

Targeted Drug Delivery: Advanced 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 advanced 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 advanced algorithmic approaches that greatly improve the worst-case guarantees for the number of required actuation steps. We validate 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.

^{*}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.

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

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

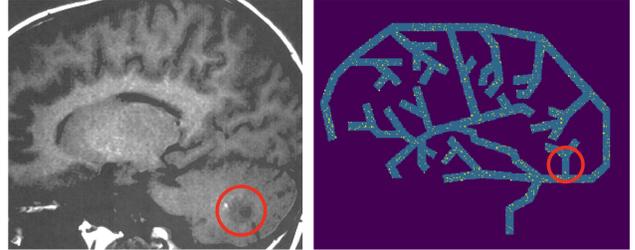


Fig. 1. (Left) An MRI image of a brain tumor (marked by the red circle), located in the cerebellum. (Right) How can the swarm of particles (indicated by yellow dots) be delivered to the target region?

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 are subjected to the same direction and distance of motion, 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 [?] 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. While this shows that delivery can always be achieved, a delivery time of this magnitude is usually impractical, as the number of grid cells may be prohibitively large. As a consequence, this warrants more research on this (literally vital) topic.

Our Contribution

We provide a number of important insights:

- We prove that minimizing the length of a command sequence for gathering all particles is NP-hard, even for an environment that consists of a planar arrangement of grid cells, explaining the observed difficulty of the problem. Our reduction implies hardness for the related localization problem.
- We develop an algorithmic strategy for collecting all particles in one destination, 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.
- 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 and demonstrating the practical suitability of our algorithmic approaches.

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 [?], [?]. An alternative are aggregates of *superparamagnetic iron oxide microparticles*, $9 \mu\text{m}$ particles that are used as a contrast agent in MRI studies [?]. 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 [?], [?]. 3D Maxwell-Helmholtz coils are often used for precise magnetic field control [?]. 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 [?], [?], [?]; see video and abstract [?] for a visualizing overview. For a recent survey on challenges related to controlling multiple microrobots (less than 64 robots at a time), see [?].

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 [?] introduced a wide range of models and methods for this concept as have Anderson and Fekete [?] 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 [?]. 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 [?]. 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 [?]; 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.

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.) Examples of polyominoes are illustrated in Figs. 2 to 5. 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 $\text{dist}(p, q)$ 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 clarity of exposition, we focus on two-dimensional scenarios; a generalization to three-dimensional settings can be achieved with a limited amount of additional work. As the main focus of this paper is the practical relevance and applicability of the overall challenge, these 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 ?

186 **Theorem 1.** MIN-GATHERING is NP-hard, even for the case
 187 of polyominoes.

188 *Proof.* We reduce from 3-SAT. For every instance Φ of 3-
 189 SAT, we construct a polyomino P_Φ as follows: For every
 190 variable, we insert a variable gadget as indicated in Fig. 2.
 191 We join all variable gadgets vertically in row to a *variable*
 192 *block*; we call the top row of each variable gadget its
 193 *variable row*. For every clause, we construct a clause gadget
 194 that contains a left (right) arm for each incident positive
 195 (negative) literal in the corresponding variable row and an
 196 exit arm in the bottom. To obtain P_Φ , we join all clause
 197 gadgets from left to right by a *bottom row* and insert a
 198 variable block at the left and right end of the bottom row.
 199 For an illustration, consider Fig. 2.

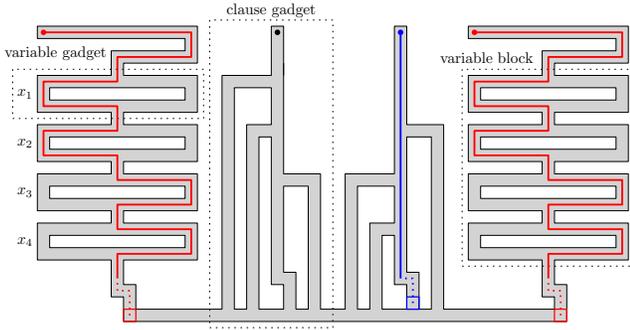


Fig. 2. The polyomino P_Φ for the 3-SAT-instance $\Phi = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3 \vee x_4)$. A sequence that merges the two red particles with $\frac{1}{2}(D+b)$ commands corresponds to a variable assignment of Φ .

