Parallel Self-Assembly of Polyominoes under Uniform Control Inputs

Sheryl Manzoor[1], Samuel Sheckman[2], Jarrett Lonsford[1], Hoyeon Kim[2], Min Jun Kim[2], and Aaron T. Becker[1]

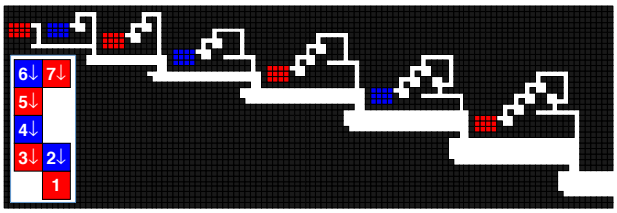
***Abstract*—We present fundamental progress on parallel self-assembly using large swarms of micro-scale particles in complex environments, controlled not by individual navigation, but by a uniform, global, external force with the same effect on each particle. Consider a 2D grid world, in which all obstacles 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, design an obstacle layout. When actuated, this layout generates copies of the input 2D structure. We analyze the movement and spatial complexity of the factory layouts. We present hardware results on both a macro-scale, gravity-based system and a micro-scale, magnetically-actuated system.**

***Index Terms*—Automation at Micro-Nano Scales; Additive Manufacturing; Underactuated Robots**

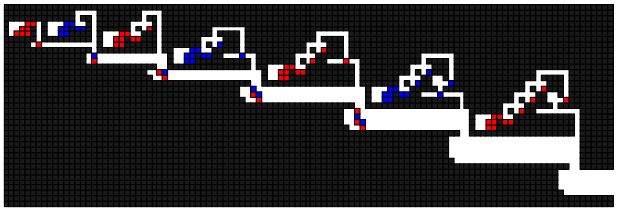
I. Introduction

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 length, 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.

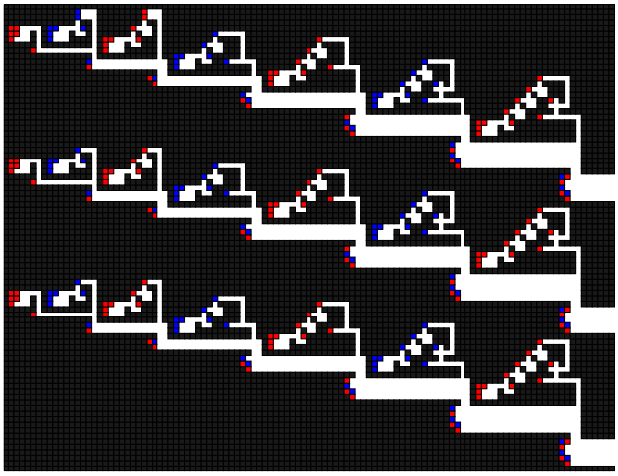
Having only one global signal that uniformly affects all robots at once limits the swarm's ability to perform complex operations. Independent control is possible by designing heterogeneous particles that respond differently to the global input, but this approach requires precise differences in each particle and is best suited for small populations. Alternatively, 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. Fig. 1 illustrates the main contribution of this paper: algorithms to produce a factory that uses global inputs to assemble arbitrary polyominoes. A *polyomino* is a 2D geometric figure formed by joining one or more equal squares edge to edge.



(a) Seven-tile polyomino factory, 0 commanded moves, 0 unit steps.



(b) Same factory, 18 commanded moves, 136 unit steps.



(c) Parallel assembly with three factories, 28 commanded moves, 221 unit steps, three complete polyominoes.

Fig. 1. Factory schematics for assembling the seven-tile polyomino in (a). Numbers and arrows on the polyomino show the build order and direction for build. All tiles are actuated simultaneously by the same global field. Red and blue tiles represent two different species that join when brought near each other. Each factory is designed so at full production every clockwise cycle of control input moves completes another polyomino. See video attachment for animation.

Manuscript received: Feb., 15, 2017; Revised May, 3, 2017; Accepted May, 26, 2017.

This paper was recommended for publication by Editor Yu Sun upon evaluation of the Associate Editor and Reviewers' comments. *This work was supported by the National Science Foundation under Grant No. [IIS-1553063], [IIS-1619278], and [IIS-1712088].

