Moving Matter: Efficient Reconfiguration of Tile Arrangements by a Single Active Robot*

Aaron T. Becker[1], Sándor P. Fekete[2], Jonas Friemel[3], Ramin Kosfeld[2], Peter Kramer[2], Harm Kube[4], Christian Rieck[5], Christian Scheffer[3], and Arne Schmidt[2]

1 Electrical Engineering, University of Houston, Texas, USA
atbecker@uh.edu

2 Computer Science, TU Braunschweig, Braunschweig, Germany
s.fekete@tu-bs.de, {kosfeld, kramer, aschmidt}@ibr.cs.tu-bs.de

3 Electrical Engineering and Computer Science, Bochum University of Applied Sciences, Bochum, Germany
{jonas.friemel, christian.scheffer}@hs-bochum.de

4 Computer Science, TU Berlin, Berlin, Germany
h.kube@campus.tu-berlin.de

5 Discrete Mathematics, University of Kassel, Kassel, Germany
christian.rieck@mathematik.uni-kassel.de

Abstract

We consider the problem of reconfiguring a two-dimensional connected grid arrangement of passive building blocks from a start configuration to a goal configuration, using a single active robot that can move on the tiles, remove individual tiles from a given location and physically move them to a new position by walking on the remaining configuration. The objective is to determine a reconfiguration schedule that minimizes the overall makespan, while ensuring that the tile configuration remains connected. We provide the following: (1) We present a generalized version of the problem, parameterized by weighted costs for moving with or without tiles, and show that this is NP-hard. (2) We give a polynomial-time constant-factor approximation for the case of disjoint start and target bounding boxes. Our algorithm yields optimal carry distance for 2-scaled instances.

Related Version arXiv:2502.09299

1 Introduction

Building and modifying structures consisting of many basic components is an important objective, both in fundamental theory and in a spectrum of practical settings. Transforming such structures with the help of autonomous robots is particularly relevant in very large [9] and very small dimensions [25] that are hard to access for direct human manipulation, e.g., in extraterrestrial space [7] or in microscopic environments [5].

Progress in material sciences has spawned discrete, light-weight materials that allow building large lattice structures of tiles that can be manipulated by simple robots for reconfiguration [21], as shown in Figure 1: The robot can move on the tile arrangement, remove individual tiles and physically relocate them to a new position by walking on the remaining configuration, which needs to remain connected at all times.

* Work from the University of Houston was partially supported by NSF grant IIS-2130793. Work from TU Braunschweig and HS Bochum was partially supported by the German Research Foundation (DFG), project “Space Ants”, FE 407/22-1 and SCHE 1931/4-1. Work from the University of Kassel was partially supported by DFG grant 522790373.

41st European Workshop on Computational Geometry, Liblice, Czech republic, April 9–11, 2025.
This is an extended abstract of a presentation given at EuroCG'25. 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.

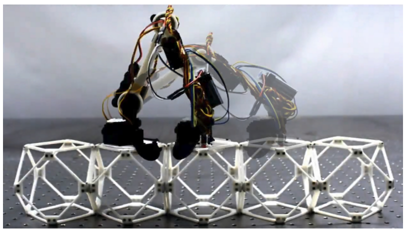

Figure 1 A simple Bill-E robot that can move on a configuration of digital, light-weight material and relocate individual voxels for overall reconfiguration. Photos adapted from [21].

How can we use such a robot to transform a given start configuration into a desired goal arrangement, as quickly as possible? We provide the following results.

1. We present a generalized version of the problem, parameterized by weighted costs for moving with or without tiles, and show that this is NP-hard.
2. We give a polynomial-time constant-factor approximation in case of disjoint start and target bounding boxes. Our approach yields optimal carry distance for 2-scaled instances.

Due to limited space, details for statements marked by ($\star$) can be found in the full version [6]. However, we provide a concise overview of the ideas.

