Tilt Automata: Gathering Particles With Uniform External Control

Sándor P. Fekete*[1], Jonas Friemel*[2], Peter Kramer*[3], Jan-Marc Reinhardt[2], Christian Rieck†[4], and Christian Scheffer*[2]

1 Department of Computer Science, TU Braunschweig and L3S, Germany
`s.fekete@tu-bs.de`

2 Department of Electrical Engineering and Computer Science, Bochum University of Applied Sciences, Germany
`{jonas.friemel,jan-marc.reinhardt,christian.scheffer}@hs-bochum.de`

3 Department of Computer Science, TU Braunschweig, Germany
`kramer@ibr.cs.tu-bs.de`

4 Institute of Mathematics, University of Kassel, Germany
`christian.rieck@mathematik.uni-kassel.de`

Abstract

Motivated by targeted drug delivery, we investigate the gathering of particles in the full tilt model of externally controlled motion planning: A set of particles is located at the tiles of a polyomino. They react uniformly to an external force by moving maximally in one of the four axis-parallel directions until they hit the boundary. The goal is to choose a sequence of directions that moves all particles to a common position. We show how to compute gathering sequences for filled polyominoes and prove it PSPACE-complete to determine if the particles in a partially filled polyomino can be gathered. The results build on a connection we establish between tilt models and synchronizing automata.

Related Version *Full Version*: `arXiv:2603.02796` [13]

1 Introduction

We investigate a problem motivated by targeted drug delivery [15]: How can initially dispersed drug-carrying particles be gathered at a common position? In micro-scale robotics [16], environmental constraints and the severely limited capabilities of the robotic agents rule out autonomous movement. Instead, a global external force, e.g., an electromagnetic field, can be used to move and steer simple robotic particles. The challenge is to use geometry to control a large swarm of particles through uniform signals within a non-uniform environment [3, 7].

Abstract models [2, 8], commonly called *tilt models*, have been developed to tackle this problem algorithmically. They model the environment as a *polyomino*, i.e., a finite, connected region of the square tiling, with particles positioned at tiles and moving on axis-parallel paths. Two main variants exist: In the *single step model* (S1), particles move via unit steps to adjacent positions, whereas in the *full tilt model* (FT), they move maximally until blocked.

Algorithms and complexity results for a variety of applications [6, 9, 14, 17] have been discovered using tilt models. We consider particles located at the tiles of a polyomino and look for a *gathering sequence*, i.e., a sequence of moves that relocates all particles to the same tile. In S1, gathering sequences of length $\mathcal{O}(n_c D^2)$ always exist, where n_c is the number of convex corners of the polyomino and D its diameter, but finding sequences of minimum length is NP-hard [4]. On the other hand, very little is known about gathering in FT.

---

* Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 530918134.
† Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 522790373.

42nd European Workshop on Computational Geometry, Hagen, Germany, March 25–27, 2026.
This is an extended abstract of a presentation given at EuroCG'26. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

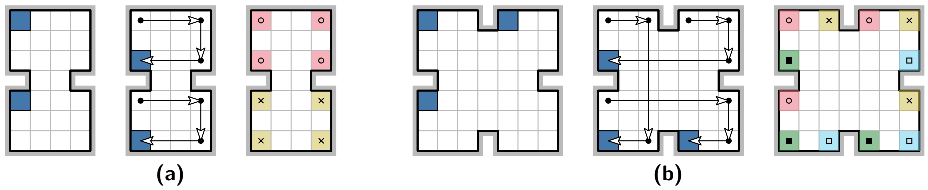

Figure 1 Two examples where particles cannot be gathered. Both depict, from left to right, a set of particles in a polyomino, moves leading to all possible configurations, and congruent positions.

Observe that particles cannot always gather in FT. The authors of [18] note that particles may be trapped in separate regions, see Figure 1(a). But gathering can even be impossible if all tiles reachable by the initial particles are mutually reachable, see Figure 1(b).

Our contributions. Let a polyomino have n corners; n_c are convex. We show the following.

- If particles are initially positioned at every tile of a polyomino, a gathering sequence of length $\mathcal{O}(n_c n^2)$ can be computed in $\mathcal{O}(n_c n^2)$ time, if one exists at all (Theorem 3.2).
- There are polyominoes with shortest gathering sequences of length $\Omega(n^2)$ (Theorem 3.3).
- It is PSPACE-complete to decide if a subset of particles can be gathered (Theorem 4.3).