[1]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

[2]S. Sheckman, H. Kim and M.J. Kim are with Lyle School of Engineering, Southern Methodist University, Dallas, TX 75205 USA {ssheckman, hoyeonk, mjkim}@smu.edu

Digital Object Identifier (DOI): see top of this page.

This paper combines microscale hybrid organic/inorganic particles 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, to use them as micro-scale robotic swarms that create complex, high resolution, 2D patterns and assemblies.

a) Microscale Biomanufacturing: Naturally derived biomaterials as building blocks for functional materials and devices are increasingly desired because they are often 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 *artificial cells*, because 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 macro-scale 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 micro-scale 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 Swarms Using Only Global Signals: Micro- and nanorobotic systems are an exciting frontier in robotics, with potential impacts in the fields of manufacturing and medicine. Chemists, biologists, and roboticists 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. At the macro-scale, automated control of devices floating in water in [31] and fluidic self assembly in [32] were presented, but as *stochastic* processes that can be controlled by turning a global signal on and off. We recently demonstrated that obstacles present in the workspace can *deterministically* break the symmetry of approximately identical robotic swarms, enabling positional configuration of robots [33]. Given sufficient 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]. 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, 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 [34], [35]. For recent advances in automated micro-assembly, see [36], but these techniques focus on a set of micro manipulators assembling one component at a time. This paper focuses on parallelizable techniques.

d) Assembly Planning: Algorithm techniques for optimizing assembly operations have a rich history, see review article [37]. Our paper determines if a polyomino has a feasible assembly sequence, similar to the planning in [38].

II. Theory: Polyomino Assembly by Global Control

This section explains how to design factories that build arbitrary-shaped 2D polyominoes. 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 a partially assembled polyomino and pass the polyomino to the next component.

A. Model

Assume the following rules: 1.) A planar grid *workspace* W is filled with a number of unit-square particles (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 particle 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 hit an obstacle, hit a stationary particle, or share an edge

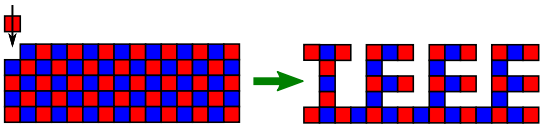


Fig. 2. Any polyomino can be constructed with two compatible robot species, shown here with red and blue tiles.

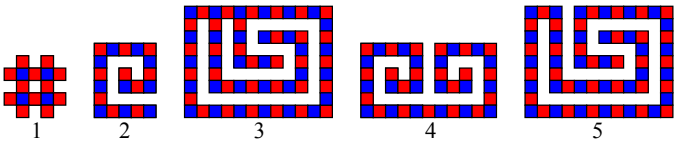


Fig. 3. Polyomino parts. Assembly difficulty increases from left to right.

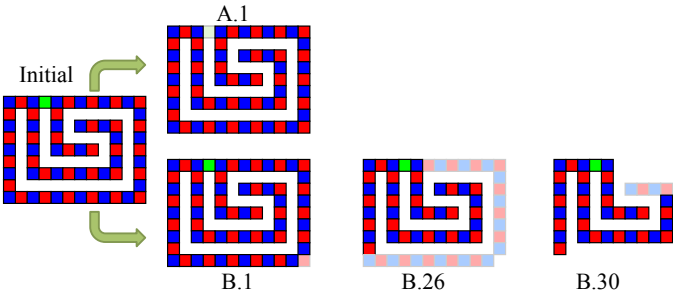


Fig. 4. Deconstruction order matters if loops are present. Loops occur when the 8-connected freespace has more than one connected component. In the top row the green tile is removed first, resulting in a polyomino that cannot be decomposed. However, if the bottom right tile is removed first, deconstruction is possible.

with a compatible particle. If a particle shares an edge with a compatible particle the two particles bond and from then on move as a unit. This paper uses *cycles* of movement commands in the order $\langle l, u, r, d \rangle$. We assume the area of W is finite and issue each command long enough for the particles to reach their maximum extent.

B. Arbitrary 2D shapes require two particle species

Polyominoes have *four-point connectivity*: a 4-connected square is a neighbor to every square that shares an edge with it.