Related work. Garcia et al. [15, 16] showed that computing optimal schedules for robotic reconfiguration is NP-hard. They designed heuristic approaches exploiting rapidly exploring random trees (RRT), and a time-dependent variant of the A* search algorithm.

A different context for reconfiguration arises from programmable matter [17, 18, 19]. Here, even finite automata are capable of building bounding boxes from tiles around polyominoes, as well as scale and rotate them while maintaining connectivity at all times [12, 24].

For arrangements that are composed of active, self-moving objects (or agents), a number of related results have been obtained. For the *sliding cube model* [13, 14], Akitaya et al. [3] show that universal sequential reconfiguration in two dimensions is possible, even while maintaining connectivity of all intermediate configurations, but minimizing the makespan of a schedule is NP-complete. Abel et al. [1] and Kostitsyna et al. [22] gave similar results in three dimensions. Most recently, Akitaya et al. [4] give results for the *parallel sliding square model*. In a related model, Fekete et al. [10, 11] show that parallel connected reconfiguration of a swarm of (labeled) agents is NP-complete, even for deciding whether there is a schedule of makespan 2, and present algorithms for computing constant stretch schedules, i.e., the ratio between the makespan of a schedule and a natural lower bound (the maximum minimum distance between an individual start and target position) is bounded by a constant.

Preliminaries. For the following, we refer to Figure 2. We are given a fixed set of n indistinguishable square *tile* modules located at discrete, unique *positions* in the infinite integer grid $\mathbb{Z}^2$. If this set induces a connected subgraph, where two positions are considered connected if either their x- or y-coordinate differs by 1, we say that the tiles form a connected *configuration* or *polyomino*. Let $\mathcal{C}(n)$ refer to the set of all polyominoes of n tiles.

Consider a *robot* that occupies a single tile at any given time and uses cardinal directions to navigate; the unit vectors $(1, 0)$ and $(0, 1)$ correspond to *east* and *north*, respectively. In discrete time steps, the robot can either move to an adjacent tile, pick up an adjacent tile (if it is not carrying one), or place a tile at an adjacent unoccupied position (if it is carrying a tile). A tile may only be picked up if the configuration remains connected without it.

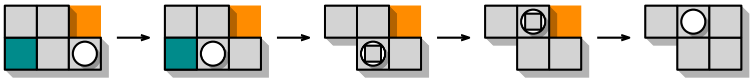

Figure 2 An example schedule for some $C_s \rightrightarrows C_t$: The robot moves in cardinal directions, walking on and modifying tiles. Tiles in $C_s \cap C_t$ are marked in gray, $C_s \setminus C_t$ in cyan, and $C_t \setminus C_s$ in orange.

Given two connected configurations C_s and C_t, a *(reconfiguration) schedule* S is a finite, connectivity-preserving sequence of operations to be performed by the robot for $C_s \rightrightarrows C_t$ exactly if it transforms C_s into C_t. Let $d_C(S)$ denote the *carry distance*, which is the sum of geodesic distances between consecutive pickups and drop-offs in S. This represents the total distance the robot travels while carrying a tile, with an additional unit of distance added each time the robot either picks up or places a tile. Similarly, the *empty distance* $d_E(S)$ is the geodesic distance walked without carrying a tile.

In this paper, we consider the Single Robot Reconfiguration problem: Given two connected configurations C_s and C_t, and a *rational weight factor* $\lambda \in [0, 1]$, our goal is to compute a schedule S for $C_s \rightrightarrows C_t$ of minimum *makespan* $|S| := \lambda \cdot d_E(S) + d_C(S)$. The associated decision problem is defined as expected with an upper bound on the length of a schedule. We refer to the minimum weighted makespan for a given instance as OPT.

2 Computational complexity of the problem

We start by investigating the computational complexity of the decision variant of the generalized reconfiguration problem. In particular, we prove that the problem is NP-hard for any rational factor $\lambda \in [0, 1]$. This generalizes a result by Garcia et al. [16] for $\lambda = 1$.

