Targeted Drug Delivery: Algorithmic Methods for Collecting a Swarm of Particles with Uniform, External Forces

Aaron T. Becker*[1], Sándor P. Fekete*[2], Li Huang*[1], Phillip Keldenich*[2], Linda Kleist*[2], Dominik Krupke*[2], Christian Rieck*[2], Arne Schmidt*[2]

***Abstract*— We investigate algorithmic approaches for targeted drug delivery in a complex, maze-like environment, such as a vascular system. The basic scenario is given by a large swarm of micro-scale particles ("agents") and a particular target region ("tumor") within a system of passageways. Agents are too small to contain on-board power or computation and are instead controlled by a global external force that acts uniformly on all particles, such as an applied fluidic flow or electromagnetic field. The challenge is to deliver all agents to the target region with a minimum number of actuation steps. We provide a number of results for this challenge. We show that the underlying problem is NP-hard, which explains why previous work did not provide provably efficient algorithms. We also develop a number of algorithmic approaches that greatly improve the worst-case guarantees for the number of required actuation steps. We evaluate our algorithmic approaches by a number of simulations, both for deterministic algorithms and searches supported by deep learning, which show that the performance is practically promising.**

I. Introduction

A crucial challenge for a wide range of vital medical problems, such as the treatment of cancer, localized infections and inflammation, or internal bleeding is to deliver active substances to a specific location in an organism. The traditional approach of administering a sufficiently large supply of these substances into the circulating blood may cause serious side effects, as the outcome intended for the target site may also occur in other places, with often undesired, serious consequences. Moreover, novel custom-made substances that are specifically designed for precise effects are usually in too short supply to be generously poured into the blood stream. In the context of targeting brain tumors (see Fig. 1), an additional difficulty is the blood-brain barrier. This makes it necessary to develop other, more focused methods for delivering agents to specific target regions.

Given the main scenario of medical applications, this requires dealing with navigation through complex vascular systems, in which access to a target location is provided by pathways (in the form of blood vessels) through a maze of obstacles. However, the microscopic size of particles necessary for passage through these vessels makes it prohibitively difficult to store sufficient energy in suitably sized microrobots, in particular in the presence of flowing blood.

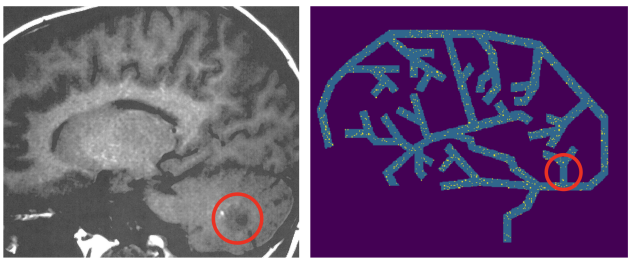

Fig. 1. (Left) An MRI image of a brain tumor (marked by the red circle), located in the cerebellum. (Right) How can the swarm of particles (indicated by yellow dots) be delivered to the target region?

A promising alternative is offered by employing a global external force, e.g., a fluidic flow or an electromagnetic field. When such a force is applied, all particles move in the same direction by the distance, unless they are blocked by obstacles in their way. While this makes it possible to move all particles at once, it introduces the difficulty of using *uniform* forces for many particles in *different* locations with different local topology to navigate them to *one* final destination. In this paper, we investigate how this objective can be achieved with a small number of actuator steps.

Previous work [14] described a basic approach that delivers all particles in a grid environment with n grid cells to a target in at most $O(n^3)$ actuator steps. This shows that delivery can always be achieved; however, a delivery time of this magnitude is usually impractical, which is why we investigate possible improvements.

Our Contribution. We provide a number of insights:

- We prove that minimizing the length of a command sequence for gathering all particles is NP-hard, even for environments that consist of grid cells in the plane, so no polynomial-time algorithms can be expected. This explains the observed difficulty of the problem and also implies hardness for the related localization problem.
- We develop an algorithmic strategy for gathering all particles with a worst-case guarantee of at most $O(kD^2)$ steps; here D denotes the maximum distance between any two points of the environment and k the number of its convex corners. Both k and D are usually much smaller than the number n of grid locations in the environment: n may be in $\Omega(D^2)$, for two-dimensional and in $\Omega(D^3)$ for three-dimensional environments.
- For the special case of hole-free environments, we can gather all particles in $O(kD)$ steps.

