Search by a Metamorphic Robotic System in a Finite 3D Cubic Grid

Ryonosuke Yamada
Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan

Yukiko Yamauchi
Faculty of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan

Abstract

We consider search in a finite 3D cubic grid by a *metamorphic robotic system* (MRS), that consists of anonymous modules. A module can perform a sliding and rotation while the whole modules keep connectivity. As the number of modules increases, the variety of actions that the MRS can perform increases. The *search problem* requires the MRS to find a target in a given finite field. Doi et al. (SSS 2018) demonstrate a necessary and sufficient number of modules for search in a finite 2D square grid. We consider search in a finite 3D cubic grid and investigate the effect of common knowledge. We consider three different settings. First, we show that three modules are necessary and sufficient when all modules are equipped with a *common compass*, i.e., they agree on the direction and orientation of the x, y, and z axes. Second, we show that four modules are necessary and sufficient when all modules agree on the direction and orientation of the vertical axis. Finally, we show that five modules are necessary and sufficient when all modules are not equipped with a common compass. Our results show that the shapes of the MRS in the 3D cubic grid have richer structure than those in the 2D square grid.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Computer systems organization → Robotic autonomy

Keywords and phrases Distributed system, metamorphic robotic system, search, and 3D cubic grid

Digital Object Identifier 10.4230/LIPIcs.SAND.2022.20

Funding This work was supported by JSPS KAKENHI Grant Numbers JP18H03202 and JST SICORP Grant Number JPMJSC1806, Japan.

1 Introduction

Swarm intelligence has shed light to collective behavior of autonomous entities with simple rules, such as ant, boid, and particles. The notion is applied to a collection of robots and swarm robotics has attracted much attention in the past two decades. Each autonomous element of the system is called a robot, module, agent, process, and sensor, and a variety of swarm robot systems have been investigated such as the *autonomous mobile robot system* [16], the *population protocol model* [3], the *programmable particles* [5], *Kilobot* [15], and *3D Catoms* [17]. Dumitrescu et al. considered the *metamorphic robotic system* (MRS), that consists of a collection of *modules* in the infinite 2D square grid [10, 9]. The modules are *anonymous*, i.e., they are indistinguishable. They are *autonomous* and *uniform*, i.e., each module autonomously decides its movement by a common algorithm. They are *oblivious*, i.e., each module has no memory of past. Thus, each module decides its behavior by observing other modules in nearby cells. Each module can perform a *sliding* to a side-adjacent cell and a *rotation* by 90 degrees around a cell. The modules must keep *connectivity*, which is defined by side-adjacency of cells occupied by modules. The authors considered *reconfiguration*, that requires the MRS to change the initial shape to a specified final shape [10]. They showed that any horizontally convex connected initial shape of an MRS can be transformed to any convex final shape via a straight chain shape. Later Dumitrescu and Pach showed that any connected initial shape can be transformed to any connected final shape via a straight chain shape [8]. In other words, the MRS has the ability of "universal reconfiguration."

Reconfiguration can generate dynamic behavior of the MRS. Dumitrescu et al. demonstrated that the MRS can move forward by repeating a reconfiguration [9]. They showed a reconfiguration that realizes the fastest *locomotion*. Doi et al. pointed out that the oblivious modules can use the shape of the MRS as global memory and the MRS can solve more complicated problems as the number of modules increases. They investigated *search* in a finite 2D square grid, that requires the MRS to find a target cell in a finite rectangle *field* [7]. Each module does not know the position of the target cell or the initial configuration of the MRS. They showed that if the modules agree on the cardinal directions (i.e., north, south, east and west), three modules are necessary and sufficient, otherwise five modules are necessary and sufficient. Nakamura et al. considered *evacuation* of the MRS from a finite rectangular field in the 2D square grid [14]. There is a hole (i.e., two side-adjacent cells) on the wall of the field, and the MRS is required to exit from it from an arbitrary initial position and arbitrary initial shape. They showed that two modules are necessary and sufficient when the modules agree on the cardinal directions, otherwise four modules are necessary and sufficient.

