two particles in a finite and connected polyomino can be collected with small particles and discrete movement by simply repeatedly moving one particle onto the other on the shortest way. The corresponding procedure CollectAB is described in Alg. 2. By iteratively collecting any two disjoint particles, the size of the distinct positions of the particle swarm can be reduced until all particles are on the same position. The two particles can be chosen with different methods and our focus will be to implement the following methods:

1) Closest pair of particles - choose a pair of particles with the closest distance between them.
2) Furthest pair of particles - choose a pair with maximum distance between them.
3) Connecting to first - choose the first two particles while searching for particles in workspace from top left to bottom right.
4) Random combinations - choose any two particles.

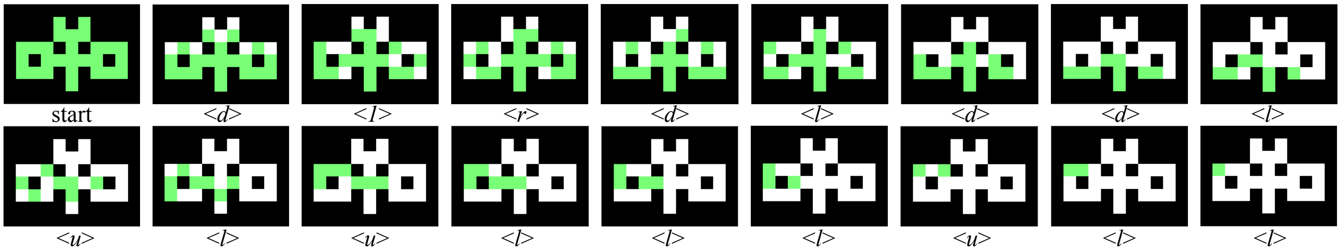

Fig. 4. With discrete inputs and particles, the gathering problem can be reduced to localizing a sensor-less robot in a known workspace. Above shows the optimal solution for a world with 27 free spaces, which required expanding 423,440 nodes with an optimal path (shown) taking 17 moves.

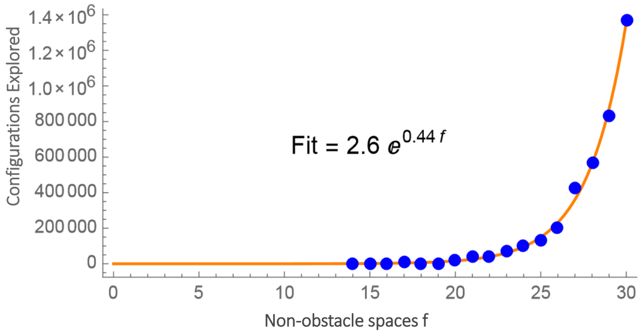

Fig. 5. Unfortunately, the optimal BFS (Algorithm1) solution requires expanding a number of nodes that increases (approximately) exponentially with the number of free spaces. A workspace with 30 free spaces required 1.6 million nodes before finding the optimal solution.

5) First to last - choose the first particle and last particle, i.e., the leftmost top and rightmost bottom particles, respectively.

Here CollectAB can be called to implement any one of the algorithms.

Algorithm 2 Collecting two particles that can overlap

```
Require: $a$ can reach $b$, Polyomino is bounded
procedure CollectAB(a: Particle, b: Particle)
    while $dist(a,b) \neq 0$ do
        Let $\mathcal{C} \in \{u,d,l,r\}^N$ be the shortest control sequence that moves $a$ onto $pos(b)$
        Execute $\mathcal{C}$
    end while
end procedure
```

Theorem 1: CollectAB collects two particles in a polyomino with $O(n^3)$ discrete control commands, where n equals the polyomino's height times its width.