*Reflecting the fact that all authors have significantly contributed to this paper, the order of authors follows the Hardy-Littlewood rule, i.e., it is alphabetical.

[1]Department of Electrical and Computer Engineering, University of Houston, USA. {atbecker,lhuang28}@uh.edu

[2]Department of Computer Science, TU Braunschweig, Germany. {s.fekete, d.krupke, p.keldenich, l.kleist, c.rieck, arne.schmidt}@tu-bs.de

- We successfully apply deep learning to search for short command sequences in individual, complex instances.
- We perform a simulation study of the various approaches, evaluating the respective performance for application-inspired instances.

A. Related Work

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 microparticles. An example are particles with a magnetic core and a catalytic surface for carrying medicinal payloads [13], [18]. An alternative are aggregates of *superparamagnetic iron oxide microparticles*, 9 μm particles that are used as a contrast agent in MRI studies [17]. Real-time MRI scanning can 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 [15], [18]. 3D Maxwell-Helmholtz coils are often used for precise magnetic field control [17]. Still needed are motion planning algorithms to guide the swarms of robots through vascular networks. To this end, we build on the techniques for controlling many simple robots with uniform control inputs presented in [5]–[7]; see video and abstract [4] for a visualizing overview. For a recent survey on challenges related to controlling multiple microrobots (less than 64 robots at a time), see [10]. Further related work includes assembling shapes by global control (e.g., see [3], [8]) or rearranging particles in a rectangle of agents in a confined workspace [19], [20].

As the underlying problem consists of bringing together a number of agents in one location, a highly relevant algorithmic line of research considers *rendezvous search*, which requires two or more independent, intelligent agents to meet. Alpern and Gal [1] introduced a wide range of models and methods for this concept as have Anderson and Fekete [2] in a two-dimensional geometric setting. Key assumptions include a bounded topological environment and robots with limited onboard computation. This is relevant to maneuvering particles through worlds with obstacles and implementation of strategies to reduce computational burden while calculating distances in complex worlds [16]. In a setting with autonomous robots, these can move independent of each other, i.e., follow different movement protocols, called *asymmetric* rendezvous in the mathematical literature [1]. 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 only by interaction with the obstacles. For an overview of a variety of other algorithmic results on gathering a swarm of autonomous robots, see the recent survey by Flocchini [11]; note that these results assume a high degree of autonomy and computational power for each individual agent, so their applicability for our scenarios is quite limited.

II. Preliminaries

The "robots" in this paper are simple particles without autonomy. We assume that their size is insignificant compared to the elementary cells in the workspace P. Due to the limited space of this paper, our description focuses on planar workspaces P, consisting of orthogonal sets of cells, so-called *pixels*, that form an edge-to-edge connected domain in the integer planar grid, i.e., a *polyomino*. (As we sketch in appropriate places, an extension to three-dimensional workspaces is largely straightforward.) An Example of a polyomino is illustrated in Fig. 2. Pixels in the planar grid not belonging to P are *blocked*: They form obstacles for particles that stop the motion from an adjacent pixel.

The particles are commanded in unison: In each step, all particles are relocated by one unit in one of the directions "Up" (u), "Down" (d), "Left" (l), or "Right" (r), unless the destination is a blocked pixel; in this case, a particle remains in its previous pixel. A motion plan is a command sequence $C = \langle c_1, c_2, c_3, \dots \rangle$, where each command $c_i \in \{u, d, l, r\}$. For a command sequence C and a non-negative integer ℓ, we denote the command sequence consisting of ℓ repetitions of C by C^ℓ.

Because the particles are small, many of them can be located in the same pixel. During the course of a command sequence, two particles π_1 and π_2 may end up in the same pixel p, if π_1 moves into p, while π_2 remains in p due to a blocked pixel. Once two particles share a pixel, any subsequent command will relocate them in unison—they will not be separated, so they can be considered to be *merged*.

The distance $\mathrm{dist}(p, q)$ between two pixels p and q is the length of a shortest path on the integer grid between p and q that stays within P. The *diameter* of a polyomino P describes the maximum distance between any two of its pixels; we denote it by D.