We utilize a previously overlooked connection between tilt models and synchronizing automata. We expect that more insights for tilt models can be gained by exploring this approach further. Conversely, the geometry of tilt models may lead to new ways to tackle open questions in automata theory. The discrepancy between the bounds on the worst-case length of gathering sequences is related to a longstanding open problem known as Černý's conjecture, see Theorem 4.2.

Due to limited space, proofs of statements marked by ($\star$) can be found in the full version [13]. Additionally, the full version contains an extensive overview of related literature and several additional results on approximating shortest gathering sequences, gathering subsets in hole-free polyominoes, and the parameterized complexity of tilt problems.

2 Preliminaries

In this section, we give the rigorous definitions omitted from Section 1 in favor of intuition. A polyomino P has an associated *dual graph* $G_P = (V, E)$, see Figure 2(b), and a *boundary* ∂P, an axis-aligned polygon separating V from $\mathbb{Z}^2 \setminus V$. We identify the vertices of V with the tiles of P and call them *pixels*. As a convention, n denotes the number of corners of ∂P, $n_c \leq n$ the number of convex corners, and $N = |V|$ the number of pixels of a given polyomino. Pixels are uniquely determined as the intersection of two *segments*, a *row segment* and a *column segment*, which are maximally contiguous parts of rows and columns; a *corner pixel* is adjacent to two perpendicular sides of ∂P, i.e., next to a convex corner, see Figure 2(a).

A polyomino is *simple* if it contains no holes. We also consider a supplementary class of polyominoes we call *mazes*: collections of horizontal and vertical segments that only intersect in T-shaped and +-shaped junctions. Formally, a maze is a *thin* polyomino (it contains no 2×2 squares) in which all corner pixels have degree 1, see Figure 2(b).

A set $C \subseteq V$ of all pixels containing particles is called a *configuration*. Pixels $p \in C$ are *occupied*. We use the set $\mathbb{D} = \{\text{U}, \text{D}, \text{L}, \text{R}\}$ as shorthand for the directions up, down, left, and right. Formally, a tilt model defines a function $\delta : \mathbb{D} \to (2^V \to 2^V)$, i.e., it maps directions to transformations of configurations. A transformation $\delta(v)$, $v \in \mathbb{D}$, is called a *move*. In

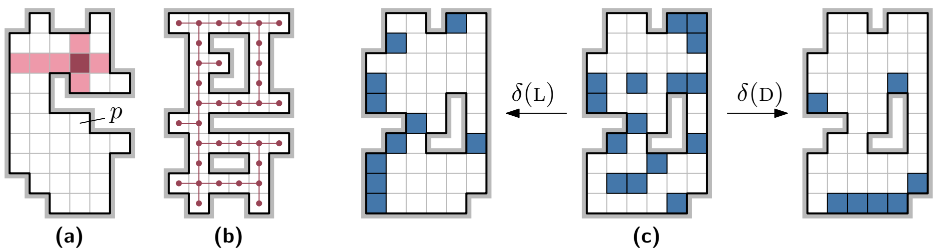

Figure 2 **(a)** A simple polyomino with a darkly shaded pixel at the intersection of two segments and a corner pixel p. **(b)** A non-simple maze and its dual graph. **(c)** Leftward and downward moves.

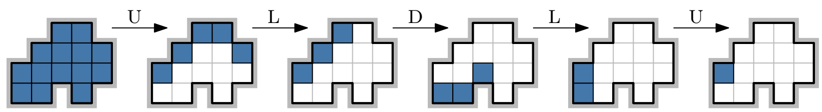

Figure 3 The sequence ULDLU is a gathering sequence for this polyomino.

this work, we consider the merging variant of the full tilt model, where particles merge upon colliding. For the direction L, $\delta(\text{L})$ is defined as follows: $p \in \delta(\text{L})(C)$ if and only if p is the leftmost pixel of the row segment R containing p and $R \cap C \neq \varnothing$. Definitions for the other directions are analogous. We extend δ to a function on $\mathbb{D}^*$, the set of sequences of directions, by setting $\delta(wv) = \delta(v) \circ \delta(w)$, for $v \in \mathbb{D}$ and $w \in \mathbb{D}^*$, and $\delta(\varepsilon)(C) = C$, where ε is the empty sequence. For simplicity, we may write $C \cdot w$ instead of $\delta(w)(C)$ and $C \xrightarrow{w} C'$ instead of $C' = \delta(w)(C)$, and use moves $\delta(v)$ and directions $v \in \mathbb{D}$ interchangeably. For singleton configurations, we define $\delta^1 : \mathbb{D}^* \to (V \to V)$ as $\delta^1(w)(p) = q$ whenever $\delta(w)(\{p\}) = \{q\}$. The uniform movement of particles allows us to reduce the behavior of arbitrary configurations to the motion of single particles.