In this paper, we investigate the effect of common knowledge on search by the MRS in the 3D cubic grid. We consider the following three cases: (*i*) modules equipped with a common *compass* (i.e., they agree on the direction and orientation of x, y, and z axes). (*ii*) modules equipped with a common vertical axis (i.e., they agree on the direction and orientation of z axis), and (*iii*) modules not equipped with a common compass (i.e., they have no agreement on directions and orientations). We demonstrate that three modules are necessary and sufficient when the modules are equipped with a common compass and five modules are necessary and sufficient when the modules are not equipped with a common compass. The numbers of sufficient modules in the 3D cubic grid are the same as those in the 2D square grid [7] because the MRS has more states in the 3D cubic grid than in the 2D square grid. For the intermediate case with a common vertical axis, we demonstrate that four modules are necessary and sufficient. Thus, our results in the 3D cubic grid show a smooth trade-off between the computational power of the MRS and common knowledge among modules, that the previous results in the 2D square grid could not find. We present search algorithms for these three settings and show the necessity by examining the state transition graph of the MRS.

Related work. Reconfiguration of swarm robot systems have been discussed for the MRS [8, 10], autonomous mobile robots [11, 16] and programmable particles [6, 12]. Michail et al. considered the *programmable matter system*, that is similar to the MRS and investigated reconfiguration by rotations only and that by rotations and slidings [13]. They showed that the combination of rotations and slidings guarantees universal reconfiguration, while rotations only cannot. They also presented $O(n^2)$-time reconfiguration algorithm by rotations and slidings, where n is the number of computing entities. Almethen et al. considered reconfiguration by *line-pushing*, where each module is equipped with the ability of pushing a line of modules [2]. They presented $O(n \log n)$-time universal reconfiguration algorithm that does not promise connectivity of intermediate shapes and $O(n\sqrt{n})$-time reconfiguration algorithm that transforms a diagonal line into a straight chain with preserving connectivity. The same authors later showed that their programmable matter system has the ability of universal reconfiguration and $O(n\sqrt{n})$-time reconfiguration algorithm together with $\Omega(n \log n)$ lower bound [1].

Little has been discussed for memory complexity of swarm robot systems. The autonomous mobile robot system consider extreme cases, where all robots are equipped with either no memory or unlimited memory [16]. Das et al. introduced the *luminous robot model*, where each robot is equipped with a light [4]. A luminous robot can change the color of its light and the color of the light can be observed by other robots. Thus, a luminous robot has a state. The authors showed that the robots can be synchronized by a constant number of colors. Doi. et al. pointed out that the number of memory-less modules of the MRS can be considered as an indicator of memory complexity [7]. The existing two papers [7, 14] demonstrated the relationship between search and evacuation in 2D square grid. Finally, the programmable particle system [5] consider computing entities with constant-size memory.

2 Preliminary

We consider a *metamorphic robotic system* (MRS) in a finite 3D cubic grid. A metamorphic robotic system consists of a collection of anonymous (i.e., indistinguishable) *modules*. A module can observe the positions of other modules in nearby cells, computes its next movement with a common algorithm, and performs the movement.

Each cell of the 3D cubic grid can adopt at most one module at a time. Cell (x, y, z) is the cell surrounded by grid points (x, y, z), $(x+1, y, z)$, $(x, y+1, z)$, $(x, y, z+1)$, $(x+1, y+1, z)$, $(x+1, y, z+1)$, $(x, y+1, z+1)$, and $(x+1, y+1, z+1)$. Cells $(x+1, y, z)$, $(x, y+1, z)$, $(x, y, z+1)$, $(x-1, y, z)$, $(x, y-1, z)$, $(x, y, z-1)$ are *side-adjacent* to cell (x, y, z). We consider the positive x direction as East, the positive y direction as North, and the positive z direction as Up.