A *configuration* of P is a set of pixels containing at least one particle. The set of all possible configurations of P is denoted by $\mathcal{P}$. We call a command sequence *gathering* if it transforms a configuration $A \in \mathcal{P}$ into a configuration A' such that $|A'| = 1$, i.e., if it merges all particles in the same pixel.

III. Algorithmic Approaches

In this section, we investigate several algorithmic approaches for two-dimensional scenarios. As the main focus of this paper is the practical relevance and applicability of the overall challenge, the theoretical details are omitted due to limited space. We start by showing that the problem is computationally hard – for several variants.

A. The problem is hard

We show that the following decision problem, which we call Min-Gathering, is hard: Given a polyomino P and a set of particles, is there a gathering sequence of length ℓ?

Theorem 1. Min-Gathering *is NP-hard, even for the case of polyominoes.*

Proof. We reduce from 3-SAT, i.e., the problem of deciding of whether a given boolean formula has a truth assignment; for more background see [12]. For every instance Φ of 3-SAT, we construct a polyomino P_Φ as follows: For every variable, we insert a variable gadget as indicated in Fig. 2. We join all variable gadgets vertically in row to a *variable block*; we call the top row of each variable gadget its *variable row*. For every clause, we construct a clause gadget that contains a left (right) arm for each incident positive (negative) literal in the corresponding variable row and an exit arm in the bottom. To obtain P_Φ, we join all clause gadgets from left to right by a *bottom row* and insert a variable block at the left and right end of the bottom row. For an illustration, consider Fig. 2.

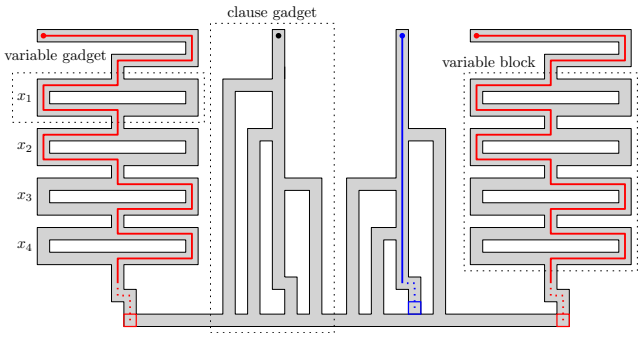

Fig. 2. The polyomino P_Φ for the 3-SAT-instance $\Phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_3 \vee x_4)$. A sequence that merges the two red particles with $\frac{1}{2}(D+b)$ commands corresponds to a variable assignment of Φ.

Let I be the instance of MIN-GATHERING consisting of P_Φ where the top row is filled with particles. We call the two leftmost particles above the variable blocks, the *red* particles and denote the length of the bottom row by b. Note that the distance between the red particles is the diameter D.

Claim. *I has a gathering sequence of length $\ell := \frac{1}{2}(D+b)$ if and only if Φ is satisfiable.*

Details for this claim are omitted for space reasons; they can be found in the full version of our paper (to appear). □

Note that the left pixel of the bottom row is one of two possible merge location for a gathering sequence of length $\frac{1}{2}(D+b)$. Therefore, the same reduction shows that problem remains hard if a target location is prescribed. In fact, an even stronger statement holds true: An instance of the polyomino P_Φ where all pixels are filled has a gathering sequence of length $\frac{1}{2}(D-b)$ if and only if Φ is satisfiable. This implies that the decision problem of ROBOT LOCALIZATION is also hard. In an instance of this problem, we are given a sensorless robot r in a polyomino, and wonder whether there exists a command sequence of length ℓ such that we know the position of r afterwards. The above observations yield:

Corollary 2. ROBOT LOCALIZATION *is NP-hard.*

B. Merging Two Particles

We start with a special class of polyominoes. We call a polyomino P *simple* if decomposing P with horizontal lines through pixel edges results in a set of rectangles $\mathcal{R}$ such that the edge-contact graph $\mathcal{C}(\mathcal{R})$ of $\mathcal{R}$ is a tree. The edge-contact graph of a set of rectangles in the plane contains a vertex for each rectangle and an edge for each side contact; a corner contact does not result in an edge. A *hole* of a polyomino P is a maximal set of blocked cells (cells not contained in P) that are connected such that there exists a closed walk within P surrounding it. As usual, simplicity of a polyomino captures the feature of not containing holes. A *shortest path from a pixel p in P to a rectangle R in $\mathcal{R}$* is a shortest path from p to a pixel q in R such that $\mathrm{dist}(p,q)$ is minimal.