▶ **Observation 2.1.** *For $w \in \mathbb{D}^*$ and $C \subseteq V$, $\delta(w)(C) = \bigcup_{p \in C}\{\delta^1(w)(p)\}$.*

A *gathering sequence for a configuration* C is a sequence of directions $w \in \mathbb{D}^*$ such that $|C \cdot w| = 1$, in which case C is a *gatherable configuration*. Moreover, a *gathering sequence for a polyomino* P is a gathering sequence for the configuration V, see Figure 3, making P a *gatherable polyomino*. We define $\text{sgs}(C)$ to be the length of a *shortest gathering sequence* for a gatherable configuration C, and assign $\text{sgs}(P) = \text{sgs}(V)$ for a gatherable polyomino P.

We investigate two problems in the full tilt model for a given polyomino P.

- FullGathering: Find a gathering sequence for P or decide that P is not gatherable.
- SubsetGathering: Decide if a given configuration $C \subseteq V$ is gatherable.

3 Gathering particles by synchronizing automata

The notation introduced in Section 2 may be considered unnecessarily heavy, but it highlights the connection between tilt models and automata. An *automaton* is a triple (Q, Σ, δ), where Q is a finite, non-empty set of *states*, Σ is a finite, non-empty *alphabet*, and $\delta : \Sigma \to (Q \to Q)$ is a *transition function* mapping letters of Σ to transformations of Q. Again, δ gets extended to a function over *words* $w \in \Sigma^*$. Now, a tilt model defines an automaton $(2^V, \mathbb{D}, \delta)$ for a polyomino P with dual graph $G_P = (V, E)$. This approach shows its merit when we consider the analogue to gathering sequences, synchronizing words.

A word $w \in \Sigma^*$ is a *synchronizing word* for a set $Q' \subseteq Q$ if there is a state $q \in Q$ such that $p \xrightarrow{w} q$ for all $p \in Q'$, in which case Q' is a *synchronizing set*. The minimum length

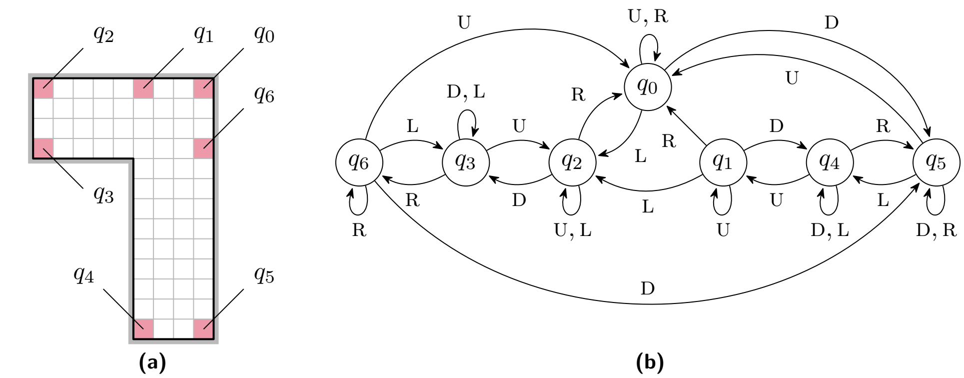

Figure 4 (a) A polyomino P and its significant pixels. **(b)** The full tilt automaton $A(P)$.

of a synchronizing word for a synchronizing set Q' is called the *reset threshold* $\mathrm{rt}(Q')$. An automaton $A = (Q, \Sigma, \delta)$ is synchronizing if Q is, in which case we define $\mathrm{rt}(A) = \mathrm{rt}(Q)$.