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. Red and blue colors are used to indicate particles of different species. □

The sufficiency of two species to construct any shape gives many options for implementation. The two species could correspond to any gendered connection, including ionic charge, magnetic polarity, or hook-and-loop type fasteners. Large populations of these two species can then be stored in separate hoppers and, like two-part epoxy, only assemble when dissimilar particles come in contact.

C. Complexity Handled in This Paper

2D part geometries vary in difficulty. Fig. 3 shows parts with increasing complexity.

Label the first particle in the assembly process the *seed particle*. Part 1 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 contains a loop, and the interior must be finished before the loop is closed. Shape 4 is the combination of a left-handed and a right-handed spiral. Adding one particle at a time in 2D cannot assemble this part, 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. Shape 5 contains compound overhangs, and may be impossible to construct with additive 2D manufacturing using only two species. 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.

D. Discovering a Build Path

Given a polyomino, Alg. 1 determines if the polyomino can be built by adding one component at a time. The problem of determining a build order is difficult because there are $O(n!)$ possible build orders, and many of them may violate the constraints given in Section II-A. 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. However, as in many robotics problems, the inverse problem of deconstruction is easier than the forward problem of construction.

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

$\mathbf{P}$ is the x, y coordinates of a 4-connected polyomino. 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** $\{\mathbf{C}, \mathbf{c}, \mathbf{m}\}$

Alg. 1 first assigns each tile in the polyomino a color, then calls the recursive function DECOMPOSE, which returns either a build order of polyomino coordinates and the directions to build, or an empty list if the part cannot be constructed. DECOMPOSE starts by calling the function ERODE. ERODE first counts the number of components in the 8-connected freespace. An 8-connected square is a neighbor to every square that shares an edge or vertex with it. If there is more than one

connected component, the polyomino contains loops. ERODE 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 $\mathbf{R}$ sorted by the number of neighbors so a tile with one neighbor is checked before tiles with two or three. This *for 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, (2) that its removal does not fragment the remaining polyomino into more than one piece in line 10, and (3) that its removal does not break a loop in line 11. If no loops are present, this algorithm requires at most $n/2(1+n)$ iterations, because there are n particles to remove, and each iteration considers one less particle than the previous iteration.

Polyominoes with loops require care, because decomposing them in the wrong order can make disassembly impossible, as shown in Fig. 4. If loops exist then ERODE may return only a partial decomposition, so DECOMPOSE must then try every possible break point and recursively call DECOMPOSE until either a solution is found, or all possible decomposition orders have been tested. The worst-case number of function calls of DECOMPOSE are proportional to the factorial of the number of loops, $O(|8\text{-CONNCOMP}(\neg\mathbf{P})|!)$. Though large, this is much less than $O(n!)$.

Algorithm 2 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 encountered. $\mathbf{d} \leftarrow \{r, d, l, u\}$

1: $\mathbf{C} \leftarrow \{\}, \mathbf{m} \leftarrow \{\}, \ell \leftarrow$ FALSE, $\mathbf{R} \leftarrow \mathbf{P}$
2: $w \leftarrow |8\text{-CONNCOMP}(\neg\mathbf{R})|$
3: **while** $1 < |\mathbf{R}|$ **do**
4: *successRemove* $\leftarrow$ FALSE
5: $\mathbf{R} \leftarrow$SORT$(\mathbf{R})$ ▷ sort by number of neighbors
6: **for** $j \leftarrow 1, j \leq |\mathbf{R}|$ **do**
7: $\mathbf{p} \leftarrow \mathbf{R}_j, \mathbf{T} \leftarrow \mathbf{R} \backslash \mathbf{R}_j$
8: **for** $k \leftarrow 1, k \leq 4$ **do**
9: **if** CHECKPATHTILE$(\mathbf{T}, \mathbf{p}, \mathbf{d}_k, \mathbf{c})$ **and**
10: $1 = |4\text{-CONNCOMP}(\mathbf{T})|$ **then**
11: **if** $w = |8\text{-CONNCOMP}(\neg\mathbf{T})|$ **then**
12: $\mathbf{R} \leftarrow \mathbf{T}$, *successRemove* $\leftarrow$ TRUE
13: $\mathbf{C}_{1+|\mathbf{R}|} \leftarrow \mathbf{p}, \mathbf{m}_{|\mathbf{R}|} \leftarrow \mathbf{d}_k$
14: **else** $\ell \leftarrow$ TRUE
15: **break**
16: **if** *successRemove* $=$ FALSE **then**
17: $\mathbf{C} \leftarrow \{\}, \mathbf{m} \leftarrow \{\}$
18: **break**
19: **if** $|\mathbf{R}| = 1$ **then**
20: $\mathbf{C}_1 \leftarrow \mathbf{R}_1$
21: **return** $\{\mathbf{R}, \mathbf{C}, \mathbf{m}, \ell\}$