Theorem 3. *For any two particles in a simple polyomino P, there exists a gathering sequence of length D.*

Proof. Let $\mathcal{R}$ be a decomposition of P into rectangles by cutting P with horizontal lines through pixel edges. Then, because P is simple, the edge-contact graph $\mathcal{C}(\mathcal{R})$ of the rectangles $\mathcal{R}$ is a tree. For an example, consider Fig. 3.

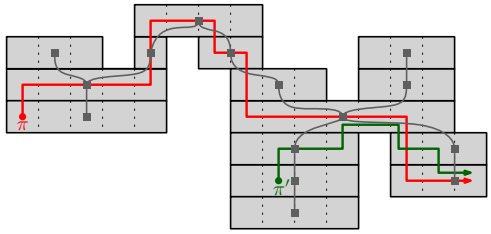

Fig. 3. A simple polyomino P, and its edge-contact graph $\mathcal{C}(\mathcal{R})$ (in gray). When the red particle π moves towards the green particle π', π and π' follow the respective red and green paths. The dotted lines separate the pixels.

For every t, let R_t and R'_t be the rectangles of P containing the two particles π and π' after applying t commands, respectively. Moreover, let S_t be a shortest path from R_t to R'_t in $\mathcal{C}(\mathcal{R})$; and let $S_t(1)$ be the successor of R_t on S_t (if it exists, i.e., $R_t \neq R'_t$).

We use the following strategy:

Phase 1: While $R_t \neq R'_t$, compute a shortest path S_t from R_t to R'_t in $\mathcal{C}(\mathcal{R})$. Move π to $S_t(1)$ via a shortest path in P. Update R_t and R'_t.

Phase 2: If $R_t = R'_t$, merge π and π' by moving π towards π' by a shortest (horizontal) path; note that this gathering sequence merges the particles within R_t.

In fact, the resulting sequence has the following property; details of the proof are omitted due to space limits.

Claim. *For every $s > t$, the rectangles R_s and R'_s are either equal to R_t or lie in the connected component C of $\mathcal{C}(\mathcal{R} \setminus R_t)$ containing R'_t.*

This claim implies that the merge location and R'_t lie in C or are equal to R_t. Consequently, in every step, π moves towards the merge location on a shortest path and thus that the gathering sequence is at most of length D. □

In the remainder, we call the strategy used to prove Theorem 3 DYNAMICSHORTESTPATH (DSP): Move one particle towards the other along a shortest path; update the shortest path if a shorter one exists. The example in Fig. 4 shows that DSP may perform significantly worse in non-simple polyominoes.

Proposition 4. *The strategy DSP may not yield a gathering sequence of length $O(D)$ in non-simple polyominoes.*

Proof. By the symmetry of P, the distance between the two particles decreases for the first time when one of them is at the left or right side of P. Therefore, denoting the number of holes by H where each hole is of height h and width w as indicated in Fig. 4, the length of the gathering sequence C is $H(6h + w) + 3$, while the diameter is bounded by $D \leq (H - 2)w + 6h + 2w + 4 = Hw + 6h + 4$. Choosing $h := cw/6$ for some constant $c \geq H$, the ratio of $|C|$ and $|D|$ can be arbitrarily large: $\frac{cHw+Hw+3}{Hw+cw+4} \geq \frac{H(c+1)}{H+c+1} \geq \frac{H}{2}$. □

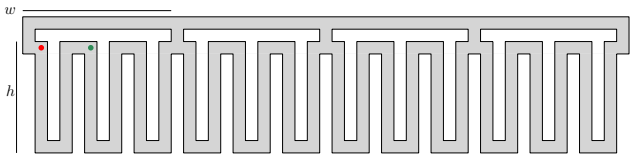

Fig. 4. When the red particle π moves towards the green particle π' by shortest paths, π visits the entire bottom path.

Nevertheless, DSP always merges two particles; the proof is omitted due to limited space.

