Parallel Self-Assembly under Uniform Control Inputs

Sheryl Manzoor, Sam Sheckman, Hoyeon Kim, Jarrett Lonsford, Minjun Kim, and Aaron T. Becker

***Abstract*— We present fundamental progress on parallel self-assembly using large swarms of micro-scale particles in complex environments, controlled not by individual navigation, but by a uniform global, external force with the same effect on each particle. Consider a 2D grid world, in which all obstacles and particles are unit squares, and for each actuation, robots move maximally until they collide with an obstacle or another robot. We present algorithms that, given an arbitrary 2D structure, designs an obstacle layout. When actuated, this layout generates copies of the input 2D structure. We analyze the spatial and time complexity of the factory layouts. We present hardware results on both a macro-scale, gravity-based system and a milli-scale, magnetically actuated system.**

I. Introduction

One of the exciting new directions of robotics is the design and development of micro- and nanorobot systems, with the goal of letting a massive swarm of robots perform complex operations in a complicated environment. Due to scaling issues, individual control of the involved robots becomes physically impossible: while energy storage capacity drops with the third power of robot size, medium resistance decreases much slower. As a consequence, current micro- and nanorobot systems with many robots are steered and directed by an external force that acts as a common control signal [1]–[7]. These common control signals include global magnetic or electric fields, chemical gradients, and turning a light source on and off.

A. Selective Control with Global Inputs

Clearly, having only one global signal that uniformly affects all robots at once poses a strong restriction on the ability of the swarm to perform complex operations. This control symmetry can be broken using interactions between the robot swarm and obstacles in the environment. This paper builds on the techniques for controlling many simple particles with uniform control inputs presented in [8]–[10], where we demonstrated how such a system could implement digital computation.

B. Model

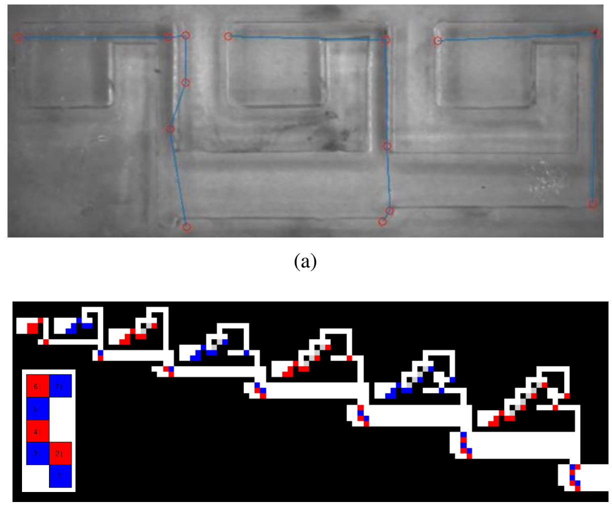
Assume the following rules:

1) A planar grid *workspace* W is filled with a number of unit-square robots (each occupying one cell of the grid) and some fixed unit-square blocks. Each unit square in the workspace is either *free*, which a particle may occupy or *obstacle* which a robot may not occupy. Each square in the grid can be referenced by its Cartesian coordinates $\boldsymbol{x} = (x, y)$.
2) All particles are commanded in unison: the valid commands are “Go Up” (u), “Go Right” (r), “Go Down” (d), or “Go Left” (l).
3) Particles all move until they
   a) hit an obstacle
   b) hit a stationary particle.
   c) share an edge with a compatible particle

   If a particle shares an edge with a compatible robot the two robots bond and from then on move as a unit. A *move sequence* $\boldsymbol{m}$ consists of an ordered sequence of moves m_k, where each $m_k \in \{u, d, r, l\}$ A representative move sequence is $\langle u, r, d, l, d, r, u, \ldots \rangle$. We assume the area of W is finite and issue each command long enough for the robots to reach their maximum extent.

*This work was supported by the National Science Foundation under Grant No. [IIS-1553063] and [IIS-1619278].

S. Manzoor, J Lonsford, and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA {smanzoor2, jllonsford, atbecker}@uh.edu S. Sheckman, H. Kim and M. Kim are with Lyle School of Engineering, Southern Methodist University, Dallas, TX 75205 USA {ssheckman, hoyeonk, mjkim}@smu.edu



(a) (b)

Fig. 1. (a) A milli-scale magnetic based prototype. (b) A seven tile factory. Each particle is actuated simultaneously by the same global field. The factory is designed so each clockwise control input assembles another component.