▶ **Theorem 2.1** ($\star$). *Single Robot Reconfiguration is* NP*-hard, parameterized by* λ.

We distinguish between two cases. If $\lambda \in (0, 1]$, we reduce from the Hamiltonian path problem in induced subgraphs of the infinite grid graph [20]. The high-level idea is to expand the grid graph of a given Hamiltonian path instance, placing small reconfiguration tasks at each vertex of the graph.

For $\lambda = 0$, we reduce from Planar Monotone 3Sat [8]. We utilize and adapt the reduction given by Akitaya et al. [3] for the sequential sliding squares problem. As $\lambda = 0$, the robot is effectively allowed to "teleport" across the configuration. Therefore, we construct variable and clause gadgets in a way that a unit square must be carried through one side of a variable in order to perform a constant number of reconfiguration steps within the clauses.

3 Constant-factor approximation

We now turn to a special case of the optimization variant in which the configurations have *disjoint bounding boxes*, i.e., there exists an axis-parallel bisector that separates them. Let this bisector be horizontal such that the target configuration lies south. We present an algorithm that computes schedules of makespan at most $c \cdot \mathsf{OPT}$ for some fixed $c \geq 1$.

3.1 2-scaled instances

We additionally impose the constraint that both the start and target configurations are 2-*scaled*, i.e., they consist of 2×2-squares of tiles aligned with a 2×2 integer grid, and show:

▶ **Theorem 3.1** (⋆). *There exists a constant c such that for any pair of 2-scaled configurations $C_s, C_t \in \mathcal{C}(n)$ with disjoint bounding boxes, we can efficiently compute a schedule for $C_s \rightrightarrows C_t$ with weighted makespan at most $c \cdot OPT$.*

Our algorithm utilizes a type of intermediate configuration called *histogram*. A histogram consists of a *base* strip of unit height and (multiple) orthogonal unit width *columns* attached to its base. The direction of its columns determines the orientation of a histogram, e.g., the histogram H_s in Figure 3 is *north-facing*. We proceed in three phases, see Figure 3.

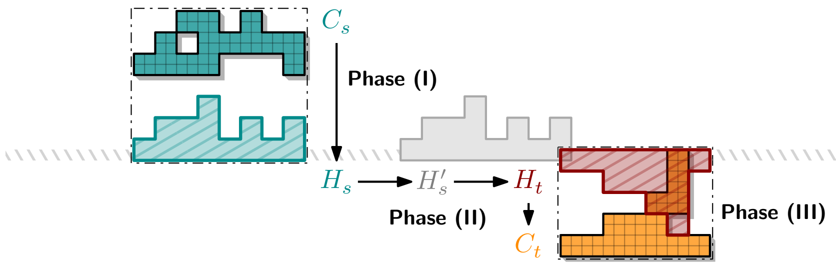

Figure 3 An example for a start and target configuration C_s and C_t, the intermediate histograms H_s and H_t sharing a baseline, and the horizontally translated H_s' that shares a tile with H_t.

Phase (I). Transform the configuration C_s into a north-facing histogram H_s.
Phase (II). Translate H_s to overlap with the target bounding box and transform it into a south-facing histogram H_t contained in the bounding box of C_t.
Phase (III). Finally, apply **Phase (I)** in reverse to obtain C_t from H_t.

Since **Phases (I)** and **(III)** are largely identical, we reduce to two subroutines: Transforming a 2-scaled configuration into a 2-scaled histogram and converting two such histograms into one another. We denote the optimal carry distance for any schedule $C_s \rightrightarrows C_t$ by $\sigma(C_s, C_t)$.

Phase (I): Transform a configuration into a histogram. We proceed by describing a subroutine that constructs a histogram from an arbitrary 2-scaled configuration by moving tiles strictly in one cardinal direction. The resulting histogram faces the opposite direction.