Proposition 5. *For every polyomino P with n pixels and diameter D and every configuration with two particles, DSP yields a gathering sequence of length $O(nD)$.*

Using a different strategy yields a better bound: The strategy MOVETOEXTREMUM (MTE) iteratively moves an extreme particle (e.g. bottom-leftmost) to an opposite extreme pixel (e.g. top-rightmost) along a shortest path.

Theorem 6. *For any two particles in a polyomino P, MTE yields a gathering sequence of length at most D^2.*

Proof. Let q be the top-rightmost pixel of P. To merge the two particles in q, our strategy is as follows: Identify the particle π that is bottom-leftmost. Apply a command sequence that moves π to q on a shortest path. Repeat.

Claim. *In each iteration, the sum of the distances Δ of the two particles to q decreases.*

Note that Δ decreases when the other particle π' has a collision. If π' had no collision, there exist a pixel that is higher or more to the right than q, contradicting the choice of q. Consequently, the sum of distances Δ, which is at most $2D$ at start, decreases at least by 1 for every D steps. Hence after $O(D^2)$ steps, Δ is reduced to 0. □

Note that there exist polyominoes, e.g., a square, where the number of pixels n is in $\Omega(D^2)$. Therefore, Theorem 6 significantly improves the bound of $O(n^3)$ in [14].

Finally, we note that a shortest gathering sequence for two particles in a non-simple polyomino may need to exceed D.

Proposition 7. *Let P be a polyomino with two particles. A shortest gathering sequence may be of length $\frac{3}{2}D - O(\sqrt{D})$.*

See Fig. 5 for the idea; technical proof details are omitted due to limited space.

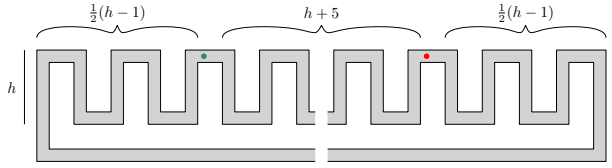

Fig. 5. A polyomino consisting of a base and S chimneys.

C. Reducing the number of particles

Now we show how to significantly decrease the number of particles with few commands to a parameter proportional to the complexity of the polyomino, namely the number of convex corners. This is particularly relevant for establishing the existence of *oblivious* gathering strategies that are capable of merging all particles in an efficient manner, even if their initial configuration is not known. (See Section IV-C.)

Lemma 8. *Let P be a polyomino with diameter D and k convex corners. For every configuration $A \in \mathcal{P}$, there exists a command sequence of length $2D$ which transforms A to a configuration $A' \in \mathcal{P}$ such that $|A'| \leq k/4$.*

Proof. We distinguish four types of convex corners; northwest (NW), northeast (NE), southwest (SW), southeast (SE). By the pigeon hole principle, one of the types occurs at most $k/4$ times; without loss of generality, let this be the NW corners.

We show that after applying the sequence $\langle l, u\rangle^D$, every particle lies in a NW corner: Consider a particle π in pixel p. Unless π lies in a NW corner, it moves for at least one command in $\{l, u\}$. Because P is finite, there exists an ℓ large enough such that π ends in a NW corner q when the command sequence $\langle l, u\rangle^\ell$ is applied, i.e., there exists an pq-path consisting of at most ℓ commands of types l and u, respectively. Because a monotone path is a shortest path, it holds that $\ell \leq D$. □

D. General Upper Bounds

Combining Lemma 8 and Theorem 3 yields:

Corollary 9. *For a set of particles in a simple polyomino P with diameter D and k convex corners, there exists a gathering sequence of length $O(kD)$.*

Lemma 8 and Theorem 6 imply the following fact:

Corollary 10. *For any set of particles in a polyomino P with diameter D and k convex corners, there exists a gathering sequence of length at most $O(kD^2)$.*

By analyzing cuboids instead of rectangles, six directions of motion instead of four, and corners in eight quadrant

directions instead of four, we obtain the analogous result for three-dimensional settings. Details are omitted from this short paper.

IV. Evaluation in Simulation

A. Overview of Evaluated Approaches

In this section, we evaluate the performance of the following approaches on practical instances in simulation.