II. Related Work

A. Microscale Biomanufacturing

Naturally derived biomaterials as building blocks for functional materials and devices are increasingly desired because they are environmentally and biologically safer than purely synthetic materials. One such class of materials, polysaccharide based hydrogels, are intriguing because they can reversibly encapsulate a variety of smaller components. Many groups have termed these loaded-alginate particles as artificial cells, in that they mimic the basic structure of living cells (membrane, cytoplasm, organelles, etc.) [11]–[13]. Construction with these micron-sized gels has numerous applications in industry, including cell manipulation, tissue engineering, and micro-particle assembly [14]–[18], but requires fundamental research in biology, medicine, and colloidal science. While there are several methods to efficiently fabricate these particulate systems, it is still challenging to construct larger composite materials out of these units [19]. Traditional methods of assembling larger macroscale systems are unemployable due to the change of dominant forces at small length scales. In particular, forces due to electromagnetic interactions dominate gravitational forces at the microscale resulting in strong adhesion and sudden shifts in the position of microparts under atmospheric conditions. To form constructs out of microgels, groups have traditionally turned to non-robotic microfluidic systems that utilize a variety of actuation methods, including mechanical, optical, dielectrophoretic, acoustophoretic, and thermophoretic [20]–[24]. While each of these methods has proven to be capable of manipulating biological cells, each method has significant drawbacks that limit their widespread application. For example, microscale mechanical, acoustophoretic, and thermophoretic manipulation methods use stimuli that can be potentially lethal to live cells [25]. Furthermore, most, if not all, of these techniques require expensive equipment and lack control schemes necessary to precisely manipulate large numbers of cells autonomously.

B. Control Microrobotic Swarms Using Only Global Signals

Today one of the most exciting new frontiers in robotics is the development of micro- and nanorobotic systems, which hold the potential to revolutionize the fields of manufacturing and medicine. Chemist, biologist, and roboticist have shown the ability to produce very large populations (10^3–10^{14}) of small scale (10^{-9}–10^{-6} m) robots using a diverse array of materials and techniques [26]–[28]. Untethered swarms of these tiny robots may be ideal for on-site construction of high-resolution macroscale materials and devices. While these new types of large-population, small-sized, robotic systems have many advantages over their larger-scale counterparts, they also present a set of unique challenges in terms of their control. Due to current limitations in fabrication, micro- and nanorobots have little-to-no onboard computation, along with limited computation and communication ability [28]–[30]. These limitations make controlling swarms of these robots individually impractical. Thus, these robotic systems are often controlled by a uniform global external signal (e.g. chemical gradients, electric and magnetic fields), which makes motion planning for large robotic populations in tortuous environments difficult. We recently demonstrated that obstacles present in the workspace can break the symmetry of approximately identical robotic swarms, enabling positional configuration of robots [31]. Given a large-enough free space, a single obstacle is sufficient for positional control over N particles. This method can be used to form complex assemblies out of large swarms of mobile microrobotic building blocks, using only a single global input signal.

C. Microrobot Based Microassembly

The ability to create microrobots, and control algorithms capable of autonomous manipulation and assembly of small scale components into functional materials is currently a major manufacturing challenge [11]. To address this challenge, teams of microrobotic systems must work together intelligently to coordinate manipulation tasks in novel environments. While several microrobots capable of performing simple manipulation and assembly tasks have been reported [12]–[17], few have shown the ability to pattern intricate designs or assemble complex multi-component parts. Recently, some groups have begun to develop cell-safe magnetically actuated microrobotic systems for cell patterning, yet their method is limited in that these systems are manually controlled, not automated, and suffer from low spatial resolution [32], [33]. In this paper, we seek to combine the use of microscale hybrid organic/inorganic actuators along with novel swarm control algorithms for mask free programmable patterning and micro-assembly. Specifically, this paper applies swarm control and particle logic computations to magnetically actuate artificial cells, so as to use them as micro-scale robotic swarms, to create complex, high resolution, 2D and 3D patterns and assemblies.

III. Theory

This section explains how to design factories that build arbitrary 2D shaped polyominos. We first assign species to individual tiles of the polyomino, second discover a build path, and finally build an assembly line of factory components that each add one tile to partially assembled polyomino and pass the polyomino to the next component.

A. Arbitrary 2D shapes require two particle species

A *polyomino* is a 2D geometric figure formed by joining one or more equal squares edge to edge. Polyominoes have four-point connectivity.

Lemma 1: Any polyomino can be constructed using just two species

Proof: Label a grid with an alternating pattern like a checkerboard. Any desired polyomino can be constructed on this checkerboard, and all joints are between dissimilar species. An example shape is shown in Fig. 2. ■

The sufficiency of two species to construct any shape gives many options for implementation. The two species could correspond to any gendered connection, including electric charge, ionic charge, magnetic polarity, or hook-and-loop type fasteners.

B. Complexity Handled in This Paper

Different 2D part geometries are more difficult to construct than others. Fig. 3 shows four parts of varying complexity. Label the first particle in the assembly process the seed

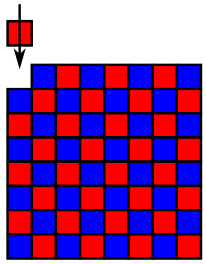


Fig. 2. Any polyomino can be constructed with two compatible robot species.