Proof: The distance between a and b equals the length of $\mathcal{C}$. After execution of $\mathcal{C}$, the distance has not increased as a is now on the previous position of b and b has at most moved $|\mathcal{C}|$ fields from it. If b had a collision during the execution of $\mathcal{C}$, the distance is even less as at least one command did not result in a move of b. As $dist(a,b) \in O(n)$, only $O(n)$ loop iterations with collisions are needed to collect a and b. Obviously, $|\mathcal{C}| \in O(n)$ and hence every loop iteration executes at most $O(n)$ commands. With every iteration without collision, the positions of a and b change each by $pos(b) - pos(a)$. This difference does only change if b had a collision, therefore the particles move in the same direction with every collision-free iteration. After $O(n)$ collision-free iterations of the loop, b has to have a collision, as the polyomino is finite. This results in $O(n^2)$ commands to reduce the distance by at least one and thus $O(n^3)$ commands suffice to collect a and b. ■

Theorem 2: CollectAB has a computational complexity of $O(n^3)$.

Proof: The shortest control sequence $\mathcal{C}$ can be calculated in $O(n)$ time by a simple breadth-first-search. Under the assumption that a command can be executed in $O(1)$, one loop iteration has a computational complexity of $O(n)$. With $O(n^2)$ loop iterations (see proof of Theorem 1), this results in an overall complexity of $O(n^3)$. ■

Theorem 3: A particle swarm of size $O(m)$ can be collected with $O(m * n^3)$ discrete control commands and a computational complexity of $O(m * n^3)$ where n equals the polyomino's height times its width.

Proof: Select two disjunct particles and execute `CollectAB`. This reduces the size of distinct positions in the particle swarm by one. After $O(m)$ executions, there is only one position left and the particle swarm is collected. ■

C. Collecting large particles in a target region

In the previous two subsections, the assumption was that the particles are relatively small, allowing several of them to be collected in the same location anywhere in the environment. If the particles are relatively large, they may block each other's way, making the motion control trickier. We can still deliver a swarm of particles to a target region by making use of discrete moves, assuming that particles are metabolized once they reach the target region, i.e., the target is "sticky". (This implies that they stay within the target region once they get there, and that they do not block each other within that region.)

Theorem 4: For a sticky target region and large particles within an environment of diameter D, a particle swarm of size $O(m)$ can be collected with $O(m \times D)$ discrete control commands.

Proof: The proof is straightforward by induction.

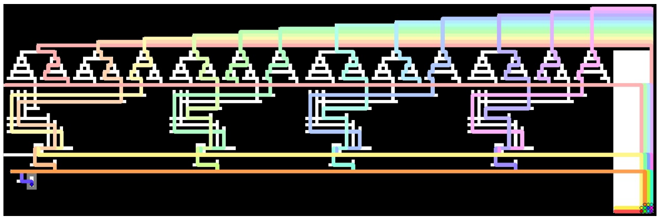

Fig. 6. NP-hardness of deciding reachability with maximal moves: In order to deliver a *single* "blue" particle into the grey target region in the lower left, a 3SAT instance has to be satisfied, corresponding to a set of decisions in the upper part of the construction.

Moving one particle to the target region takes at most D moves, which leaves all other particles within distance D. ■

Note that the extent of the environment is critical. If we are dealing with m particles within an environment of size $n \times n$, then we get the following.

Corollary 5: For a sticky target region and large particles, a particle swarm within an environment of size $n \times n$ can be collected with $O(n^3)$ discrete control commands.

If all particles of the swarm are relatively close to the target region, the complexity can be stated differently.

Corollary 6: For a sticky target region and large particles, a particle swarm of size $O(m)$ that fills a square environment around a target region can be collected with $O(m^{3/2})$ discrete control commands.

D. Collecting large particles with maximal moves

Our results rely on being able to limit the extent of the motion, i.e., having *discrete moves*. If that is not the case, i.e., in the case of *maximal moves*, in which each particle moves until it is stopped by an obstacle or another stopped particle, the problem becomes considerably harder and may indeed be intractable. (See Fig. 6 for the basic idea.) As we showed in previous work [22], deciding whether even a *single* particle can be delivered to a target region is already an NP-hard problem; this implies not only that finding a solution is computationally hard, but that there are instances in which no solution exists.