E. Assembling Tiles

Algorithm 3 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. $\mathbf{d} \leftarrow \{u, d, l, r\}$

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

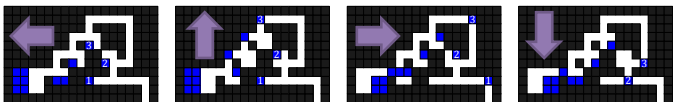


Fig. 5. Hopper with five delays. The hopper is filled with similarly-labelled robots that will not combine. Every clockwise command cycle releases one robot from the hopper.

1) Hopper Construction: Two-part adhesives react when components mix. Placing components in separate containers prevents mixing. Similarly, storing many particles 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. 5 releases one particle every cycle. Delay blocks are used to ensure the nth part hopper does not start releasing particles until cycle n. For ease of exposition, this paper has a unique hopper for each tile position. This enables precise positioning of different materials, but a particle logic system could use just two hoppers, similar to our particle logic systems in [9].

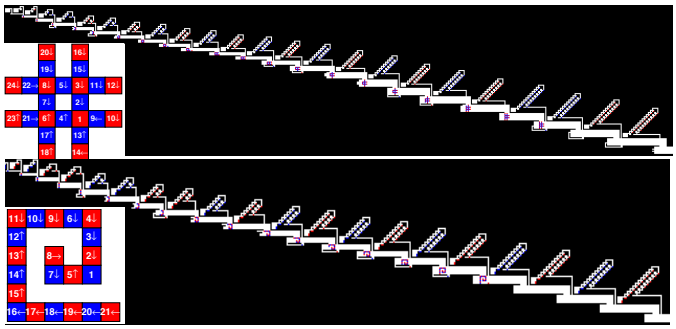


Fig. 6. A twenty-four tile factory, step 82 for a '#' shape and a twenty-one tile factory, step 66 for a spiral (zoom in for details in this vector graphic).

F. 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 $(n_c, \mathbf{b}, m, C, c, w)$ described in Alg. 5 is called to generate 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. 6.

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.

```
F ← {}                                   ▷ the factory obstacle array
{C, c, m} ← FINDBUILDPATH(P)
if {} = m then
    return F
{A, b} ←FACTORYFIRSTTILE(n_c, c_i, w)
for i ← 2, i ≤ |c|) do
    {A, b} ←FACTORYADDTILE(n_c, b, m_{i−1}, C_i, c_i, w)
    F ←CONCATFACTORIES(F, A)
return F
```

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

```
{hopper} ←HOPPER(c, n_c, w)
if m = d and (C_x ≤ max b_x or C_y < min b_y) then
    {A, b} ←DOWNDIR(hopper, b, C)
else if m = l and (C_y ≤ max b_y or C_x > max b_x) then
    {A, b} ←LEFTDIR(hopper, b, C)
else if m = l and (C_x ≥ max b_x or C_y > max b_y) then
    {A, b} ←UPDIR(hopper, b, C)
else if m = r and (C_y ≥ min b_y or C_x < min b_x) then
    {A, b} ←RIGHTDIR(hopper, b, C)
return {A, b}
```

III. ANALYSIS

This section analyzes the travel distance and space required for a factory and gives simulation results. Algorithms 1—5 were coded in MATLAB and are available at [39].

A. Maximum Distance Travelled

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 polyominoes are produced. Clever design of delays in the part hoppers ensures no unconnected tiles are released. During *production* cycles, one polyomino is finished each cycle. Once the first part hopper empties, the $n-1$ *wind down* cycles each produce a complete polyomino as each successive hopper empties. This section analyzes maximum distance, defined as the maximum distance any tile must move. There are two results, *construction distance*, the maximum distance required to assemble a single polyomino from scratch, and *cycle distance*, the maximum distance required during production cycles to advance all partial assemblies one cycle. Since a polyomino contains n tiles, the *construction distance* during production cycles is n·(*cycle distance*).