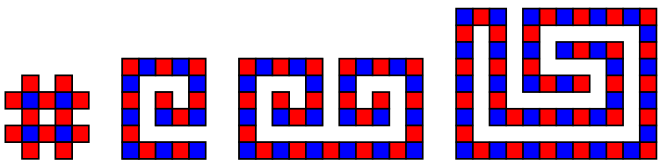


Fig. 3. Polyomino parts. Difficulty increases from left to right. The right parts cannot be built by additive construction.

particle. The part on the left is shaped as a '#' symbol. Though it has an interior hole, any of the 16 particles could serve as the seed particle, and the shape could be constructed around it. The second shape is a spiral, and must be constructed from the inside-out. If the outer spiral was completed first, there would be no path to add particles to finish the interior because added particles would have to slide past compatible particles. Increasing the number of species would not solve this problem, because there is a narrow passage through the spiral that forces incoming parts to slide past the edges of all the bonded particles.

The third shape is the combination of a left-handed and a right-handed spiral. This part cannot be assembled by adding one particle at a time, because each spiral must be constructed from the inside-out. Instead, this part must be divided into sub-assemblies that are each constructed, and then combined. The fourth shape contains compound overhangs, and may be impossible to construct with additive manufacturing. The algorithms in this paper detect if the desired shape can be constructed one particle at a time. If so, a build order is provided, and a factory layout is designed.

```
talk about loops. If no loop, n^2 time,
if loops, n^2 * (number of loops)!
```

```
simplify fig 3, ad fig 4 for loops
```

C. Discovering a Build Path

Given a polyomino, Alg. 1 determines if the polyomino can be built by adding one component at a time. The forward problem of determining a build order is difficult because there are $O(n!)$ possible build orders, and many of them may be violate the constraints give in Section . Each new tile must have a straight-line path to its goal position in the polyomino that does not collide with any other particle, does not slide past an opposite species of tile, and terminates in a mating configuration with an opposite species tile. As in many robotics problem, the inverse problem is easier. The inverse problem of deconstruction maintains an array of the remaining tiles in the polyomino $\mathbf{R}$. In the inner for loop at line 8, a temporary array $\mathbf{T}$ is generated that contains all but the jth tile in $\mathbf{R}$. This loop simply checks (1) if the jth tile can be removed along a straight-line path without colliding with any other particle or sliding past an opposite species of tile in line 9, and (2) that its removal does not fragment the remaining polyomino into more than one piece in line 10. This algorithm requires at most $1/2n(1+n)$ iterations, because there are n particles to remove, and each iteration considers one less particle than the previous iteration.

Algorithm 1 FINDBUILDPATH($\mathbf{P}$)

$\mathbf{P}$ is the x, y coordinates of a 4-connected polyomino that has at least a 1-tile empty border. Returns $\mathbf{C}$, $\mathbf{c}$ and $\mathbf{m}$ where $\mathbf{C}$ contains sequence of polyomino coordinates, $\mathbf{c}$ is a vector of color labels and $\mathbf{m}$ is a vector of directions for assembly.

1: $\mathbf{c} \leftarrow$LABELCOLOR($\mathbf{P}$)
2: $\{\mathbf{C}, \mathbf{m}\} =$DECOMPOSE($\mathbf{P}, \mathbf{c}$)
3: **return** $\{\mathbf{C}, \mathbf{c}, \mathbf{m}\}$

Algorithm 2 DECOMPOSE($\mathbf{P}, \mathbf{c}$)

$\mathbf{P}$ is the x, y coordinates of a 4-connected polyomino and $\mathbf{c}$ is a vector of color labels. Returns $\mathbf{C}$, and $\mathbf{m}$ where $\mathbf{C}$ contains sequence of polyomino coordinates and $\mathbf{m}$ is a vector of directions for assembly.

1: $\{\mathbf{R}, \mathbf{C}, \mathbf{m}, \ell\} \leftarrow$ERODE($\mathbf{P}, \mathbf{c}$)
2: **if** $|\mathbf{R}| = 0$ **or** $\neg\ell$ **then**
3:     **return** $\{\mathbf{C}, \mathbf{m}\}$
4: $\mathbf{d} \leftarrow \{u, d, l, r\}, \mathbf{R} \leftarrow \mathbf{P}$
5: **for** $j \leftarrow 1, j \leq |\mathbf{R}|$ **do**
6:     $\mathbf{p} \leftarrow \mathbf{R}_j, \mathbf{T} \leftarrow \mathbf{R} \backslash \mathbf{R}_j$
7:     **for** $k \leftarrow 1, k \leq 4$ **do**
8:         **if** (CHECKPATHTILE($\mathbf{T}, \mathbf{p}, \mathbf{d}_k, \mathbf{c}$) **and**
9:             $1 = |$4-CONNCOMP($\mathbf{T}$)$|$) **then**
10:             $\{\mathbf{C2}, \mathbf{m2}\} =$DECOMPOSE($\mathbf{T}, \mathbf{c}$)
11:             **if** $\mathbf{C2} \neq \{\}$ **then**
12:                 $\mathbf{C}_{1+|\mathbf{R}|} \leftarrow \mathbf{p}, \mathbf{m}_{|\mathbf{R}|} \leftarrow \mathbf{d}_k$
13:                 $\mathbf{C}_{1:|\mathbf{C2}|} \leftarrow \mathbf{C2}, \mathbf{m}_{1:|\mathbf{m2}|} \leftarrow \mathbf{m2}$
14:                 **return** $\{\mathbf{C}, \mathbf{m}\}$
15:             **break**
16: $\mathbf{C} \leftarrow \{\}, \mathbf{m} \leftarrow \{\}$
17: **return** $\{\mathbf{C}, \mathbf{m}\}$