V. Results and Inferences

Experiment one compares the optimal algorithm (Alg. 1) versus three varients of the greedy algorithm (Alg. 2). Fig. 7 compares the number of moves required to converge for the optimal strategy (Alg. 1) versus the greedy strategy for small worlds ranging from 5 free spaces to 30 free spaces. Workspaces with more spaces were not considered because the optimal algorithm's requires exponential time in the number of free spaces. This plot shows that the optimal algorithm requires approximately half the moves of the greedy algorithms. The plot has an upward moving trend in general as the number of free spaces increases, but there are local minimums corresponding to easier configurations, which leads to downward spikes in the plot. The number of moves taken to completely collect particles also depends on the complexity of the workspace and does not completely depend on the number of free spaces.

For small workspaces the best result among the greedy algorithms changes and so we cannot determine which is the best using small workspaces. To further compare the greedy strategies, we tested the algorithms on larger workspaces. The largest workspace in Fig. 8 has 8,493 non-obstacle positions. Fig. 8 demonstrates that choosing which particles to pairwise collect in Alg. 2 has a large impact on convergence time. We conducted a comparison study between the number of moves and the resulting unique particles. As discussed earlier in Alg. 1 (Section IV-A) and Alg. 2 (Section IV-B), getting the number of unique particles down to '1' signifies completion of the collecting algorithm. In the leaf vascular network, the majority of particle collection occurs during the first steps, with a long tail distribution to collect the final particles, as shown in the top row of images. Fig. 8 shows that *connecting to first*, discussed in Section IV.B, outperforms the other algorithms. This can also be validated by further testing to compare the three greedy algorithms on larger workspaces.

We simulated bounded worlds of varied sizes from 500 free spaces to 8,493 free spaces. The results are represented in Fig. 9, based on Fig. 1 because biological vasculatures are our goal application. This graph plot has a smoother trend compared to the plot in Fig. 7 because the ratio of free space to node complexity is similar for worlds in Fig. 9. The important observation is the consistency that *connecting to first* performs best. This validates *connecting to first* as the best of the compared algorithms. This is good news because unlike the other two techniques, which involve distance calculation between all pairs of particles, there is negligible calculation involved in the *connecting to first* algorithm. The data for which two particle are in top-leftmost location is readily available from the row, column indices of the particles.

VI. Conclusion and Future work

This paper presented optimal and greedy algorithms to collect small globally commanded particles with guarantees that these algorithms will always collect particles for any bounded world which can be represented as a connected polyomino. Algorithm *connecting to first* combines both low computational time and, in simulations, requires fewer moves than five other algorithms. It requires 50,607 moves to converge all the particles in the complete leaf world shown in Fig. 1 (see video [4]). We also introduced challenges inherent with large particle collection, which poses new problems and complexities.

The technology to fabricate microbots is rapidly improving and so has interest in microrobots for potential applications in drug delivery.

There are many opportunities for future work, including refining the algorithms to handle large particles. This paper assumed the workspace was bounded. That assumption is violated in biological vascular systems, which connect to larger vasculature. One avenue for future research is to add

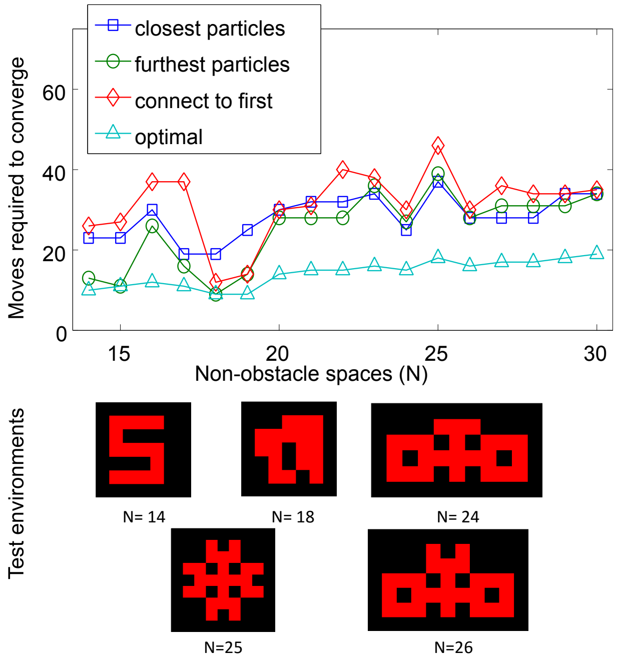

Fig. 7. The greedy strategy requires 1.95 as many moves as the optimal strategy in a test with 17 different test environments. Below the plot are examples of some of the test environments used.