- The approach StaticShortestPath (SSP) iteratively merges pairs of particles by moving one to the position of the other along a shortest path, see Alg. 2 in [14].
- The approach DynamicShortestPath (DSP).
- The approach MoveToExtremum (MTE). Among the eight options, we choose an extremum that minimizes the initial sum of distances to both particles.
- The heuristic MinSumToExtremum (MSTE) generalizes the idea of MTE. It selects an extremum with the smallest initial sum of distances to all particles and iteratively performs a command that decreases this sum the most. If no command decreases the sum, two particles are selected and merged by MTE. Afterwards, MSTE resumes.
- Additionally, we evaluate a machine learning approach ReinforcementLearning (RL) based on a deep learning network for Q-learning that is trained via reinforcement learning to solve an instance; for details, we refer to Section IV-D.

In addition to the commands $\{u, d, l, r\}$, we also allow diagonal motions in the experiments. Moreover, a target location for the particles is prescribed. While the strategy ReinforcementLearning directly supports this, in all other strategies, the particles are merged in any location of the polyomino and then transported to the target location along a shortest path in unison.

For the strategies SSP, DSP, and MTE, a significant parameter is the choice of the next pair of particles to be merged. For these strategies, we evaluate the options of (a) choosing a pair uniformly at random (RandomPair) or (b) choosing the pair with maximal distance (DistantPair).

B. Simulation Results

We evaluated our approaches on the three polyominoes depicted in Fig. 6 that are inspired by vascular networks. For each algorithm and polyomino, we carried out at least 128 trials with exactly 1000 randomly distributed distinct particles that were to be gathered in a target pixel.

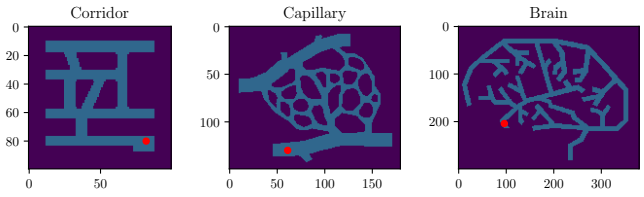

Fig. 6. The polyominoes `Corridor`, `Capillary`, and `Brain` on which we evaluate the approaches. Target locations are indicated by a red dot.

Overall, ReinforcementLearning shows a significantly better performance than the other approaches, see Fig. 7; note that this comes at the expense of significant time spent on local optimization by carrying out extensive training for each individual polyomino, while the combinatorial algorithms takes considerably less computation time. Among these, DistantPair show on average a better performance than RandomPair for nearly all instances and algorithms. Moving particles to a corner first, as suggested by Lemma 8,

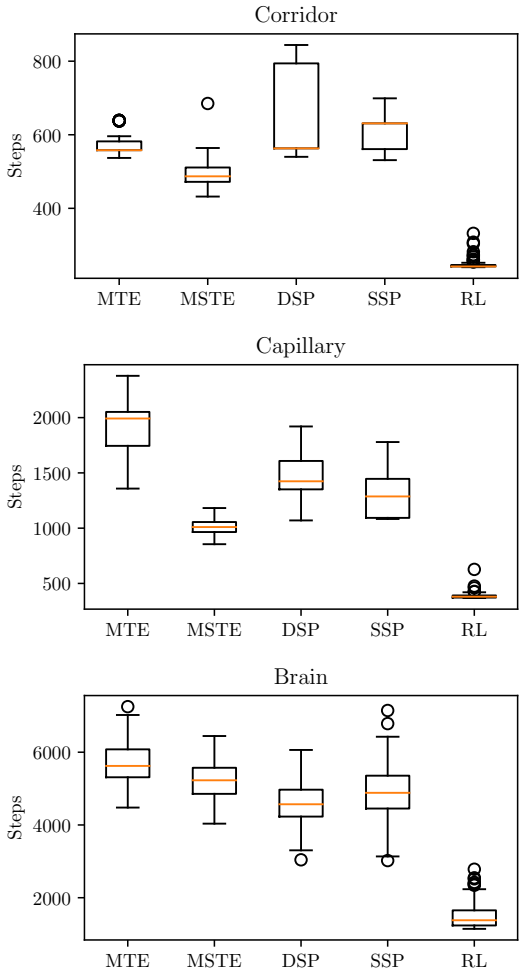

Fig. 7. Comparison of the algorithms on the three environments with 1000 uniformly random particles. The boxes show the upper and lower quartile, the whiskers the range, the orange line the median, and the circles the outliers. For `Brain`, RandomPair is shown for SSP, DSP, and MTE; otherwise, DistantPair is shown.

most of the time led to an increase in steps. This is due to most steps being used to merge the last few remaining particles, as discussed in the next section.