▶ **Lemma 3.2** (⋆). *Let C_s be a 2-scaled polyomino and let H_s be a histogram that can be created from C_s by moving tiles in only one cardinal direction. We can efficiently compute a schedule of makespan $\mathcal{O}(n + \sigma(C_s, H_s))$ for $C_s \rightrightarrows H_s$ with optimal carry distance.*

Our strategy is simply as follows: We iteratively move sets of tiles by two units into the respective target direction, until the histogram is constructed. We give a high-level explanation of our approach by example of a north-facing histogram, as depicted in Figure 4.

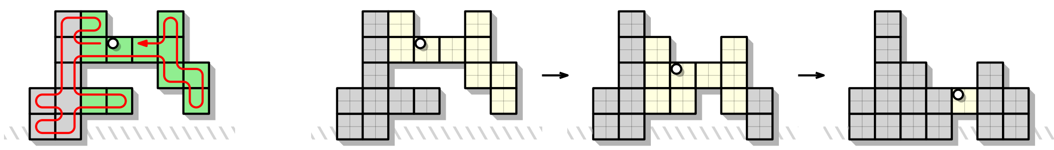

Figure 4 Left: A walk across all tiles (red), the set H (gray) and two free components (green). Right: Based on the walk, the free components are iteratively moved south to reach a histogram shape. The free component that is going to be translated south next is highlighted in yellow.

Let P be any intermediate 2-scaled polyomino obtained by moving tiles south while realizing $C_s \rightrightarrows H_s$. Let H be the set of maximal vertical strips of tiles that contain a base tile in H_s, i.e., all tiles that do not need to be moved further south. We define the *free components* of P as the set of connected components in $P \setminus H$. By definition, once a tile becomes part of H, it is not moved again until the target histogram H_s is obtained.

The robot now simply walks across the entire polyomino, and whenever it enters a free component, it performs a subroutine that translates the component south by two units by repeatedly moving the northernmost tile in a column south. As the configuration is 2-scaled, this guarantees connectivity; we refer to Figure 4 for an illustration.

▶ **Lemma 3.3** (⋆). *Given a free component F of a 2-scaled polyomino P, we can efficiently compute a schedule of makespan $\mathcal{O}(|F|)$ to translate F in the target direction by two units.*

By applying Lemma 3.3 on the whole polyomino P instead of just a free component, we can translate P in any direction with asymptotically optimal makespan.

▶ **Corollary 3.4.** *Any 2-scaled polyomino can be translated by k units in any cardinal direction by a schedule of weighted makespan $\mathcal{O}(n \cdot k)$.*

Phase (II): Reconfigure a histogram into a histogram. By the assumption of the existence of a horizontal bisector between the bounding boxes of C_s and C_t, the histogram H_s is north-facing, whereas H_t is south-facing. The bounding box of C_s is vertically extended to share exactly one y-coordinate with the bounding box of C_t, and this is where both histogram bases are placed; see Figure 3. By Corollary 3.4, the tiles in H_s can be moved toward H_t in asymptotically optimal makespan until the histogram bases share a tile.

▶ **Lemma 3.5** (⋆). *Let H_s be a north-facing and H_t a south-facing histogram that share at least one base tile. We can efficiently compute a schedule of makespan $\mathcal{O}(n + \sigma(H_s, H_t))$ for $H_s \rightrightarrows H_t$ with optimal carry distance.*

Figure 5 illustrates our approach: We iteratively move the northernmost westernmost tile of H_s to the northernmost westernmost unoccupied position in H_t until H_t is constructed. That position may not be reachable initially, in which case we first extend the histogram base in western direction. This ordering ensures that tiles are moved on shortest paths to H_t.