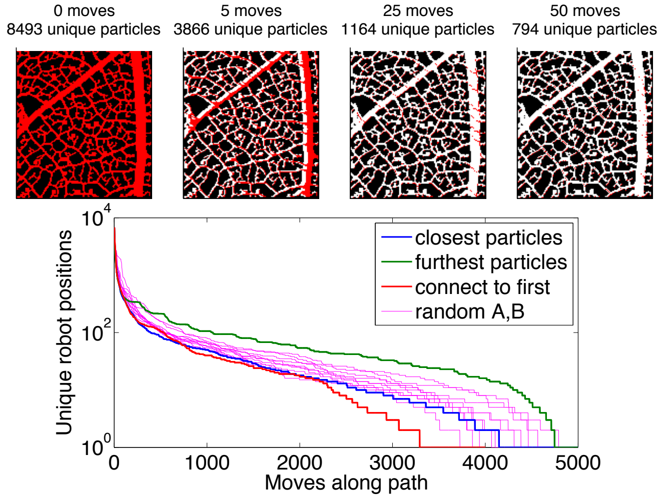

Fig. 8. Choosing which two particles to collect during each COLLECTAB step changes convergence time. Gathering by collecting the furthest particles (green) performs poorly. Connecting the two closest nodes (blue) is better, but both strategies are beat by the strategy that chooses the first two (the top-most, left-most) particles each iteration.

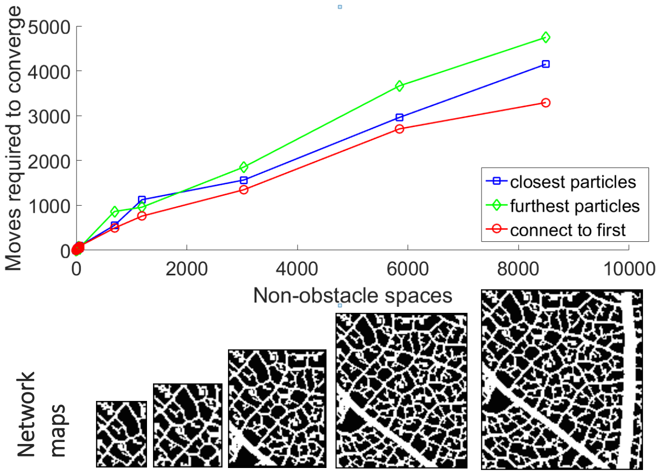

Fig. 9. Comparing the required number of moves to collect overlapping particles to a position for three greedy algorithms (Alg. 2) for five different, connected vascular networks.

constraints to serve as virtual walls and actively prevent particles from escaping through a set of exits. Additional complexities such as medium viscosity and wall friction must be studied before the algorithms are applied in vivo/in vitro. Future work should focus not only on gathering, but also on avoiding accumulation in sensitive regions.

REFERENCES

[1] A. Becker, E. Demaine, S. Fekete, and J. McLurkin, "Particle computation: Designing worlds to control robot swarms with only global signals," in *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, May 2014, pp. 6751–6756.

[2] A. Becker, E. D. Demaine, S. P. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *Algorithms for Sensor Systems*, ser. Lecture Notes in Computer Science, P. Flocchini, J. Gao, E. Kranakis, and F. Meyer auf der Heide, Eds. Springer Berlin Heidelberg, 2014, vol. 8243, pp. 51–66. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-45346-5_5

[3] A. Becker, E. D. Demaine, S. P. Fekete, S. H. M. Shad, and R. Morris-Wright, "Tilt: The video. designing worlds to control robot swarms with only global signals," in *31st International Symposium on Computational Geometry (SoCG'15)*, 2015, pp. 16–18.