Finding a synchronizing word of size $\mathcal{O}(|Q|^3)$, if one exists, can be done in polynomial time, e.g., with Eppstein's algorithm [12]. Černý's conjecture, named in honor of foundational work by Černý [11], states that $\mathrm{rt}(A) \leq (|Q|-1)^2$ holds for every synchronizing automaton A. More details on synchronizing automata can be found in several excellent surveys [21, 24, 25].

Observation 2.1 tells us that a gathering sequence for a polyomino P corresponds precisely to a synchronizing word for the automaton $(V, \mathbb{D}, \delta^1)$. Thus, we can solve FullGathering by applying Eppstein's algorithm [12] to $(V, \mathbb{D}, \delta^1)$ to decide if it is synchronizing in $\mathcal{O}(N^2)$ time and, if so, find a gathering sequence for P in $\mathcal{O}(N^3)$ time. We now sketch algorithms with improved running times of $\mathcal{O}(n^2)$ and $\mathcal{O}(n_c n^2)$, respectively, that use only ∂P as input.

The key observation is that δ very quickly reduces the set of occupied pixels to one of size $\mathcal{O}(n)$. Consider the set $S \subseteq V$ of *significant pixels* that contains all corner pixels and possibly *helper pixels*, which are found by extending sides of ∂P that meet at reflex corners until they hit another side of ∂P. Figure 4(a) illustrates the significant pixels of a polyomino P with helper pixels q_1 and q_6. Observe that $|S| \leq 2n - n_c$ because at most two helper pixels are included per reflex corner. Any sequence of two moves in perpendicular directions changes the position of a particle from any pixel $p \in V$ to one in S, e.g., $\delta^1(\text{DL})(p) \in S$. S is closed under δ^1, i.e., after the move sequence DL we can focus on δ^1 restricted to S. We call the resulting automaton $A(P) = (S, \mathbb{D}, \delta^1)$ the *full tilt automaton* of P, see Figure 4(b).

▶ **Lemma 3.1.** *A polyomino P is gatherable if and only if the full tilt automaton $A(P)$ is synchronizing, in which case* $\mathrm{rt}(A(P)) \leq \mathrm{sgs}(P) \leq \mathrm{rt}(A(P)) + 1$ *holds.*

Proof. A gathering sequence for P is a synchronizing word for $A(P)$ because $S \subseteq V$. Conversely, assume w is a synchronizing word for S. If $w = \varepsilon$, then P is a single pixel and ε a gathering sequence. Otherwise, assume without loss of generality that $w = \text{L}w'$ for some $w' \in \mathbb{D}^*$ (rotate P if necessary). Then $\text{DL}w \in S$ and $\text{DL}w = \text{DLL}w'$ is a gathering sequence for P, which can be shortened to $\text{DL}w' = \text{D}w$ because moves in the full tilt model are idempotent, i.e., $\delta(v) \circ \delta(v) = \delta(v)$ for all $v \in \mathbb{D}$. ◀

It is a well-known fact [19, 25] that an automaton (Q, Σ, δ), $|Q| > 1$, is synchronizing if and only if every unordered pair of states $\{s, t\} \subseteq Q$ has a synchronizing word, suggesting a strategy of iteratively merging pairs. To efficiently determine synchronizing words for pairs in $A(P)$, we construct the *pair automaton* $A(P)^{\leq 2} = (S^{\leq 2}, \mathbb{D}, \delta^{\leq 2})$, where $S^{\leq 2} = \{\{p, q\} :$

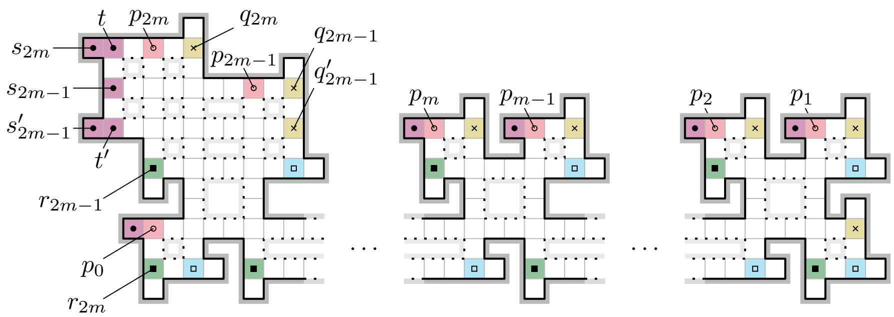

Figure 5 A family of polyominoes P_m with $\text{sgs}(P_m) \in \Omega(n^2)$. Important classes of pixels are marked with distinct colors and symbols. Without the dotted areas, the polyominoes become mazes.