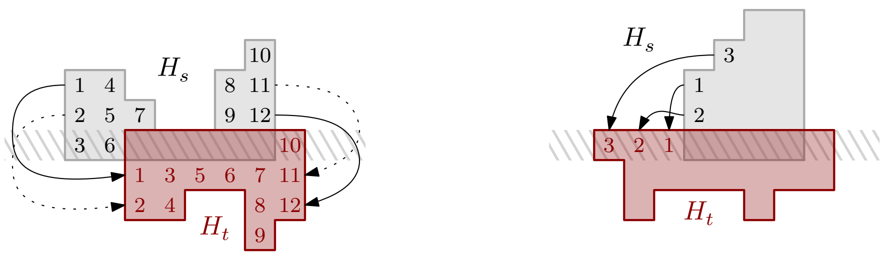

Figure 5 Left: Ordering of tile moves for $H_s \rightrightarrows H_t$. Right: If the westernmost unoccupied position in H_t is unreachable, the base may need to be extended first.

By Lemmas 3.2 and 3.5, tiles are moved with optimal carry distance in **Phases (I)**, **(II)**, and **(III)**. We can show that the combined paths remain shortest possible, yielding an optimal schedule for $\lambda = 0$, i.e., if there is no movement cost when the robot is not carrying a tile.

▶ **Corollary 3.6.** *For any pair of 2-scaled configurations $C_s, C_t \in \mathcal{C}(n)$ with disjoint bounding boxes and $\lambda = 0$, we can efficiently compute an optimal schedule for $C_s \rightrightarrows C_t$.*

3.2 General instances

The key advantage of 2-scaled instances is the absence of cut vertices, which simplifies the maintenance of connectivity during the reconfiguration, which we now tackle separately.

Most parts of our previous method already work independent of the configuration scale. The only required modification concerns the translation of free components, as the polyomino may become disconnected while moving free components that are not 2-scaled. The key technique here is that two auxiliary tiles can be used to preserve connectivity at cut vertices that need to be moved; an example is illustrated in Figure 6. The use of auxiliary tiles to preserve connectivity is also exploited in other models [2, 23]. For this, we can use any two non-cut vertex tiles from the starting configuration, e.g., any leaf from a spanning tree of C_s.

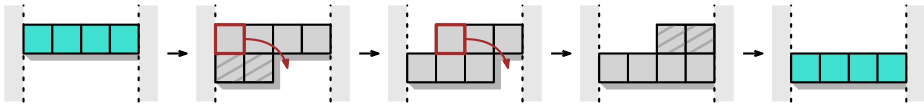

Figure 6 Translating a *corridor* (in cyan) of width 4 south, using two auxiliary (hatched) tiles.

The high level idea of the adjustment is as follows: We decompose each free component F into its *elements*; (1) maximal vertical *strips* of unit width and (2) maximal horizontal *corridors* of unit height. As translating a single element may cause disconnection to adjacent elements, we apply a recursive strategy that handles the elements *blocking* our translation, i.e., that would yield a disconnected configuration, first. We then move the current element, and finally process all other adjacent elements. Finally, we obtain the following.

▶ **Theorem 3.7** (⋆). *There exists a constant c such that for any pair of configurations $C_s, C_t \in \mathcal{C}(n)$ with disjoint bounding boxes, we can efficiently compute a schedule for $C_s \rightrightarrows C_t$ with weighted makespan at most $c \cdot \mathsf{OPT}$.*

4 Conclusions and future work

Our paper presents progress on the reconfiguration problem for tile-based structures with a single active robot. In particular, we showed that the problem is NP-hard for any weighted cost function based on walking and carrying. Complementarily, we developed an constant-factor approximation algorithm to reconfigure two polyominoes into one another in the case that both configurations are contained in disjoint bounding boxes.

Several open questions remain: It seems plausible that our methods can be generalized to be performed by many robots in parallel. Much more intricate is the question on whether a fully distributed approach is possible. Finally, can we adapt our approach to instances in which the bounding boxes of the configurations are intersecting, i.e., overlapping or nested?