D. Assembling Tiles

1) Hopper Construction: Two-part adhesives react when the components mix. Placing the components in separate containers prevents mixing. Similarly, storing many particles

Algorithm 3 ERODE($\mathbf{P}, \mathbf{c}$)

$\mathbf{P}$ is the x, y coordinates of a 4-connected polyomino and $\mathbf{c}$ is a vector of color labels. Returns $\mathbf{R}$, $\mathbf{C}$, $\mathbf{m}$, and ℓ where $\mathbf{R}$ is a list of coordinates of the remaining polyomino, $\mathbf{C}$ contains sequence of tile coordinates that were removed, $\mathbf{m}$ is a vector of directions for assembly, and ℓ if loops were detected.

$\mathbf{C} \leftarrow \{\}, \mathbf{m} \leftarrow \{\}, \ell \leftarrow$ FALSE
$\mathbf{d} \leftarrow \{u, d, l, r\}, \mathbf{R} \leftarrow \mathbf{P}$
$w \leftarrow |$8-CONNCOMP$|(\neg\mathbf{R})$
for $i \leftarrow 1, i < |\mathbf{P}|$ **do**
  successRemove $\leftarrow$ FALSE
  for $j \leftarrow 1, j \leq |\mathbf{R}|$ **do**
    $\mathbf{p} \leftarrow \mathbf{R}_j, \mathbf{T} \leftarrow \mathbf{R} \backslash \mathbf{R}_j$
    for $k \leftarrow 1, k \leq 4$ **do**
      if CHECKPATHTILE($\mathbf{T}, \mathbf{p}, \mathbf{d}_k, \mathbf{c}$) **and**
        $1 = |$4-CONNCOMP$(\mathbf{T})|$ **then**
        if $w = |$8-CONNCOMP$(\neg\mathbf{T})|$ **then**
          $\mathbf{R} \leftarrow \mathbf{T}$, *successRemove* $\leftarrow$ TRUE
          $\mathbf{C}_{1+|\mathbf{R}|} \leftarrow \mathbf{p}, \mathbf{m}_{|\mathbf{R}|} \leftarrow \mathbf{d}_k$
        else $\ell \leftarrow$ TRUE
        break
  if *successRemove* $=$ FALSE **then**
    $\mathbf{C} \leftarrow \{\}, \mathbf{m} \leftarrow \{\}$
    break
if $|\mathbf{R}| = 1$ **then**
  $\mathbf{C}_1 \leftarrow \mathbf{R}_1$
return $\{\mathbf{R}, \mathbf{C}, \mathbf{m}, \ell\}$

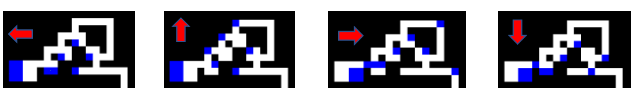


Fig. 4. Hopper with delays. The hopper is filled with similarly-labelled robots that will not combine. Every clockwise command sequence $\langle u, r, d, l \rangle$ releases one robot from the hopper.

of a single species in separate containers allows controlled mixing.

We can design *part hoppers*, containers that store similarly labelled particles. These particles will not bond with each other. The hopper shown in Fig. 4 releases one particle every cycle.

discuss delays

E. Part Assembly Jigs

Assembly is an iterative procedure. A factory layout is generated by BUILDFACTORY($\mathbf{P}, n_c$), described in Alg. 4. This function takes a 2D polyomino $\mathbf{P}$ and, if $\mathbf{P}$ has a valid build path, designs an obstacle layout to generate n_c copies of the polyomino. A polyomino is composed of $|\mathbf{P}| = n$ tiles.

For each tile, the function FACTORYADDTILE($n, \mathbf{b}, m, C, c, w$) described in Alg. 5 is called to generates an obstacle configuration $\mathbf{A}$. $\mathbf{A}$ forms a hopper that releases a particle each iteration and a chamber that temporarily holds the partially-assembled polyomino $\mathbf{b}$ and guides the new particle C to the correct mating position. A 24-tile factory is shown in Fig. 5.