$p, q \in S\}$ and $\delta^{\leq 2}(v)(\{p, q\}) = \{\delta^1(v)(p)\} \cup \{\delta^1(v)(q)\}$. $S^{\leq 2}$ also contains all singletons $\{p\} \subseteq S$. A word w with $p \cdot w = q \cdot w$ in $A(P)$ corresponds precisely to a path labeled w from $\{p, q\}$ to $\{p \cdot w\}$ in the digraph underlying $A(P)^{\leq 2}$. Using BFS on this digraph allows us to compute the next step on a shortest path to a singleton for every pair of states, if one exists.

To cut down the number of pairs that need to be considered, we initially reduce S to convex corner pixels by repeatedly applying two perpendicular moves, e.g., DL. Observe that if $p \in S$ is a helper pixel, then $p \cdot \text{DL} < p$ in terms of their lexicographic order, as at least one coordinate decreases. Thus, $\mathcal{O}(n)$ applications of DL suffice to eliminate all helper pixels (and all corner pixels not at lower-left corners) from S and $\mathcal{O}(n_c)$ pairs remain.

▶ **Theorem 3.2** (⋆). *Given the boundary ∂P of a polyomino P, one can decide in $\mathcal{O}(n^2)$ time if P is gatherable, and, if so, compute a gathering sequence of length $\mathcal{O}(n_c n^2)$ in $\mathcal{O}(n_c n^2)$ time.*

▶ **Theorem 3.3** (⋆). *There is an infinite family P_m, $m \in \mathbb{N}$, of gatherable simple polyominoes such that $\text{sgs}(P_m) \in \Omega(n^2)$, where the number of corners n grows with m.*

Proof sketch. The basic idea of the construction, depicted in Figure 5, is to place two particles on a directed cycle of length $2m + 1$, spaced apart by m positions. Moving against the direction of the cycle leads to dead ends, except at p_{2m-1}. There, LU allows to skip one step on the cycle, which can be used to reduce the distance between the particles by 1. To reduce the distance again, the whole cycle has to be traversed, resulting in $\Omega(m)$ iterations of $\Omega(m)$ moves to gather the particles, for a total of $\Omega(m^2) = \Omega(n^2)$ moves. ◀

4 Simulating binary automata with uniformly controlled particles

If Černý's conjecture is true, then the $\Omega(n^2)$ lower bound on $\text{sgs}(P)$ is actually tight.

▶ **Conjecture 4.1.** *For every gatherable polyomino P with n corners,* $\text{sgs}(P) \in \mathcal{O}(n^2)$.

The geometric structure of polyominoes should make Conjecture 4.1 easier to prove than Černý's conjecture. Alas, we must dampen this hope, because particles in polyominoes can be used to simulate binary automata, i.e., automata with alphabet $\Sigma = \{0, 1\}$.

To simulate an automaton $A = (Q, \{0, 1\}, \delta)$, the main idea is to build a polyomino $P(A)$ that has every state $q \in Q$ represented by a pixel $\overline{q}$, and place a single particle whose presence at $\overline{q}$ signifies that A is in state q. We use the clockwise cycle URDL and the counter-clockwise cycle DRUL to express transitions for 0 and 1, respectively. By carefully constructing the boundary of $P(A)$, we guarantee that all minimal sequences of moves between two pixels $\overline{q_i}$

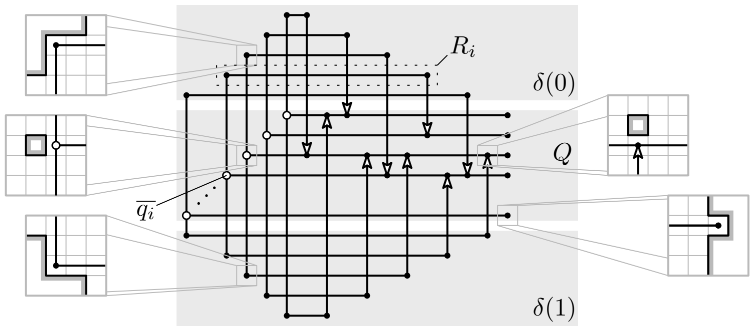

Figure 6 Schematic overview of the construction of $P(A)$ from a binary automaton A. Three blocks represent, from top to bottom, transitions for 0, states $q_i \in Q$, and transitions for 1.