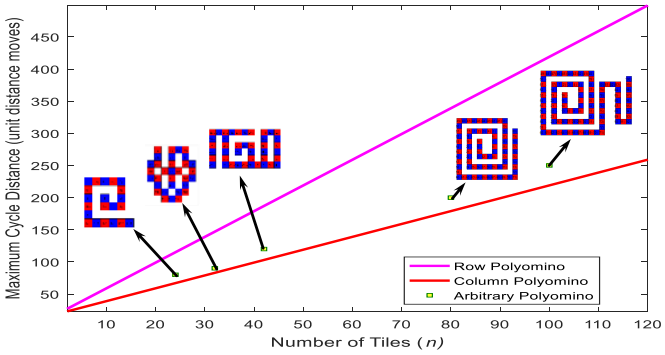


Fig. 7. Worst-case cycle distance plotted as a function of polyomino size n. The cycle distance is the sum of distances to move during the r, d, l, u moves each cycle. Cycle distance increases linearly with polyomino size and is upper bounded by row parts and lower bounded by column parts. Total construction distance for a particle is n·cycle distance.

Cycle distance is the sum of the maximum distances moved in each direction. As shown in Fig. 7, polyominoes shaped as a $n \times 1$ row require the longest distance of $4n+16$. Polyominoes shaped as a $1 \times n$ column require the least distance of $2n+16$. Construction distance therefore requires $O(n^2)$ distance.

B. Space Required

The space required by a factory is a function of the widths of individual sub-assemblies and height of the last sub-assembly.

The 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 the number of copies n_c of the polyomino, width of the hopper w, position of the sub-assembly i, and rows of the sub-assembled polyomino $\mathbf{b}_y$ as in (1). If a tile is added before the top row of $\mathbf{b}$, then an additional row is added to the height. The width of the sub-assembly can be calculated similarly as in (2) and (3). In a case where twice of $\mathbf{b}_x$ is greater than $width_{\text{hopper+delays}}$ then additional columns are added to the left of the subassembly. When a tile is added to $\mathbf{b}$ using a down move, width also depends on the location of the column, $column_{\text{loc}}$, to which the tile is added.

$$height(i) = \left\lceil \frac{n_c}{w} \right\rceil + 2\left(\left\lceil \frac{i}{2} \right\rceil + \mathbf{b}_y \right) + \begin{cases} 4, & \text{for } m = l \text{ or } d, i \geq 2 \\ 7, & \text{for } m = u \text{ or } r, i \geq 2 \end{cases} \quad (1)$$

$$width_{\text{hopper+delays}} = w + 2\left\lceil \frac{i}{2} \right\rceil + 8, i \geq 2 \quad (2)$$

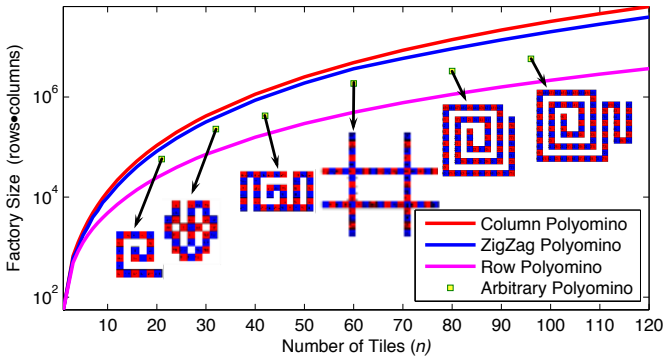


Fig. 8. Factory size grows quadratically with the number of tiles, and is upper bounded by column-shaped polynominos and lower bounded by row-shaped polyominos.

$$width(i) = width_{\text{hopper+delays}} + \begin{cases} (\mathbf{b}_x - column_{\text{loc}}), & \text{for } m = d \\ 0 & \text{for } m \neq d \end{cases} \quad (3)$$

Because a factory requires $O(n)$ rows and $O(n)$ columns, the total required space is $O(n^2)$. As shown in Fig. 8, the required size is upper bounded by column-shaped polynominoes and lower bounded by row-shaped polyominoes, and is $O(n^2)$.