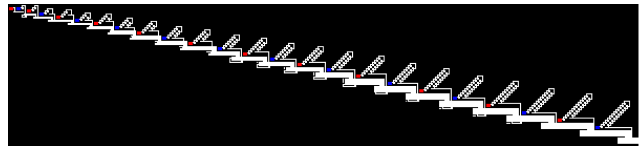


Fig. 5. A twenty-four tile factory (full resolution – zoom in for details).

Algorithm 4 BUILDFACTORY($\mathbf{P}, n_c$)

$\mathbf{P}$ is the x, y coordinates of a 4-connected polyomino. n_c is the number of parts desired. Returns a two dimensional array $\mathbf{F}$ containing the factory obstacles and filled hoppers.

$\mathbf{F} \leftarrow \{\}$ ▷ the factory obstacle array
$\mathbf{b} \leftarrow \{\}$ ▷ the part being built
$\{\mathbf{C}, \mathbf{c}, \mathbf{m}\} \leftarrow$ FINDBUILDPATH($\mathbf{P}$)
if $\{\} = \mathbf{m}$ **then**
  return F
$\{\mathbf{A}, \mathbf{b}\} \leftarrow$FACTORYFIRSTTILE($n_c, \mathbf{c}_i, w$)
for $i \leftarrow 2, i \leq |\mathbf{c}|$) **do**
  $\{\mathbf{A}, \mathbf{b}\} \leftarrow$FACTORYADDTILE($n_c, \mathbf{b}, \mathbf{m}_{i-1}, \mathbf{C}_i, \mathbf{c}_i, w$)
  $\mathbf{F} \leftarrow$CONCATFACTORIES($\mathbf{F}, \mathbf{A}$)
return F

Algorithm 5 FACTORYADDTILE($n_c, \mathbf{b}, m, C, c, w$)

$\{$**hopper**$\} \leftarrow$HOPPER(c, n_c, w)
if $m = d$ **and** $(C_y \leq \max \mathbf{b}_y$ **or** $C_y > \min \mathbf{b}_x)$ **then**
  $\{\mathbf{A}, \mathbf{b}\} \leftarrow$DOWNDIR(**hopper**, $\mathbf{b}, \mathbf{C}$)
else if $m = l$ **and** $(C_x \leq \max \mathbf{b}_x$ **or** $C_y > \min \mathbf{b}_y)$ **then**
  $\{\mathbf{A}, \mathbf{b}\} \leftarrow$LEFTDIR(**hopper**, $\mathbf{b}, \mathbf{C}$)
else if $m = u$ **and** $(C_y \leq \min \mathbf{b}_y$ **or** $C_x > \min \mathbf{b}_x)$ **then**
  $\{\mathbf{A}, \mathbf{b}\} \leftarrow$UPDIR(**hopper**, $\mathbf{b}, \mathbf{C}$)
else if $m = r$ **and** $(C_y \leq \min \mathbf{b}_x$ **or** $C_y > \min \mathbf{b}_y)$ **then**
  $\{\mathbf{A}, \mathbf{b}\} \leftarrow$RIGHTDIR(**hopper**, $\mathbf{b}, \mathbf{C}$)
return $\{\mathbf{A}, \mathbf{b}\}$

IV. ANALYSIS

This section analyzes the time and space required for a factory and gives simulation results.

A. Running Time

Running a factory simulation has three phases, ramp up, production, and wind down. During the $n - 1$ *ramp up* cycles, the first polyomino is being constructed one tile at a time and no polyominos are produced. Clever design of delays in the part hoppers ensures no unconnected tiles

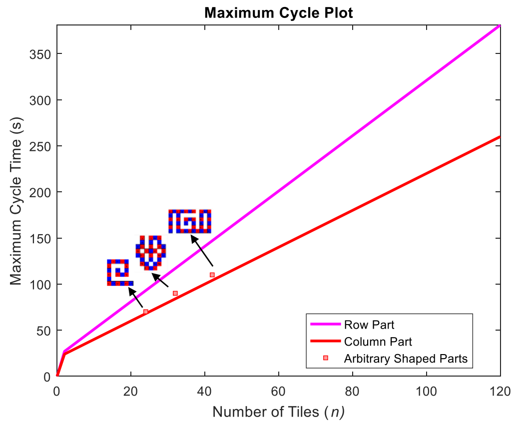


Fig. 6. Cycle time plotted against number of tiles n. The cycle time is the sum time to move in the r, d, l, u moves each cycle. Cycle time increases linearly and is upper bounded by row parts and lower bounded by column parts. Total construction time for a particle is $n\cdot$ cycle time.