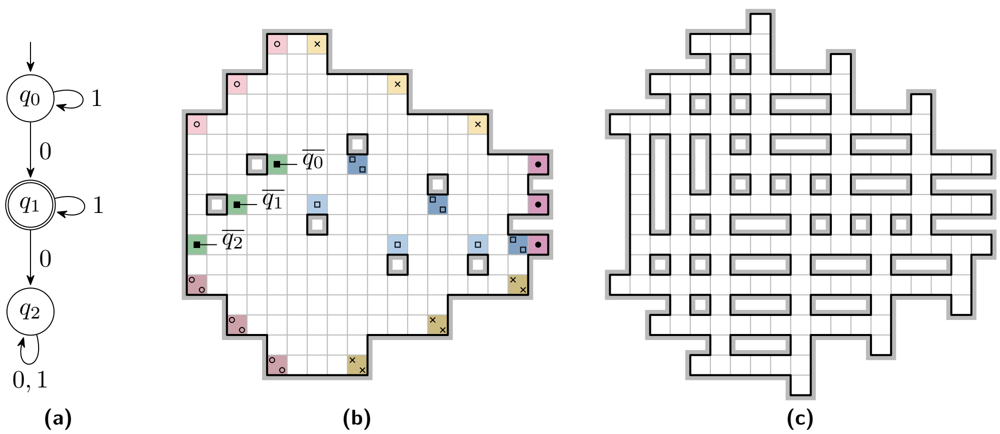

Figure 7 **(a)** The binary automaton A_0. **(b)** The polyomino $P(A_0)$. Colors and symbols indicate congruence classes of pixels reachable from representatives $\overline{q_i}$. **(c)** The maze $M(A_0)$.

and $\overline{q_j}$ can be decomposed into combinations of URDL and DRUL, thus ensuring that minimal sequences of moves between pixels representing states correspond to transitions in A.

Without loss of generality, let $Q = \{q_0, q_1, \ldots, q_{|Q|-1}\}$, and set $\overline{Q'} = \{\overline{q} : q \in Q'\}$ for sets $Q' \subseteq Q$. Our construction of $P(A)$ is illustrated in Figure 6. Every state $q_i \in Q$ gets assigned a row of width $6|Q|$ including the representative pixel $\overline{q_i}$. These rows are stacked, with left-aligned rows of width $6|Q| - 1$ in between, and the $\overline{q_i}$ are aligned on an upward diagonal, from $\overline{q_{|Q|-1}}$ in the bottom left up to $\overline{q_0}$. A 1×1 hole is placed to the left of every $\overline{q_i}$.

To implement the transitions $\delta(0)(q_i)$ in the top block, rows R_i of width $6i+3$ are stacked from top to bottom and aligned on the left with $\overline{q_i}$. A 1×1 hole is placed below the pixel at the intersection of the rightmost column of R_i and the row of $\overline{q_j}$ when $q_j = \delta(0)(q_i)$. This allows us to simulate a transition for 0 by the sequence of moves URDL. Similar rows and holes are constructed for $\delta(1)(q_i)$ in the bottom block from bottom to top, allowing us to simulate 1 by DRUL. $P(A)$ has $n = 12(2|Q| - 1)$ corners. An example is shown in Figure 7.

▶ **Theorem 4.2** ($\star$). *A set $S \subseteq Q$ of states of an automaton $A = (Q, \{0, 1\}, \delta)$ is synchronizing if and only if its set of representatives $\overline{S}$ is gatherable in $P(A)$, in which case $\mathrm{sgs}(\overline{S}) = 4\,\mathrm{rt}(S)$.*

Thus, a quadratic upper bound for $\text{sgs}(\overline{S})$ yields a quadratic upper bound for $\text{rt}(S)$. While this would not settle Černý's conjecture, it would be remarkable progress. The existence of such an upper bound is still open, suggesting that a proof for Conjecture 4.1 may be elusive.

We can also adapt the construction of $P(A)$ to get a maze $M(A)$, possibly with $\Omega(|Q|^2)$ corners, by adding more holes as well as pixels of degree 1 next to corner pixels, see Figure 7(c).