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

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

207 Details for this claim (and thus, a full proof of Theorem 1)
 208 are omitted for space reasons; they can be found in the full
 209 version of our paper. \square

210 Note that the left pixel of the bottom row is one of two
 211 possible merge location for a gathering sequence of length
 212 $\frac{1}{2}(D+b)$. Therefore, the same reduction shows that problem
 213 remains hard if a target location is prescribed. In fact, an even
 214 stronger statement holds true: An instance of the polyomino
 215 P_Φ where all pixels are filled has a gathering sequence of
 216 length $\frac{1}{2}(D-b)$ if and only if Φ is satisfiable. This implies
 217 that the decision problem of ROBOT LOCALIZATION is also
 218 hard. In an instance of this problem, we are given a sensorless
 219 robot r in a polyomino, and wonder whether there exists
 220 a command sequence of length ℓ such that we know the
 221 position of r afterwards. The above observations yield:

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

B. Merging Two Particles

223 We start with a special class of polyominoes. We call a
 224 polyomino P *simple* if decomposing P with horizontal lines
 225 through pixel edges results in a set of rectangles \mathcal{R} such
 226 that the edge-contact graph $\mathcal{C}(\mathcal{R})$ of \mathcal{R} is a tree. The edge-
 227 contact graph of a set of rectangles in the plane contains a
 228 vertex for each rectangle and an edge for each side contact;
 229 a corner contact does not result in an edge. A *hole* of a
 230 polyomino P is a maximal set of blocked cells (cells not
 231 contained in P) that are connected such that there exists a
 232 closed walk within P surrounding it. As usual, simplicity of
 233 a polyomino captures the feature of not containing holes. A
 234 *shortest path* from a pixel p in P to a rectangle R in \mathcal{R}
 235 is a shortest path from p to a pixel q in R such that $\text{dist}(p, q)$
 236 is minimal. 237

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

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

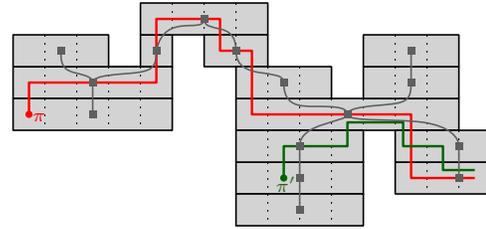


Fig. 3. A simple polyomino P , and its edge-contact graph $\mathcal{C}(\mathcal{R})$ (in gray). When the red particle π moves towards the green particle π' , π and π' follow the respective red and green paths.

243 For every t , let R_t and R'_t be the rectangles of P
 244 containing the two particles π and π' after application of t
 245 commands, respectively. Moreover, let S_t be a shortest path
 246 from R_t to R'_t in $\mathcal{C}(\mathcal{R})$; and let $S_t(1)$ be the successor of R_t
 247 (if it exists, i.e., $R_t \neq R'_t$). 248

249 We pursue the following strategy:

250 **Phase 1:** While $R_t \neq R'_t$, compute a shortest path S_t from
 251 R_t to R'_t in $\mathcal{C}(\mathcal{R})$. Move π to $S_t(1)$ via a shortest path in P .
 252 Update R_t and R'_t . 253

254 **Phase 2:** If $R_t = R'_t$, merge π and π' by moving π
 255 towards π' by a shortest (horizontal) path; note that this
 256 gathering sequence merges the particles within R_t . 257