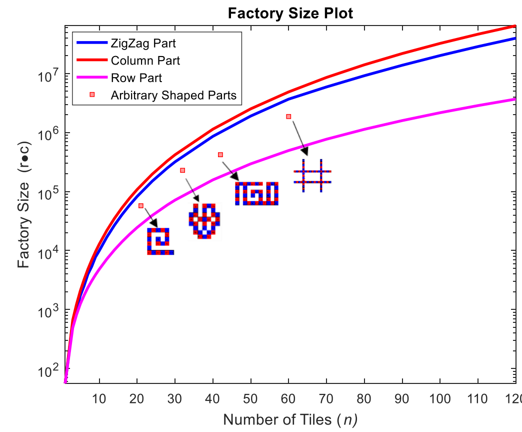


Fig. 7. Factory size grows quadratically with the number of tiles, and is upper bounded by column-shaped polynominos and lower bounded by row-shaped polyominos.

are released. During *production* cycles, one polyomino is finished each cycle. Once the first part hopper empties, the $m-1$ *wind down* cycles each produce a complete polynomino as successive hoppers also empty. This section analyzes running time, defined as the time required for each commanded move until all tiles are stopped. We assume all tiles move unit distance in unit time. There are two results, the *construction time*, the time required to assemble a single polynomino from scratch, and the *cycle time*, the time required during production cycles to advance all partial assemblies one cycle. Since a polyomino contains n tiles, the *construction time* during production cycles is $n\cdot$ *cycle time*.

Cycle time is the sum of the maximum distances moved in each direction. As shown in Fig. 6, polyominos shaped as a $n\times 1$ row require the longest time of $4n+16$. Polyominos shaped as a $1\times n$ column require the least time of $2n+16$. Construction time therefore requires $O(n^2)$ time.

B. Space Required

The space required by a factory is a function of the size of individual sub-assemblies.

$$height(n) = \begin{cases} \lceil n_c/w \rceil + 2((\lceil n/2 \rceil + 1) + (\mathbf{b}_y + 1)), \\ \text{for } m = l \text{ or } d, n \geq 2 \\ \lceil n_c/w \rceil + 2((\lceil n/2 \rceil + 1) + (\mathbf{b}_y + 2)), \\ \text{for } m = u \text{ or } r, n \geq 2 \end{cases}$$

$$hopper_{width} = w + (2\lceil n/2 \rceil + 8), n \geq 2 \qquad (1)$$

$$width(n) = \begin{cases} hopper_{width} + (\mathbf{b}_x - column_{loc}), \\ \text{for } m = d \text{ and } \mathbf{b}_x < hopper_{width} \\ \mathbf{b}_x + 1, \\ \text{for } m = d \text{ and } \mathbf{b}_x \geq hopper_{width} \\ hopper_{width}, \\ \text{for } m \neq d \text{ and } \mathbf{b}_x < hopper_{width} \\ \mathbf{b}_x + 3, \\ \text{for } m \neq d \text{ and } \mathbf{b}_x \geq hopper_{width} \end{cases}$$

First sub-assembly is constructed separately and it does not have any delay. Beginning from the second sub-assembly, height can be computed as a function of number of copies n_c of the polyomino, width of the hopper w, position of the sub-assembly n and the rows of the sub-assembled polyomino $\mathbf{b}_y$. Similarly, width of the sub-assembly can also be calculated. If the length of partially assembled polyomino is greater than the length of hopper and delays then width depends on $\mathbf{b}_x$. For tile added to a sub-assembled polyomino on down move, width also depends on the location of the column $column_{loc}$ to which tile is added.

Because a factory requires $O(n)$ rows and $O(n)$ columns, the total requires space is $O(n^2)$. As shown in Fig. 7, the required size is upper bounded by column-shaped polynominos and lower bounded by row-shaped polyominos, and is $O(n^2)$.

C. Simulation Results

Algorithms 1 through 5 were coded in MATLAB and are available at [34].

V. Experiment

To demonstrate the algorithm, we developed a magnetic control stage and alginate micro-particles.

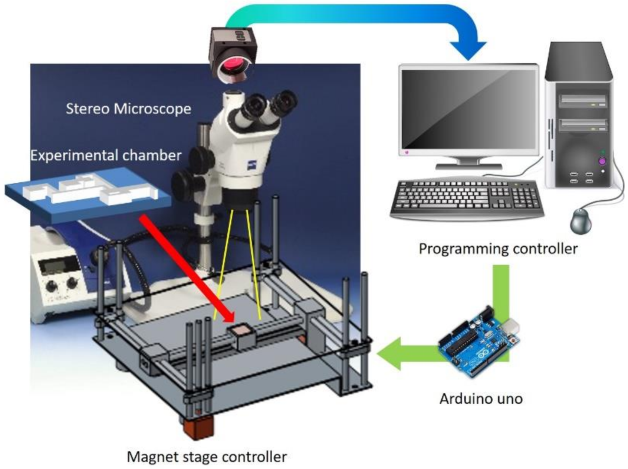


Fig. 8. Experimental platform.