▶ **Theorem 4.3.** SubsetGathering *is* PSPACE*-complete, even in mazes.*

Proof. It is easy to see that a non-deterministic algorithm for SubsetGathering could run in polynomial space, which implies containment in NPSPACE = PSPACE [23].

We reduce from SubsetSynchronizability, which is PSPACE-hard [20] even for binary automata [27]: Given a binary automaton $A = (Q, \{0, 1\}, \delta)$ and a set $S \subseteq Q$, determine if S is synchronizing. We construct $P(A)$ for a given binary automaton $A = (Q, \{0, 1\}, \delta)$ in polynomial time. By Theorem 4.2, $S \subseteq Q$ is synchronizing if and only if $\overline{S}$ is gatherable. This property remains true when $P(A)$ is turned into a maze $M(A)$. ◀

Theorem 4.3 also follows from the PSPACE-hardness proof for the so-called Occupancy problem from [1], but we welcome an alternative proof in line with our automata theme.

References

1 Jose Balanza-Martinez, Timothy Gomez, David Caballero, Austin Luchsinger, Angel A. Cantu, Rene Reyes, Mauricio Flores, Robert T. Schweller, and Tim Wylie. Hierarchical shape construction and complexity for slidable polyominoes under uniform external forces. In *Symposium on Discrete Algorithms (SODA)*, pages 2625–2641, 2020. doi:10.1137/1.9781611975994.160.

2 Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, Golnaz Habibi, and James McLurkin. Reconfiguring massive particle swarms with limited, global control. In *Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, pages 51–66, 2013. doi:10.1007/978-3-642-45346-5_5.

3 Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, Hamed Mohtasham Shad, and Rose Morris-Wright. Tilt: The video – designing worlds to control robot swarms with only global signals. In *International Symposium on Computational Geometry (SoCG)*, pages 16–18, 2015. doi:10.4230/LIPICS.SOCG.2015.16.

4 Aaron T. Becker, Sándor P. Fekete, Li Huang, Phillip Keldenich, Linda Kleist, Dominik Krupke, Christian Rieck, and Arne Schmidt. Targeted drug delivery: Algorithmic methods for collecting a swarm of particles with uniform, external forces. In *International Conference on Robotics and Automation (ICRA)*, pages 2508–2514, 2020. doi:10.1109/ICRA40945.2020.9196551.

5 Aaron T. Becker, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, Christian Rieck, Christian Scheffer, and Arne Schmidt. Tilt assembly: Algorithms for micro-factories that build objects with uniform external forces. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 11:1–11:13, 2017. doi:10.4230/LIPIcs.ISAAC.2017.11.

6 Aaron T. Becker, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, Christian Rieck, Christian Scheffer, and Arne Schmidt. Tilt assembly: Algorithms for micro-factories that build objects with uniform external forces. *Algorithmica*, 82(2):165–187, 2020. A conference version related to this paper appeared at ISAAC 2017 [5]. doi:10.1007/S00453-018-0483-9.

7 Aaron T. Becker, Golnaz Habibi, Justin Werfel, Michael Rubenstein, and James McLurkin. Massive uniform manipulation: Controlling large populations of simple robots with a common input signal. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 520–527, 2013. doi:10.1109/IROS.2013.6696401.

8 David Caballero, Angel A. Cantu, Timothy Gomez, Austin Luchsinger, Robert T. Schweller, and Tim Wylie. Hardness of reconfiguring robot swarms with uniform external control in limited directions. *Journal of Information Processing*, 28:782–790, 2020. doi:10.2197/IPSJJIP.28.782.

9 David Caballero, Angel A. Cantu, Timothy Gomez, Austin Luchsinger, Robert T. Schweller, and Tim Wylie. Fast reconfiguration of robot swarms with uniform control signals. *Natural Computing*, 20(4):659–669, 2021. doi:10.1007/S11047-021-09864-0.

10 Ján Černý. Poznámka k homogénnym experimentom s konečnỳmi automatmi. *Matematicko-fyzikálny časopis*, 14(3):208–216, 1964. URL: https://dml.cz/dmlcz/126647.

11 Ján Černý. A note on homogeneous experiments with finite automata. *Journal of Automata, Languages and Combinatorics*, 24(2–4):123–132, 2019. The original article was published in 1964 in Slovak [10]. English translation by Markus Holzer and Bianca Truthe. doi:10.25596/jalc-2019-123.