IV. Experiment

To demonstrate Algs. 1–5, we developed two platforms at two size scales, a macro-scale demonstration board using gravity as the external force and magnetic attraction between red and blue particles for assembly, and a micro-scale magnetic control stage with alginate micro-particles.

A. Macro-scale, Gravity-Based Prototype

The gravity-based model shown in Fig. 9 uses a white workspace, red sliders for particles with magnetic north out, blue sliders for particles with magnetic south out, and black stop blocks for workspace obstacles. This model uses gravity as a global input to manipulate the red and blue sliders.

a) Construction and assembly: The macro-scale, reconfigurable, gravity-based model used to demonstrate parallel assembly was manufactured from laser cut acrylic, plastic dowel rods, and 3.2×3.2×1.6 mm^3 neodymium magnets. The workspace was made from a 0.6 by 0.3 meter sheet of 6.35 mm thick white acrylic. A laser cutter was used to make a grid of slider tracks 3.25 mm deep and 3.25 mm wide in the workspace as well as four holes with a diameter of 3.2 mm around each intersection of the grid for stop blocks to be securely placed. The stop blocks are made of similar black acrylic with four plastic dowel rods so they may be securely placed onto the workspace. The particles were made from similar red and blue acrylic sheets and are approximately 25 mm in diameter. The sliders have eight laser cut slots to house the magnets and have a small plastic dowel rod inserted in the center to ensure the sliders follow the tracks of the workspace.

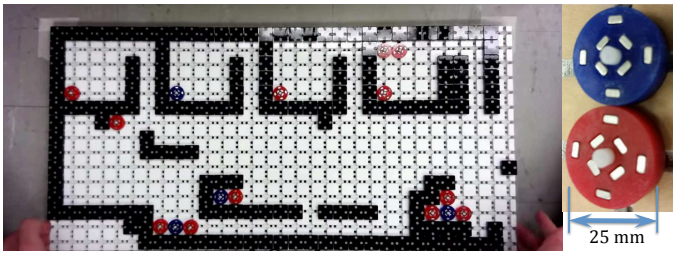


Fig. 9. A macro-scale demonstration of particle assembly using gravity as the external force and magnetic attraction between red and blue particles for assembly. Inset shows details of the magnetic sliders with magnets of opposite polarity facing outwards. See video attachment for a demonstration.

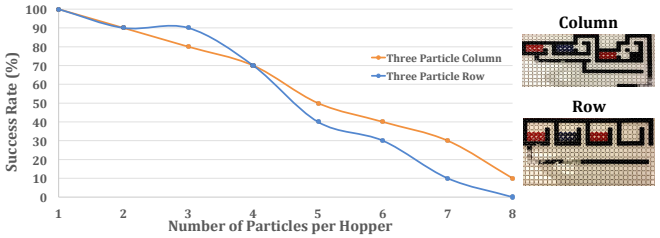


Fig. 10. Results from assembly of macro-scale, three tile row and column polyominoes. Each data point represents 10 trials.

b) Forces Involved: When the macro-scale demonstration is tilted at an angle of 20° most of the sliders will break free from the average static friction force of 0.0074 N and move across the workspace. At this angle the average force of weight contributing to the motion of the sliders is 0.0092 N, just enough to overcome the friction. Since the average magnetic breaking strength of the sliders is 0.1 N, sliders of opposite charge should be able to connect and overcome the force of motion of the sliders. However, there are instances where this connection does not overcome the force of motion due to a high tilt angle needed to break static friction.

c) Macro Scale Results: Fig. 10 shows results of experimentation for a three tile row and column polyominoes. Success rate is high when the number of sliders in each hopper is small. This is because the system was designed for a small number of particles and the magnetic repulsion of like particles can misalign the sliders.