References

1 Zachary Abel, Hugo A. Akitaya, Scott Duke Kominers, Matias Korman, and Frederick Stock. A universal in-place reconfiguration algorithm for sliding cube-shaped robots in a quadratic number of moves. In *Symposium on Computational Geometry (SoCG)*, pages 1:1–1:14, 2024. doi:10.4230/LIPICS.SOCG.2024.1.

2 Hugo A. Akitaya, Esther M. Arkin, Mirela Damian, Erik D. Demaine, Vida Dujmovic, Robin Y. Flatland, Matias Korman, Belén Palop, Irene Parada, André van Renssen, and Vera Sacristán. Universal reconfiguration of facet-connected modular robots by pivots: The $O(1)$ musketeers. *Algorithmica*, 83:1316–1351, 2021. doi:10.1007/s00453-020-00784-6.

3 Hugo A. Akitaya, Erik D. Demaine, Matias Korman, Irina Kostitsyna, Irene Parada, Willem Sonke, Bettina Speckmann, Ryuhei Uehara, and Jules Wulms. Compacting squares: Input-sensitive in-place reconfiguration of sliding squares. In *Scandinavian Symposium on Algorithm Theory (SWAT)*, pages 4:1–4:19, 2022. doi:10.4230/LIPICS.SWAT.2022.4.

4 Hugo A. Akitaya, Sándor P. Fekete, Peter Kramer, Saba Molaei, Christian Rieck, Frederick Stock, and Tobias Wallner. Sliding squares in parallel, 2024. doi:10.48550/arXiv.2412.05523.

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. *Algorithmica*, 82(2):165–187, 2020. doi:10.1007/S00453-018-0483-9.

6 Aaron T. Becker, Sándor P. Fekete, Jonas Friemel, Ramin Kosfeld, Peter Kramer, Harm Kube, Christian Rieck, Christian Scheffer, and Arne Schmidt. Moving matter: Efficient reconfiguration of tile arrangements by a single active robot, 2025. doi:10.48550/arXiv.2502.09299.

7 Mohamed Khalil Ben-Larbi, Kattia Flores Pozo, Tom Haylok, Mirue Choi, Benjamin Grzesik, Andreas Haas, Dominik Krupke, Harald Konstanski, Volker Schaus, Sándor P. Fekete, Christian Schurig, and Enrico Stoll. Towards the automated operations of large distributed satellite systems. Part 1: Review and paradigm shifts. *Advances in Space Research*, 67(11):3598–3619, 2021. doi:10.1016/j.asr.2020.08.009.

8 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal on Computational Geometry and Applications*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.

9 Sándor P. Fekete. Coordinating swarms of objects at extreme dimensions. In *International Workshop on Combinatorial Algorithms (IWOCA)*, pages 3–13, 2020. doi:10.1007/978-3-030-48966-3_1.

10 Sándor P. Fekete, Phillip Keldenich, Ramin Kosfeld, Christian Rieck, and Christian Scheffer. Connected coordinated motion planning with bounded stretch. *Autonomous Agents and Multi-Agent Systems*, 37(2), 2023. doi:10.1007/S10458-023-09626-5.

11 Sándor P. Fekete, Peter Kramer, Christian Rieck, Christian Scheffer, and Arne Schmidt. Efficiently reconfiguring a connected swarm of labeled robots. *Autonomous Agents and Multi-Agent Systems*, 38(2), 2024. doi:10.1007/s10458-024-09668-3.

12 Sándor P. Fekete, Eike Niehs, Christian Scheffer, and Arne Schmidt. Connected reconfiguration of lattice-based cellular structures by finite-memory robots. *Algorithmica*, 84(10):2954–2986, 2022. doi:10.1007/s00453-022-00995-z.

13 Robert Fitch, Zack J. Butler, and Daniela Rus. Reconfiguration planning for heterogeneous self-reconfiguring robots. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 2460–2467, 2003. doi:10.1109/IROS.2003.1249239.