C. Oblivious Merging

In practice, it may be expensive to determine the position of the individual particles; therefore, *oblivious* approaches that do not need this information may be of interest. Such a setting is equivalent to the situation where initially, each pixel contains a particle; a gathering sequence for all particles is

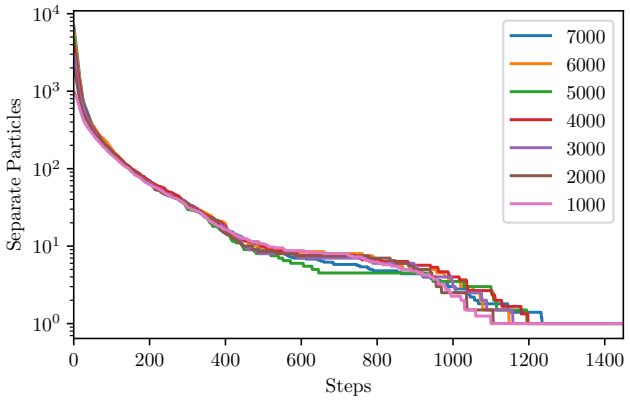

Fig. 8. Number of particle groups over time using MINSUMTOEXTREMUM on `Capillary` (7169 pixel). This shows that the number of start particles (7000, 6000, ..., 1000) has negligible impact on the number of steps needed. Collecting larger amounts of particles can be slightly quicker in some cases due to the involved randomness and the non-optimal method.

certainly a gathering sequence for any other (partial) initial distribution of particles. Recall that Corollary 2 implies that this problem remains NP-hard. In order to estimate the cost of this restriction in practice, we study how the number of populated grid cells behaves over time, depending on the initial number of particles; see Figs. 8 and 9.

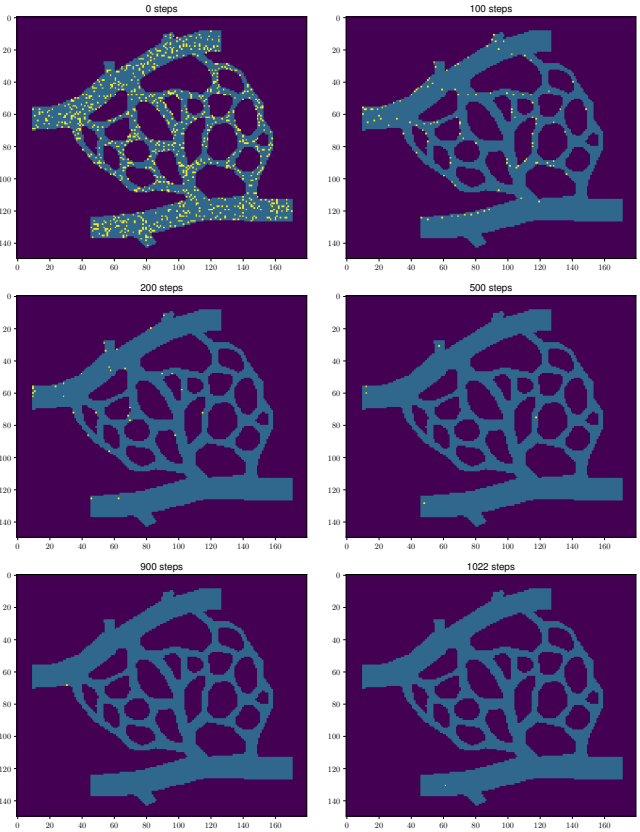

Fig. 9. The process of gathering 1000 particles in the target location with MINSUMTOEXTREMUM. Gathered particles aggregate to a single particle.

Because the number of populated grid cells decreases very sharply in the beginning and almost all steps are used to merge the few remaining groups of particles, we can conclude that missing knowledge of the position of the individual particles has negligible cost for uniform distributions.