B. Micro-scale, Magnetic-Based Prototype

We designed a custom magnetic control stage to generate the global control inputs. This stage generates a magnetic drag force by moving a permanent magnet.

a) Experimental setup: Figure 11 shows a system schematic. The permanent magnet can translate in x and y-axes, actuated by stepper motors and moving on linear rails. The neodymium permanent magnet field strength is 1.32 T and dimensions are 50.8 × 50.8 mm^2 (K&J Magnetics). The microfluidic factory layout produced for this experiment was fabricated through traditional photolithography methods. A silicon wafer was selected as the microfabrication substrate. SU-8 2150 photoresist (MicroChem) was then spin coated onto the substrate, giving a thickness of 300 μm. The channel width is 500 μm. Channels were then filled with motility buffer composed of Dionized Water and 10% Tween 20. All microrobots used for these experiments were loaded alginate

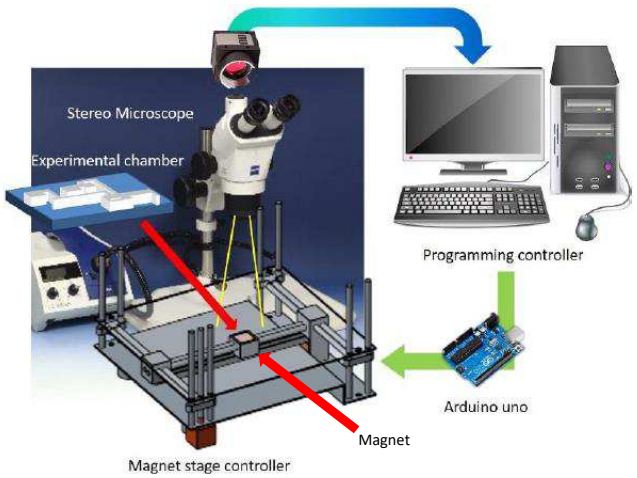


Fig. 11. Experimental platform.

paramagnetic hydrogels, otherwise known as artificial cells. Alginate microrobots can encapsulate both organic and nonorganic materials, which makes them the best suited form of microrobots to create different types of species. The alginate microrobots were fabricated using a centrifugal method, using the following equation [35] to generate particles of diameter d_p:

$$d_p = \sqrt[3]{(6d_n\sigma_p)/(\rho_p g)} \tag{4}$$

where d_n, σ_p, ρ_p, and g are the diameter of the nozzle, surface tension of the alginate solution, density of alginate solution, and the applied gravitational force, respectively. The surface tension of alginate is 65.46 mN/m, and a density of 1.1 g/cm^3. The average microrobot size is 300 μm, and were composed of a concentration of 5% (w/v) Alginate-Na and 5% (w/v) concentration of CaCl2, and then encapsulated with 10% (w/v) nano-paramagnetic particles (Iron oxide, Sigma-Aldrich). Alginate microrobots were transported at each hopper in the microfluidic factory layout, by way of a pipette. To show the process, one alginate particle was loaded in each hopper. The experimental channel was placed at the center of the stage with the magnet centered beneath the microfluidic factory layout. This position was saved as the home position for the permanent magnet. Stepper motors controlled the stage position. An Arduino UNO programmed in C++ commanded these motors using a 2 Hz control loop. After a command was initiated, such as each direction in the $\langle r, d, l, u\rangle$ cycle, the permanent magnet returned to the home position. A non-zero magnetic gradient in the horizontal plane is only generated when the magnet moves out of its home position. The layout was observed through a stereomicroscope and the installed camera (Motion Pro X3) captured the procedure at 30 fps. The observed field of view at 0.65$\times$ magnification is 23.6$\times$18.9 mm^2.

b) Experimental result: Using a factory layout generated by Alg. 4, we demonstrated micro-scale assembly using multiple alginate microrobots. The initial scene is shown in Fig. 12(a). The first assembly operation was then orchestrated by moving the magnet in a clockwise direction, following the