The MRS moves in a finite *field*, which is a cuboid of width w, depth d and height h with its two diagonal cells being $(0, 0, 0)$ and $(w-1, d-1, h-1)$. We consider two types of *planes*; the first type is a set of cells forming a plane perpendicular to one of the x, y, and z axes. The second type is a set of cells parallel to one of the x, y, and z axes and diagonal to the remaining two axes. For example, $\{(x, y, z) | y = s\}$ for some $s \in \mathcal{Z}$ is a plane of the first type and $\{(x, y, z) | x+y = s'\}$ for some $s' \in \mathcal{Z}$ is a plane of the second type. A *line* of cells is a set of cells forming a horizontal or vertical line on a plane. For example, $\{(x, y, z) | y = u, z = v\}$ for some $u, v \in \mathcal{Z}$ is a line and $\{(x, y, z) | x+y = u', z = v'\}$ for some $u', v' \in \mathcal{Z}$ is a line.

The field is surrounded by six planes, which we call *walls*. More precisely, the walls are $\{(x, y, z) | x = -1\}$ (the West wall), $\{(x, y, z) | y = -1\}$ (the South wall), $\{(x, y, z) | z = -1\}$ (the Bottom wall), $\{(x, y, z) | x = w\}$ (the East wall), $\{(x, y, z) | y = d\}$ (the North wall), and $\{(x, y, z) | z = h\}$ (the Top wall).

All modules synchronously perform observation, computation, and movement in each discrete time $t = 0, 1, 2, \ldots$. A *configuration* of the MRS is the set of cells occupied by the modules. We say two modules are *side-adjacent* if they are in the two side-adjacent cells. We also say that a module m is side-adjacent to cell c if the cell occupied by m is side-adjacent to c. Given a configuration of the MRS, consider a graph where each vertex corresponds to a module and there is an edge between two vertices if the corresponding modules are side-adjacent. If this graph is connected, we say the MRS is *connected*.

A module can perform two types of movements, *sliding* and *rotation*.

1. Sliding: When two modules m_i and m_j are side-adjacent, another module m_k can move from a cell side-adjacent to m_i to an empty cell side-adjacent to m_k and m_j along m_i and m_j. During the movement, m_i and m_j cannot perform any movement. See Figure 1 as an example.

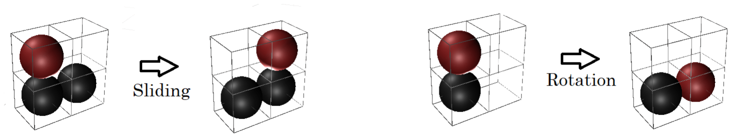

Figure 1 Sliding and rotation. The red modules perform movements.

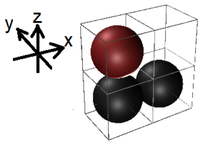

Figure 2 Example of an observation at the red module with a local coordinate system.

2. Rotation: When two modules m_i and m_j are side-adjacent, m_i can move to a cell side-adjacent to m_j by rotating $\pi/2$ in some direction. There are six cells side-adjacent to m_j and m_i can move to four of them by rotation. During the movement, m_j cannot move and the cells that m_i passes must be empty. See Figure 1 as an example.

Note that several modules can move at the same time as long as their moving tracks do not overlap. The modules must keep two types of connectivity at each time step.

1. At the beginning of each time step, the modules must be connected.
2. At each time step, the modules that do not move must be connected.

We assume that each module obtains the result of an observation and moves to the next cell in its *local x-y-z coordinate system.* We assume that the origin of the local coordinate system of a module is its current cell and all local coordinate systems are right-handed. In this paper, we consider three types of MRSs with different degree of agreement on the coordinate system. When all modules agree on the directions and orientations of x, y, and z axes, we say the MRS is equipped with a *common compass.* When all modules do not agree on the directions or orientations of x, y, and z axes, we say the MRS is not equipped with a common compass. Hence, local coordinate systems are not consistent among the modules. As an intermediate model, we consider modules that agree on the direction and orientation of the vertical axis. In this case, we say the MRS is equipped with a *common vertical axis.* The *state* of the MRS is its local shape. If the modules are equipped with a common compass or a common vertical axis, the state of the MRS contains common directions and orientations. Otherwise the state of the MRS does not contain any directions and orientations.