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

258 **Claim.** For every $s > t$, the rectangles R_s and R'_s are either
 259 equal to R_t or lie in the connected component C of $\mathcal{C}(\mathcal{R} \setminus R_t)$
 260 containing R'_t . 261

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

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

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

273 *Proof.* By the symmetry of P , the distance between the two
 274 particles decreases for the first time when one of them is at
 275 the left or right side of P . Therefore, denoting the number
 276 of holes by H where each hole is of height h and width w
 277 as indicated in Fig. 4, the length of the gathering sequence
 278 C is $H(6h + w) + 3$, while the diameter is bounded by
 279 $D \leq (H - 2)w + 6h + 2w + 4 = Hw + 6h + 4$. Choosing
 280 $h := cw/6$ for some constant $c \geq H$, the ratio of $|C|$ and $|D|$
 281 can be arbitrarily large: $\frac{cHw + Hw + 3}{Hw + cw + 4} \geq \frac{H(c+1)}{H+c+1} \geq \frac{H}{2}$. \square

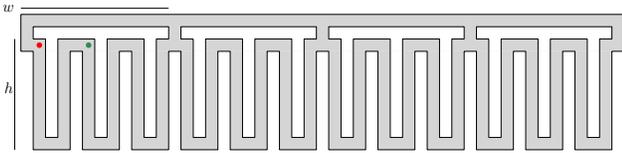


Fig. 4. When the red particle π moves towards the green particle π' by shortest paths, π visits the entire bottom path.

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

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

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

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

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

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

299 Note that Δ decreases when the other particle π' has a
 300 collision. If π' had no collision, there exist a pixel that is
 301 higher or more to the right than q , contradicting the choice of
 302 q . Consequently, the sum of distances Δ , which is at most $2D$
 303 at start, decreases at least by 1 for every D steps. Hence after
 304 $O(D^2)$ steps, Δ is reduced to 0. \square

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

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

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

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

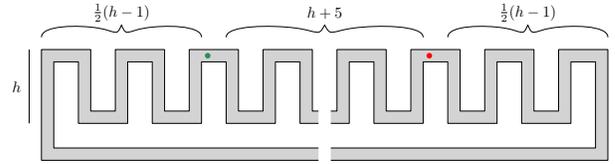


Fig. 5. A polyomino consisting of a base and S chimneys.

314 **C. Reducing the number of particles**

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

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

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

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

340 **D. General Upper Bounds**

341 Combining Lemma 8 and Theorem 3 yields:

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

345 Lemma 8 and Theorem 6 imply the following fact:

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

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

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

354 IV. EVALUATION IN SIMULATION

355 A. Overview of Evaluated Approaches

356 In this section, we evaluate the performance of the follow-
 357 ing approaches on practical instances in simulation.

- 358 • The approach `STATICSHORTESTPATH` (SSP) iteratively
 359 merges pairs of particles by moving one to the position
 360 of the other along a shortest path, see Alg. 2 in [?].
- 361 • The approach `DYNAMICSHORTESTPATH` (DSP).
- 362 • The approach `MOVETOEXTREMUM` (MTE). Among
 363 the eight options, we choose an extremum that mini-
 364 mizes the initial sum of distances to both particles.
- 365 • The heuristic `MINSUMTOEXTREMUM` (MSTE) gener-
 366 alizes the idea of MTE. It selects an extremum with
 367 the smallest initial sum of distances to all particles
 368 and iteratively performs a command that decreases this
 369 sum the most. If no command decreases the sum, two
 370 particles are selected and merged by MTE. Afterwards,
 371 MSTE resumes.
- 372 • Additionally, we evaluate a machine learning approach
 373 `REINFORCEMENTLEARNING` (RL) based on a deep
 374 learning network for Q-learning that is trained via
 375 reinforcement learning to solve an instance; for details,
 376 we refer to Section IV-D.