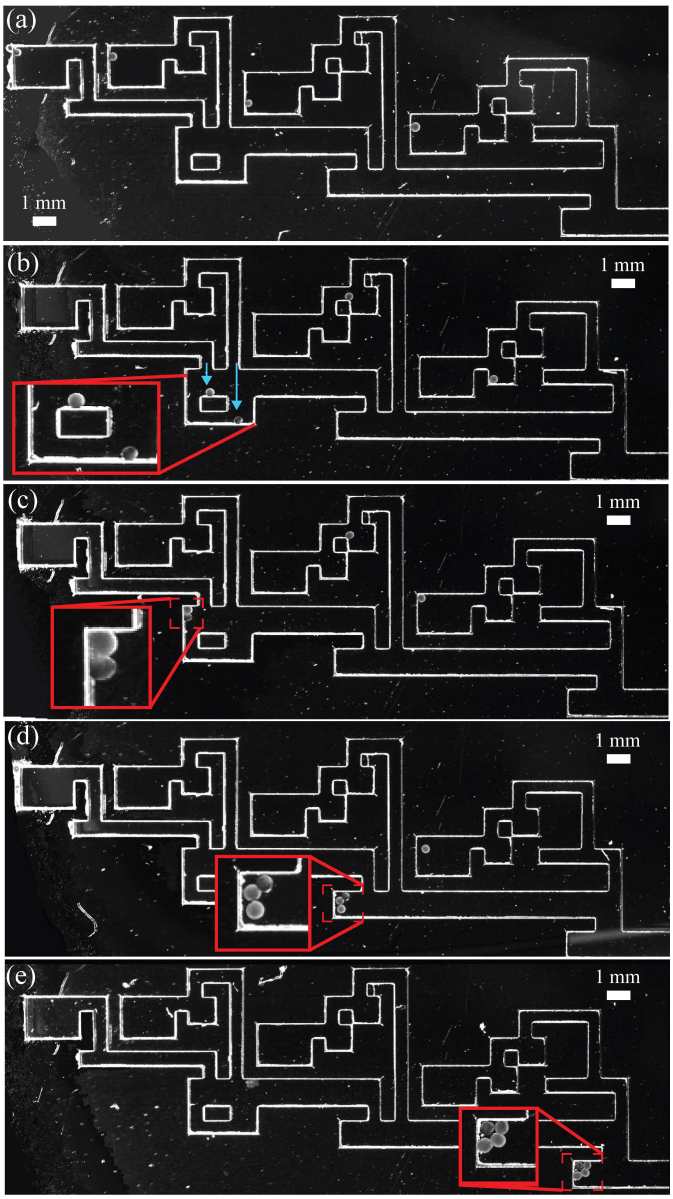


Fig. 12. Experimental results of Alg. 4. (a) shows individual alginate particles in initial positions. (b) After initial movements of $\langle r, d, l, u, r, d\rangle$, the alginate microrobots move to the position shown. (c) After $\langle l, u\rangle$ inputs, the system produces the first multi-microrobot polyomino. (d) Shows the next three microrobot polyomino after applying multiple $\langle r, d, l, u\rangle$ cycles. (e) After the alginate microrobots have moved through the microfluidic factory layout, the final 4-particle polyomino is generated.

$\langle r, d, l, u\rangle$ cycle as indicated in Fig. 12(b-d). Each input was applied sufficiently long to ensure all alginate microrobots touched a wall. Completion of the square polyomino is shown in Fig. 12(e).

V. 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.

Future work could extend Algorithms 1–5 to three dimensions. Additional work could focus on reducing the number of

cycles. To build a polyomino, our current algorithm requires n cycles. Parts could be decomposed into subassemblies, which would enable more complex parts to be created and enable construction in logarithmic number of cycles. Future work should also increase the robustness of micro- and macro-scale assembly. Furthermore, techniques to improve particle movement speed should be investigated.

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 *IEEE International Conference on Robotics and Automation*, 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 *IEEE International Conference on Robotics and Automation*. 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] G. Mermoud, M. Mastrangeli, U. Upadhyay, and A. Martinoli, "Real-time automated modeling and control of self-assembling systems," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 4266–4273.

[32] M. Mastrangeli, F. Schill, J. Goldowsky, H. Knapp, J. Brugger, and A. Martinoli, "Automated real-time control of fluidic self-assembly of microparticles," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5860–5865.

[33] 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 *IEEE/RSJ International Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 520–527.

[34] 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.

[35] 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.

[36] J. Seymour and D. J. Cappelleri, "Automated microassembly sequence planning with sub-assemblies," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2016, pp. V004T08A018–V004T08A018.

[37] M. F. F. Rashid, W. Hutabarat, and A. Tiwari, "A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches," *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 1, pp. 335–349, 2012.

[38] Q. Su, "Computer aided geometric feasible assembly sequence planning and optimizing," *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 1, pp. 48–57, 2007. [Online]. Available: http://dx.doi.org/10.1007/s00170-006-0447-0

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