14 Robert Fitch, Zack J. Butler, and Daniela Rus. Reconfiguration planning among obstacles for heterogeneous self-reconfiguring robots. In *International Conference on Robotics and Automation (ICRA)*, pages 117–124, 2005. doi:10.1109/ROBOT.2005.1570106.

15 Javier Garcia, Michael Yannuzzi, Peter Kramer, Christian Rieck, and Aaron T. Becker. Connected reconfiguration of polyominoes amid obstacles using RRT*. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 6554–6560, 2022. doi:10.1109/IROS47612.2022.9981184.

16 Javier Garcia, Michael Yannuzzi, Peter Kramer, Christian Rieck, Sándor P. Fekete, and Aaron T. Becker. Reconfiguration of a 2D structure using spatio-temporal planning and load transferring. In *International Conference on Robotics and Automation (ICRA)*, pages 8735–8741, 2024. doi:10.1109/ICRA57147.2024.10611057.

17 Robert Gmyr, Kristian Hinnenthal, Irina Kostitsyna, Fabian Kuhn, Dorian Rudolph, and Christian Scheideler. Shape recognition by a finite automaton robot. In *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 52:1–52:15, 2018. doi:10.4230/LIPIcs.MFCS.2018.52.

18 Robert Gmyr, Kristian Hinnenthal, Irina Kostitsyna, Fabian Kuhn, Dorian Rudolph, Christian Scheideler, and Thim Strothmann. Forming tile shapes with simple robots. *Natural Computing*, 19(2):375–390, 2020. doi:10.1007/s11047-019-09774-2.

19 Kristian Hinnenthal, David Liedtke, and Christian Scheideler. Efficient shape formation by 3D hybrid programmable matter: An algorithm for low diameter intermediate structures. In *Symposium on Algorithmic Foundations of Dynamic Networks (SAND)*, pages 15:1–15:20, 2024. doi:10.4230/LIPICS.SAND.2024.15.

20 Alon Itai, Christos H. Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982. doi:10.1137/0211056.

21 Benjamin Jenett, Amira Abdel-Rahman, Kenneth Cheung, and Neil Gershenfeld. Material–robot system for assembly of discrete cellular structures. *IEEE Robotics and Automation Letters*, 4(4):4019–4026, 2019. doi:10.1109/LRA.2019.2930486.

22 Irina Kostitsyna, Tim Ophelders, Irene Parada, Tom Peters, Willem Sonke, and Bettina Speckmann. Optimal in-place compaction of sliding cubes. In *Scandinavian Symposium on Algorithm Theory (SWAT)*, pages 31:1–31:14, 2024. doi:10.4230/LIPICS.SWAT.2024.31.

23 Othon Michail, George Skretas, and Paul G. Spirakis. On the transformation capability of feasible mechanisms for programmable matter. *Journal of Computer and System Sciences*, 102:18–39, 2019. doi:10.1016/j.jcss.2018.12.001.

24 Eike Niehs, Arne Schmidt, Christian Scheffer, Daniel E. Biediger, Michael Yannuzzi, Benjamin Jenett, Amira Abdel-Rahman, Kenneth C. Cheung, Aaron T. Becker, and Sándor P. Fekete. Recognition and reconfiguration of lattice-based cellular structures by simple robots. In *International Conference on Robotics and Automation (ICRA)*, pages 8252–8259, 2020. doi:10.1109/ICRA40945.2020.9196700.

25 Yangsheng Zhu, Mingjing Qi, Zhiwei Liu, Jianmei Huang, Dawei Huang, Xiaojun Yan, and Liwei Lin. A 5-mm untethered crawling robot via self-excited electrostatic vibration. *IEEE Transactions on Robotics*, 38(2):719–730, 2022. doi:10.1109/TRO.2021.3088053.