[4] A. V. Mahadev, D. Krupke, J.-M. Reinhardt, S. P. Fekete, and A. T. Becker, "Collecting a swarm in a grid environment using shared, global inputs," Youtube, Mar 2016. [Online]. Available: https://www.youtube.com/watch?v=rY7Br4l4SHY

[5] S. Chowdhury, W. Jing, and D. J. Cappelleri, "Controlling multiple microrobots: recent progress and future challenges," *Journal of Micro-Bio Robotics*, vol. 10, no. 1-4, pp. 1–11, 2015. [Online]. Available: http://dx.doi.org/10.1007/s12213-015-0083-6

[6] P. Pouponneau, J.-C. Leroux, and S. Martel, "Magnetic nanoparticles encapsulated into biodegradable microparticles steered with an upgraded magnetic resonance imaging system for tumor chemoembolization," *Biomaterials*, vol. 30, no. 31, pp. 6327–6332, 2009.

[7] J. Litvinov, A. Nasrullah, T. Sherlock, Y.-J. Wang, P. Ruchhoeft, and R. C. Willson, "High-throughput top-down fabrication of uniform magnetic particles," *PloS one*, vol. 7, no. 5, p. e37440, 2012.

[8] L. Mellal, D. Folio, K. Belharet, and A. Ferreira, "Magnetic microbot design framework for antiangiogenic tumor therapy," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1397–1402.

[9] J.-B. Mathieu and S. Martel, "Magnetic microparticle steering within the constraints of an mri system: proof of concept of a novel targeting approach," *Biomedical microdevices*, vol. 9, no. 6, pp. 801–808, 2007.

[10] J. O'Kane, "Almost-sensorless localization," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2005.

[11] J. O'Kane and S. M. LaValle, "Almost-sensorless localization," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3764–3769.

[12] J. S. Lewis and J. M. O'Kane, "Planning for provably reliable navigation using an unreliable, nearly sensorless robot," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1342–1357, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/11/1342.abstract

[13] S. Akella and M. T. Mason, "Orienting toleranced polygonal parts," *The International Journal of Robotics Research*, vol. 19, no. 12, pp. 1147–1170, Dec. 2000.

[14] G. Dudek, K. Romanik, and S. Whitesides, "Localizing a robot with minimum travel," *SIAM Journal on Computing*, vol. 27, no. 2, pp. 583–604, 1998.

[15] S. Alpern and S. Gal, *The theory of search games and rendezvous*, ser. International Series in Operations Research and Managment Science. Boston, Dordrecht, London: Kluwer Academic Publishers, 2003. [Online]. Available: http://opac.inria.fr/record=b1117985

[16] E. J. Anderson and S. P. Fekete, "Two dimensional rendezvous search," *Operations Research*, vol. 49, no. 1, pp. 107–118, 2001.

[17] M. Meghjani and G. Dudek, "Multi-robot exploration and rendezvous on graphs," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5270–5276.

[18] P. Zebrowski, Y. Litus, and R. T. Vaughan, "Energy efficient robot rendezvous," in *Computer and Robot Vision, 2007. CRV'07. Fourth Canadian Conference on*. IEEE, 2007, pp. 139–148.

[19] G. Aloupis, J. Cardinal, S. Collette, F. Hurtado, S. Langerman, and J. ORourke, "Draining a polygon—or—rolling a ball out of a polygon," *Computational geometry*, vol. 47, no. 2, pp. 316–328, 2014.

[20] R. A. Malloy, *Plastic part design for injection molding*. Hanser Gardner Publications, 1994.

[21] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Massive uniform manipulation: Controlling large populations of simple robots with a common input signal," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov. 2013, pp. 520–527.

[22] A. Becker, E. Demaine, S. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, Sophia Antipolis, France, Sep. 2013, pp. 51–66.

Preprint submitted to 12th Conference on Automation Science and Engineering.
Received March 28, 2016.