The *search problem* requires the MRS to find a *target* placed at one cell in the field from a given initial configuration. We call the cell containing the target the *target cell.* The MRS *finds* a target when one of its modules enters the target cell.

When a module executes the common algorithm, the input is the *observation* of cells in a cube of size $(2k+1)\times(2k+1)\times(2k+1)$ centered at the module (i.e., its *k-neighborhood*). The value of k is fixed by the algorithm. A module detects whether each cell in its k-neighborhood is a wall cell or not and whether it is occupied by a module or not. Let C_m be the set of cells occupied by some modules, C_w be the set of wall cells, and C_e be the set of the remaining (i.e., empty) cells of an observation. More precisely, each set of cells is a set of coordinates of the corresponding cells observed in the local coordinate system of the module. For example, in Figure 2, the result of an observation at the red module is $C_m = \{(0,-1,0),(1,-1,0)\}$, $C_w = \emptyset$, and C_e is the remaining cells. When a common algorithm outputs coordinate (a,b,c) at a module, the module moves to (a,b,c) in its local coordinate system.

When we describe an algorithm, the elements of C_m, C_w, and C_e are specified in a "canonical coordinate system," i.e., the global coordinate system. When the modules are equipped with a common compass, without loss of generality, we assume that the common compass is identical to the global coordinate system. Thus, each module computes its movement by checking C_m, C_w, and C_e. When the modules agree on a common vertical axis, without loss of generality, we assume that the vertical axis is identical to z axis of the global coordinate system. Each module computes its movement by rotating the current observation by $\pi/2$, π, $3\pi/2$, and 2π around the common vertical axis and comparing the results with C_m, C_w, and C_e. It selects an output with a movement and if there are multiple outputs with movement it nondeterministically selects one of them. When the modules are not equipped with a common compass, a module checks 24 rotations of the current observation and selects an output in the same way as the above case.

3 Search algorithms for an MRS in a finite 3D cubic grid

In this section, we present search algorithms with small number of modules. Our proposed algorithms are based on a common strategy. Since the MRS does not know the position of the target cell, we make the MRS visit all cells of the field. The proposed algorithms slice the field into planes and the MRS visits the cells of each plane by sweeping each row or column of the plane. Thus, the proposed algorithms are extensions of the search algorithms by Doi et al. in a finite 2D square grid [7].

3.1 Search with a common compass

We show the following theorem by a search algorithm for the MRS of three modules equipped with a common compass.

▶ **Theorem 1.** *The MRS of three modules equipped with a common compass can solve the search problem in a finite 3D cubic grid from any initial configuration.*

The proposed algorithm considers planes that is parallel to the z-axis and the angles between it and each of x-axis and y-axis are $\pi/2$. Each plane is represented as $\{(x,y,z)|x+y = s\}$ for $s = 1, 2, \dots$. The MRS moves along each line parallel to the x-y plane $\{(x,y,z)|x+y = s, z = t\}$ for $t = 0, 1, 2, \dots$. Figure 3 shows a moving track of the proposed algorithm.

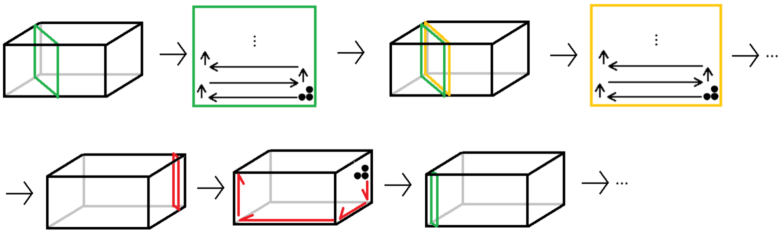

Figure 3 Search by three modules equipped with a common compass.

The MRS continues to search each plane until it reaches the northeasternmost plane. Then, it moves along the edges of the field so that it returns to the southwesternmost plane. It starts searching each plane again to visit all cells of the field.

The MRS moves forward or turns by repeating a sequence of movements, that we call a *move sequence*. The proposed algorithm consists of the following move sequences.