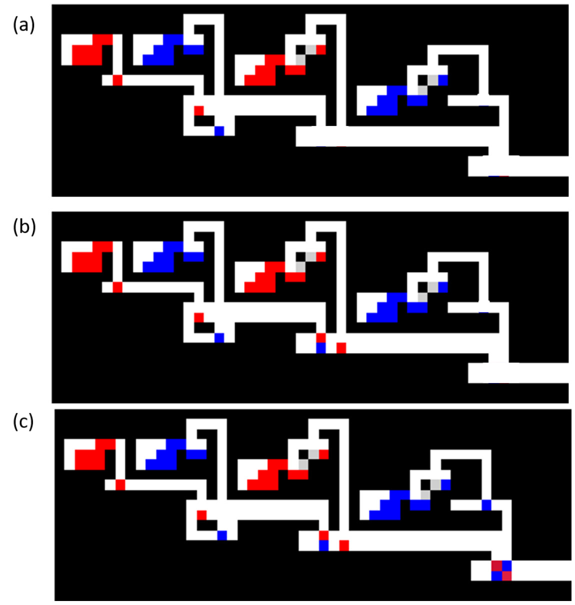


Fig. 9. Fig. Construction of a microrobotic polyomino from four alginate artificial cells. (a) Initial position of alginate microrobots at all chambers, (b) First assemble by two microrobots from two chambers, (c) Final result of construction.

a) Experimental setup: This stage generates a magnetic drag force by moving a permanent magnet. The permanent magnet is able to move x, y direction as following two mail shafts. The permanent magnet has T and the dimension is cm^2. The main channel is made up PDMS and it was filled with motility buffer. The alginate microrobot was fabricated using . After the alginate microrobots were located at each chamber in the channel, the experimental channel was located on the center of the stage where a magnet was positioned initially. The stage controller was manipulated by a C++ programming through an Arduino UNO. The channel was observed by a stereo microscope and the installed camera captured all sequent images (fps). The scheme of system is shown in Fig. 8.

b) Experimental result: Using one of construction maps, it is available to demonstrate the map using multiple alginate microrobots. The initial scene is shown in Fig. 9a and the first assemble was manipulated moving the magnet in a clockwise direction as indicated in Fig. 9b. The alginate microrobots moved in the oriented direction until coming into contact with an object. The final completion of a square polyomino is shown in the lower right corner in Fig. 9c. In addition, other polyominoes were simultaneously being manufactured.

VI. Conclusion

This work introduces a new model for additive assembly that enables efficient parallel construction because it does not depend on individual control of each agent. Instead, the workspace is designed to direct particles. This enables a simple global control input to produce a complex output.

Interesting applications will aim at microfluidics work.

Future work could extend Algorithms 1–5 to three dimensions. To build a polyomino, our current algorithm requires time that grows linearly with the number of tiles in a polyomino. Parts can be decomposed into subassemblies, which would enable more complex parts to be created and enable construction in logarithmic time.

References

[1] B. R. Donald, C. G. Levey, I. Paprotny, and D. Rus, “Planning and control for microassembly of structures composed of stress-engineered MEMS microrobots,” *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 218–246, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/2/218.abstract

[2] P.-T. Chiang, J. Mielke, J. Godoy, J. M. Guerrero, L. B. Alemany, C. J. Villagómez, A. Saywell, L. Grill, and J. M. Tour, “Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules,” *ACS Nano*, vol. 6, no. 1, pp. 592–597, Feb. 2011.

[3] H.-W. Tung, D. R. Frutiger, S. Panè, and B. J. Nelson, “Polymer-based wireless resonant magnetic microrobots,” in *IEEE International Conference on Robotics and Automation*, May 2012, pp. 715–720.

[4] E. Diller, J. Giltinan, and M. Sitti, “Independent control of multiple magnetic microrobots in three dimensions,” *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 614–631, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/5/614.abstract

[5] W. Jing, N. Pagano, and D. Cappelleri, “A tumbling magnetic microrobot with flexible operating modes,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 5514–5519.

[6] Y. Ou, D. H. Kim, P. Kim, M. J. Kim, and A. A. Julius, “Motion control of magnetized tetrahymena pyriformis cells by magnetic field with model predictive control,” *Int. J. Rob. Res.*, vol. 32, no. 1, pp. 129–139, Jan. 2013.

[7] D. de Lanauze, O. Felfoul, J.-P. Turcot, M. Mohammadi, and S. Martel, “Three-dimensional remote aggregation and steering of magnetotactic bacteria microrobots for drug delivery applications,” *The International Journal of Robotics Research*, 11 2013. [Online]. Available: http://ijr.sagepub.com/content/early/2013/11/11/0278364913500543

[8] A. Becker, E. Demaine, S. Fekete, G. Habibi, and J. McLurkin, “Reconfiguring massive particle swarms with limited, global control,” in *International Symposium on Algorithms and Experiments for Sensor*

Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS), Sophia Antipolis, France, Sep. 2013, pp. 51–66.

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

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

[11] T. M. S. Chang, “Therapeutic applications of polymeric artificial cells,” *Nature Reviews Drug Discovery*, vol. 4, no. 3, pp. 221–235, 2005.