12 David Eppstein. Reset sequences for monotonic automata. *SIAM Journal on Computing*, 19(3):500–510, 1990. doi:10.1137/0219033.

13 Sándor P. Fekete, Jonas Friemel, Peter Kramer, Jan-Marc Reinhardt, Christian Rieck, and Christian Scheffer. Tilt automata: Gathering particles with uniform external control. *CoRR*, 2026. arXiv:2603.02796.

14 Phillip Keldenich, Sheryl Manzoor, Li Huang, Dominik Krupke, Arne Schmidt, Sándor P. Fekete, and Aaron T. Becker. On designing 2D discrete workspaces to sort or classify polyominoes. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018. doi:10.1109/IROS.2018.8594150.

15 Matthias Konitzny, Yitong Lu, Julien Leclerc, Sándor P. Fekete, and Aaron T. Becker. Gathering physical particles with a global magnetic field using reinforcement learning. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 10126–10132, 2022. doi:10.1109/IROS47612.2022.9982256.

16 Fabian C. Landers, Lukas Hertle, Vitaly Pustovalov, Derick Sivakumaran, Cagatay M. Oral, Oliver Brinkmann, Kirstin Meiners, Pascal Theiler, Valentin Gantenbein, Andrea Veciana, Michael Mattmann, Silas Riss, Simone Gervasoni, Christophe Chautems, Hao Ye, Semih Sevim, Andreas D. Flouris, Josep Puigmartí-Luis, Tiago Sotto Mayor, Pedro Alves, Tessa Lühmann, Xiangzhong Chen, Nicole Ochsenbein, Ueli Moehrlen, Tilman Schubert, Zsolt Kulcsar, Philipp Gruber, Miriam Weisskopf, Quentin Boehler, Salvador Pané, and Bradley J. Nelson. Clinically ready magnetic microrobots for targeted therapies. *Science*, 390(6774):710–715, 2025. doi:10.1126/science.adx1708.

17 Arun Mahadev, Dominik Krupke, Sándor P. Fekete, and Aaron T. Becker. Mapping and coverage with a particle swarm controlled by uniform inputs. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1097–1104, 2017. doi:10.1109/IROS.2017.8202280.

18 Arun V. Mahadev, Dominik Krupke, Jan-Marc Reinhardt, Sándor P. Fekete, and Aaron T. Becker. Collecting a swarm in a grid environment using shared, global inputs. In *International Conference on Automation Science and Engineering (CASE)*, pages 1231–1236, 2016. doi:10.1109/COASE.2016.7743547.

19 Balas K. Natarajan. An algorithmic approach to the automated design of parts orienters. In *Symposium on Foundations of Computer Science*, pages 132–142, 1986. doi:10.1109/SFCS.1986.5.

20 Igor K. Rystsov. Polynomial complete problems in automata theory. *Information Processing Letters*, 16(3):147–151, 1983. doi:10.1016/0020-0190(83)90067-4.

21 Sven Sandberg. Homing and synchronizing sequences. In *Model-Based Testing of Reactive Systems*, pages 5–33, 2004. doi:10.1007/11498490_2.

22 Walter J. Savitch. Deterministic simulation of non-deterministic turing machines (detailed abstract). In *Symposium on Theory of Computing (STOC)*, pages 247–248, 1969. doi:10.1145/800169.805439.

23 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970. A conference version of this paper appeared at STOC 1969 [22]. doi:10.1016/S0022-0000(70)80006-X.

24 Mikhail V. Volkov. Synchronizing automata and the Černý conjecture. In *Language and Automata Theory and Applications (LATA)*, pages 11–27, 2008. doi:10.1007/978-3-540-88282-4_4.

25 Mikhail V. Volkov. Synchronization of finite automata. *Russian Mathematical Surveys*, 77(5):819–891, 2022. doi:10.4213/rm10005e.

26 Vojtěch Vorel. Subset synchronization of transitive automata. In *Automata and Formal Languages (AFL)*, pages 370–381, 2014. doi:10.4204/EPTCS.151.26.

27 Vojtěch Vorel. Subset synchronization and careful synchronization of binary finite automata. *International Journal of Foundations of Computer Science*, 27(5):557–578, 2016. A conference version of this paper appeared at AFL 2014 [26]. doi:10.1142/S0129054116500167.