- Move sequence M_{NW} (Figure 4). The blue module is in cell (x, y, z) at first. By this move sequence, the green module reaches cell $(x-1, y+1, z)$. By repeating M_{NW} n times, some modules visit the cells $(x-k, y+k, z)(0 \leq k \leq n)$. That is, it visits all the cells of the horizontal line $\{(x, y, z)|x+y=s, z=t\}$.
- Move sequence M_{TurnNW} (Figure 5). By this move sequence, the MRS changes its move sequence from M_{NW} to M_{SE}.
- Move sequence M_{SE} (Figure 6). The blue module is in cell (x, y, z) at first. By this move sequence, the green module reaches cell $(x+1, y-1, z)$. By repeating M_{SE} n times, some modules visit the cells $(x+k, y-k, z)(0 \leq k \leq n)$. That is, it visits all the cells of the horizontal line $\{(x, y, z)|x+y=s, z=t\}$.
- Move sequence M_{TurnSE} (Figure 7). By this move sequence, the MRS changes its move sequence from M_{SE} to M_{NW}.
- Move sequence M_T (Figure 8). By this move sequence, the MRS changes its move sequence from M_{SE} to M_D.
- Move sequence M_D (Figure 9). The blue module is in cell (x, y, z) at first. By this move sequence, the green module reaches cell$(x, y, z-1)$. By repeating M_D n times, some modules visit the cells $(x, y, z-k)(0 \leq k \leq n)$. That is, it visits all the cells of the line $\{(x, y, z)|x=s, y=t\}$.
- Move sequence M_B (Figure 10). By this move sequence, the MRS changes its move sequence from M_D to M_{NW}.
- Move sequence $M_{NECorner}$ (Figure 11). By this move sequence, the MRS changes its move sequence from M_D to $M_{WallBottom}$.
- Move sequence $M_{WallBottom}$ (Figure 12). The blue module is in cell $(x, y, 0)$ at first. By this move sequence, the green module reaches cell $(x, y-1, 0)$ along the edge. By repeating $M_{WallBottom}$ n times, some modules visit the cells $(x, y-k, 0)(0 \leq k \leq n)$. That is, it visits all the cells of the line $\{(x, y, z)|x=s, z=0\}$.
- Move sequence $M_{SWCorner}$ (Figure 13). By this move sequence, the MRS changes its move sequence from $M_{WallBottom}$ to M_{Up}.
- Move sequence M_{Up}. (Figure 14) The blue module is in cell $(0, 0, z)$ at first. By this move sequence, the green module reaches cell $(0, 0, z+1)$. By repeating M_{Up} n times, some modules visit the cells $(0, 0, k)(0 \leq k \leq n)$. That is, it visits all the cells of the line $\{(x, y, z)|x=0, y=0\}$.

The proposed algorithm consists of the following seven steps.

Step 1 The MRS repeats M_{NW}, that makes it move in the northwest direction along a horizontal line on a plane $\{(x, y, z)|x+y=s\}$ for some s.

Step 2 When the MRS reaches the north or west wall, it changes the moving direction to southeast by M_{TurnNW}.

Step 3 The MRS repeats M_{SE}, that makes it move in the southeast direction along a horizontal line on $\{(x, y, z)|x+y=s\}$. This movement makes the MRS move along the same horizontal line as Step 1.

Step 4 If the MRS is adjacent to the top wall it moves to the plane $\{(x, y, z)|x+y=s+1\}$ Then, it repeats M_D shown in Figure 9, that makes it move down along the south wall or east wall until it reaches the bottom wall. Then, it leaves the wall by M_B. It starts searching the new plane by repeating Steps 1, 2, 3, and 4. Otherwise, it proceeds to Step 5.

Step 5 When the MRS reaches the south or east wall, it moves to the row above by M_{TurnSE}. Then, it repeats Steps 1, 2, and 3 so that it visits all cells on the new horizontal line.