[12] S. Prakash, *Artificial cells, cell engineering and therapy*. Elsevier, 2007.

[13] T. M. S. Chang, *Artificial cells: biotechnology, nanomedicine, regenerative medicine, blood substitutes, bioencapsulation, and cell/stem cell therapy*. World Scientific, 2007, vol. 1.

[14] D. B. Weibel, W. R. DiLuzio, and G. M. Whitesides, “Microfabrication meets microbiology,” *Nature Reviews Microbiology*, vol. 5, no. 3, pp. 209–218, 2007.

[15] J. J. Abbott, Z. Nagy, F. Beyeler, and B. Nelson, “Robotics in the small,” *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.

[16] C. Yi, C.-W. Li, S. Ji, and M. Yang, “Microfluidics technology for manipulation and analysis of biological cells,” *Analytica Chimica Acta*, vol. 560, no. 1, pp. 1–23, 2006.

[17] J. Castillo, M. Dimaki, and W. E. Svendsen, “Manipulation of biological samples using micro and nano techniques,” *Integrative Biology*, vol. 1, no. 1, pp. 30–42, 2009.

[18] M. Sitti, H. Ceylan, W. Hu, J. Giltinan, M. Turan, S. Yim, and E. Diller, “Biomedical applications of untethered mobile milli/microrobots,” *Proceedings of the IEEE*, vol. 103, no. 2, pp. 205–224, Feb 2015.

[19] R. E. Assal, P. Chen, and U. Demirci, “Highlights from the latest articles in advanced biomanufacturing at micro-and nano-scale,” *Nanomedicine*, vol. 10, no. 3, pp. 347–350, 2015.

[20] J. P. Desai, A. Pillarisetti, and A. D. Brooks, “Engineering approaches to biomanipulation,” *Annu. Rev. Biomed. Eng.*, vol. 9, pp. 35–53, 2007.

[21] P. Y. Chiou, A. T. Ohta, and M. C. Wu, “Massively parallel manipulation of single cells and microparticles using optical images,” *Nature*, vol. 436, no. 7049, pp. 370–372, 2005.

[22] C. W. Shields IV, C. D. Reyes, and G. P. López, “Microfluidic cell sorting: a review of the advances in the separation of cells from debulking to rare cell isolation,” *Lab on a Chip*, vol. 15, no. 5, pp. 1230–1249, 2015.

[23] P. Augustsson, J. Persson, S. Ekström, M. Ohlin, and T. Laurell, “Decomplexing biofluids using microchip based acoustophoresis,” *Lab on a Chip*, vol. 9, no. 6, pp. 810–818, 2009.

[24] D. Vigolo, R. Rusconi, H. A. Stone, and R. Piazza, “Thermophoresis: microfluidics characterization and separation,” *Soft Matter*, vol. 6, no. 15, pp. 3489–3493, 2010.

[25] Y.-H. Lin, Y.-W. Yang, Y.-D. Chen, S.-S. Wang, Y.-H. Chang, and M.-H. Wu, “The application of an optically switched dielectrophoretic (odep) force for the manipulation and assembly of cell-encapsulating alginate microbeads in a microfluidic perfusion cell culture system for bottom-up tissue engineering,” *Lab on a Chip*, vol. 12, no. 6, pp. 1164–1173, 2012.

[26] M. Rubenstein, C. Ahler, and R. Nagpal, “Kilobot: A low cost scalable robot system for collective behaviors,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3293–3298.

[27] Y. Ou, D. H. Kim, P. Kim, M. J. Kim, and A. A. Julius, “Motion control of magnetized tetrahymena pyriformis cells by a magnetic field with model predictive control,” *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 129–140, 2013.

[28] P.-T. Chiang, J. Mielke, J. Godoy, J. M. Guerrero, L. B. Alemany, C. J. Villagomez, A. Saywell, L. Grill, and J. M. Tour, “Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules,” *ACS nano*, vol. 6, no. 1, pp. 592–597, 2011.

[29] 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.

[30] B. R. Donald, C. G. Levey, I. Paprotny, and D. Rus, “Planning and control for microassembly of structures composed of stress-engineered mems microrobots,” *The International journal of robotics research*, vol. 32, no. 2, pp. 218–246, 2013.

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

[32] S. Tasoglu, E. Diller, S. Guven, M. Sitti, and U. Demirci, “Untethered micro-robotic coding of three-dimensional material composition,” *Nature communications*, vol. 5, 2014.

[33] S. Tasoglu, C. Yu, H. Gungordu, S. Guven, T. Vural, and U. Demirci, “Guided and magnetic self-assembly of tunable magnetoceptive gels,” *Nature communications*, vol. 5, 2014.

[34] S. Manzoor and A. T. Becker, “Particle Assembly, https://github.com/aabecker/particlecomputation/tree/master/assembly,” Feb. 2017.