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

384 For the strategies SSP, DSP, and MTE, a significant
 385 parameter is the choice of the next pair of particles to be
 386 merged. For these strategies, we evaluate the options of
 387 (a) choosing a pair uniformly at random (`RANDOMPAIR`) or
 388 (b) choosing the pair with maximal distance (`DISTANTPAIR`).

389 B. Simulation Results

390 We evaluated our approaches on the three polyominoes
 391 depicted in Fig. 6 that are inspired by vascular networks.
 392 For each algorithm and polyomino, we carried out at least
 393 128 trials with exactly 1000 randomly distributed distinct
 394 particles that were to be gathered in a target pixel.

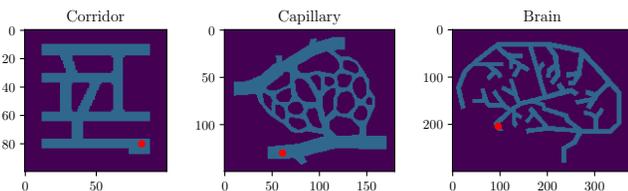


Fig. 6. The polyominoes `Corridor`, `Capillary`, and `Brain` on which we evaluate the approaches. Target locations are indicated by a red dot.

Overall, `REINFORCEMENTLEARNING` shows a significantly better performance than the other approaches, see Fig. 9; 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, 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.

407 C. Oblivious Merging

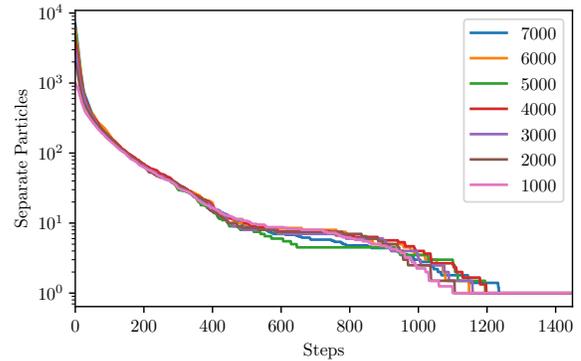


Fig. 7. Number of particle groups over time using `MINSUMTOEXTREMUM` on `Capillary` (7169 pixel). This shows that the number of start particles (7000, 6000, ..., 1000) has negligible impact on the number of steps needed. Collecting larger amounts of particles can be slightly quicker in some cases due to the involved randomness and the non-optimal method.

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 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. 7 and 8.

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.

424 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 [?] with slightly different data pre-processing 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.

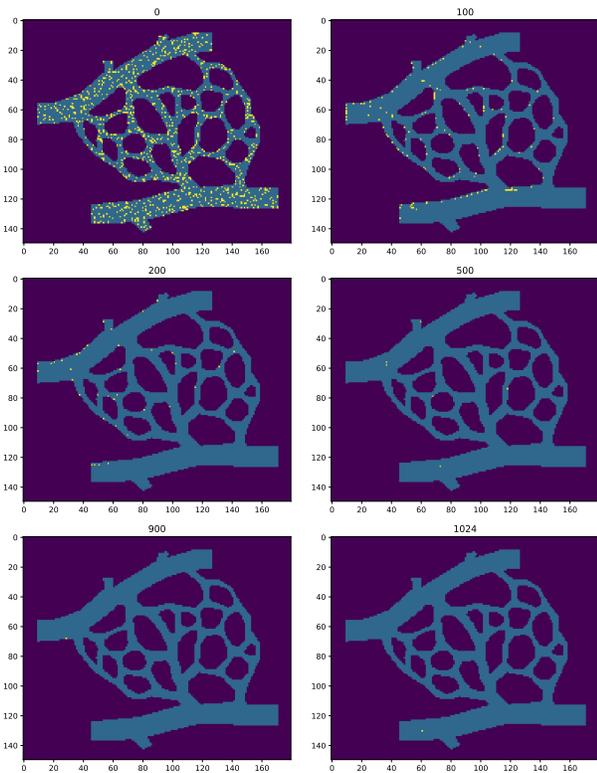


Fig. 8. The process of gathering 1000 particles in the target location with MINSUMTOEXTREMUM.