D. Deep Learning Implementation

The reinforcement learning approach uses the synchronous Advantage Actor-Critic (A2C) method combined with an intrinsic curiosity mechanism (ICM). We use the OpenAI implementation of A2C [9] with slightly different data preprocessing and hyperparameter settings. The environment wrapper begins by applying sticky actions and max pooling and then scales the gray-scale image to a 84×84 format. Then the neural network is fed a stack of four successive frames.

The feature extraction uses four convolutional layers with 32 $(8 \times 8, s = 4)$, 64 $(4 \times 4, s = 2)$, and 64 $(2 \times 2, s = 1)$ filters, respectively. The output of each layer is activated by a leaky rectified linear unit (Leaky ReLU). After flattening, the output of the last convolutional layer is mapped to the policy (dimension = 4 or 8, depending on available actions) via a fully connected layer (512 units). The value function is also mapped from the last convolutional layer, with output dimension 1. A2C employs 128 parallel agents with different particle distributions to collect experience. The learning rate is set to 0.0001. Each agent collects 2048 rollouts (steps) before the four-epoch update in network weights. During each update, the mini batch size is set to 32.

V. CONCLUSIONS

We have described a spectrum of methodological progress on an important problem of great practical relevance. This exposition focuses on two-dimensional scenarios, but a generalization to three-dimensional settings appears to be straightforward. In addition, we point out three other relevant directions for future research.

Firstly, our algorithmic simulations indicate the strength of our methods. However, the different outcomes for deterministic as well as ML approaches indicate that further, more detailed algorithmic studies are warranted to understand the most successful line of attack; this includes studies of the necessary tradeoff between computation time and number of actuation steps, but also includes modified models in which an actuation step may be able to move particles by more than an elementary distance. Secondly, how can we deal with random errors in actuation and navigation? Our insights into oblivious methods clearly indicate that these should remain tractable, but more detailed considerations for frequency and amount of errors should provide quantifications and error-correcting approaches. Finally, it is typically not necessary for our application scenarios to gather *all* particles in a target area; moving an appropriate fraction should usually suffice. Fig. 8 visualizes a slightly different aspect, but still highlights the prospect that a considerably reduced number of actuation steps may be achieved.

References

[1] S. Alpern and S. Gal, *The theory of search games and rendezvous*, ser. International Series in Operations Research and Management Science. Boston, Dordrecht, London: Kluwer Academic Publishers, 2003.

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

[3] J. Balanza-Martinez, A. Luchsinger, D. Caballero, R. Reyes, A. A. Cantu, R. Schweller, L. A. Garcia, and T. Wylie, "Full tilt: Universal constructors for general shapes with uniform external forces," in *30th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019, pp. 2689–2708.

[4] A. T. 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)*, 2015, pp. 16–18.

[5] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6751–6756.

[6] A. T. 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 (ALGOSENSORS)*, 2014, pp. 51–66.

[7] A. T. Becker, E. D. Demaine, S. P. Fekete, J. Lonsford, and R. Morris-Wright, "Particle computation: Complexity, algorithms, and logic," *Natural Computing*, vol. 18, no. 1, pp. 181–201, 2019.

[8] A. T. Becker, S. P. Fekete, P. Keldenich, D. Krupke, C. Rieck, C. Scheffer, and A. Schmidt, "Tilt assembly: algorithms for micro-factories that build objects with uniform external forces," *Algorithmica*, pp. 1–23, 2017.

[9] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *arXiv:1808.04355*, 2018.

[10] 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.

[11] P. Flocchini, "Gathering," in *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, P. Flocchini, G. Prencipe, and N. Santoro, Eds. Springer, 2019, pp. 63–82.

[12] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[13] 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.

[14] A. V. Mahadev, D. Krupke, J.-M. Reinhardt, S. P. Fekete, and A. T. Becker, "Collecting a swarm in a 2D environment using shared, global inputs," in *13th Conference on Automation Science and Engineering (CASE)*, 2016, pp. 1231–1236.

[15] 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.

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

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

[18] 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.

[19] Y. Zhang, X. Chen, H. Qi, and D. Balkcom, "Rearranging agents in a small space using global controls," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3576–3582.

[20] Y. Zhang, E. Whiting, and D. Balkcom, "Assembling and disassembling planar structures with divisible and atomic components," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 945–954, 2018.