432 Then the neural network (with architecture as depicted in
433 Fig. 10) is fed a stack of four successive frames.

434 The feature extraction uses four convolutional layers with
435 32 ($8 \times 8, s = 4$), 64 ($4 \times 4, s = 2$), and 64 ($2 \times 2, s = 1$)
436 filters, respectively. The output of each layer is activated by
437 a leaky rectified linear unit (Leaky ReLU). After flattening,
438 the output of the last convolutional layer is mapped to the
439 policy (dimension = 4 or 8, depending on available actions)
440 via a fully connected layer (512 units). The value function is
441 also mapped from the last convolutional layer, with output
442 dimension 1. A2C employs 128 parallel agents with different
443 particle distributions to collect experience. The learning rate
444 is set to 0.0001. Each agent collects 2048 rollouts (steps)
445 before the four-epoch update in network weights. During
446 each update, the mini batch size is set to 32.

447 V. CONCLUSIONS

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

454 Firstly, our algorithmic simulations indicate the strength
455 of our methods. However, the different outcomes for deter-
456 ministic as well as ML approaches indicate that further, more
457 detailed algorithmic studies are warranted to understand the
458 most successful line of attack; this includes studies of the

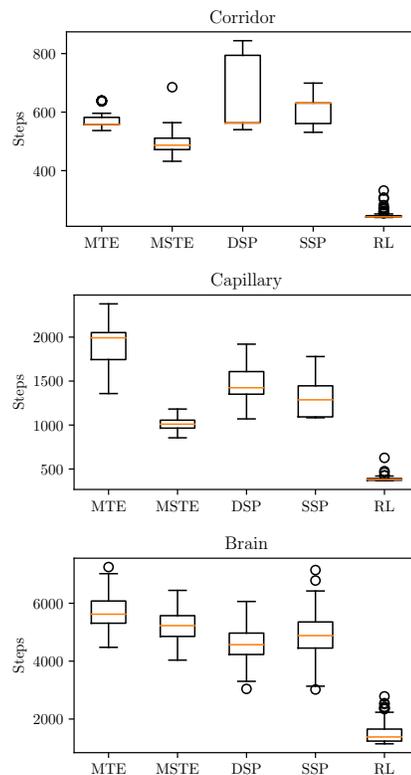


Fig. 9. Comparison of the algorithms on the three environments with 1000 uniformly random particles. The boxes show the upper and lower quartile, the whiskers the range, the orange line the median, and the circles the outliers. For Brain, RANDOMPAIR is shown for SSP, DSP, and MTE; otherwise, DISTANTPAIR is shown.

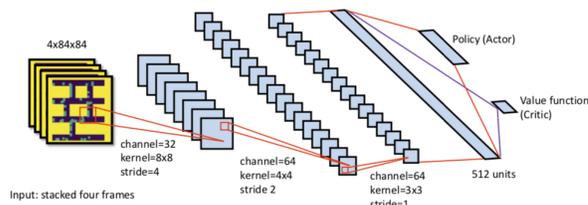


Fig. 10. Illustration of A2C neural network architecture

459 necessary tradeoff between computation time and number of
460 actuation steps, but also includes modified models in which
461 an actuation step may be able to move particles by more
462 than an elementary distance. Secondly, how can we deal with
463 random errors in actuation and navigation? Our insights into
464 oblivious methods clearly indicate that these should remain
465 tractable, but more detailed considerations for frequency and
466 amount of errors should provide quantifications and error-
467 correcting approaches. Finally, it is typically not necessary
468 for our application scenarios to gather *all* particles in a target
469 area; moving an appropriate fraction should usually suffice.
470 Fig. 7 visualizes a slightly different aspect, but still highlights
471 the prospect that a considerably reduced number of actuation
472 steps may be achieved.