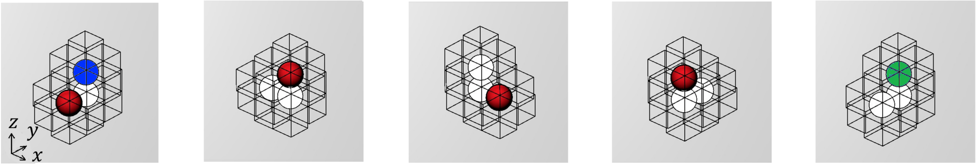

Figure 4 Move to northwest. In each figure, the red module moves. When the blue module is in cell (x, y, z), after this move sequence, the green module reaches cell $(x-1, y+1, z)$.

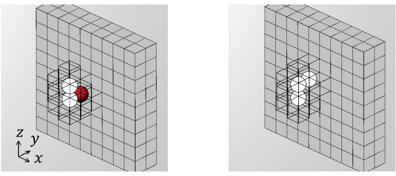

Figure 5 Turn on the north or west wall. In the first figue, the red module moves.

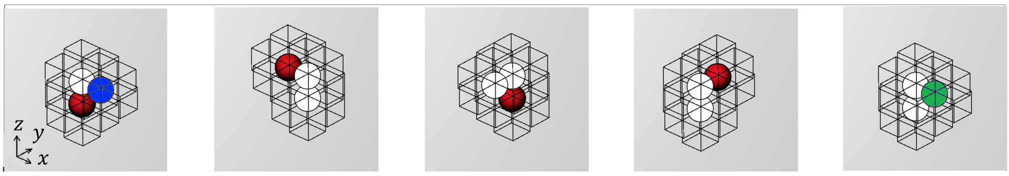

Figure 6 Move to southeast. In each figure, the red module moves. When the blue module is in cell (x, y, z), after this move sequence, the green module reaches $(x+1, y-1, z)$.

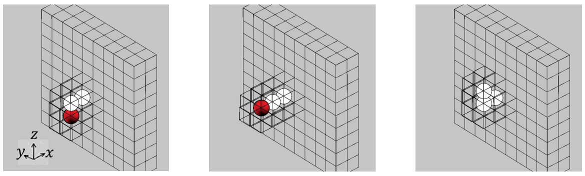

Figure 7 Turn on the south or east wall. In each figure, the red module moves.

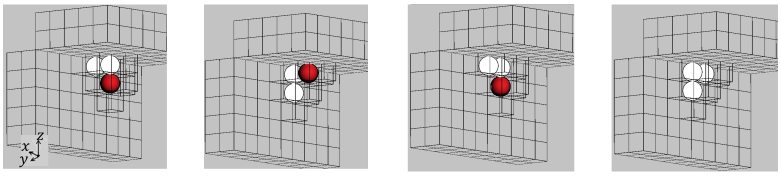

Figure 8 Move around the top of the north or east wall. In each figure, the red module moves.

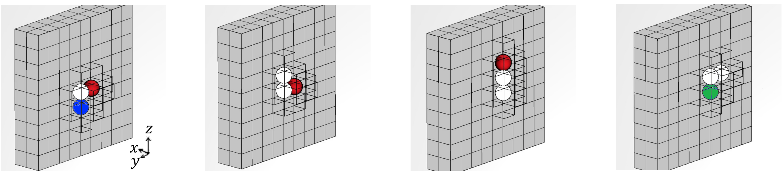

Figure 9 Move down on the north or east wall. In each figure, the red module moves. When the blue module is in cell (x, y, z), after this move sequence, the green module reaches cell $(x, y, z-1)$.

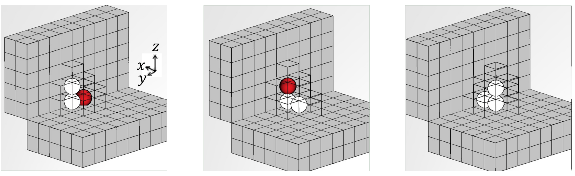

Figure 10 Leaving the bottom of the north or east wall. In each figure, the red module moves.

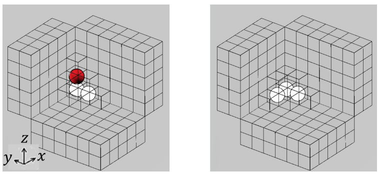

Figure 11 Move on the northeast corner. In the first figure, the red module moves.

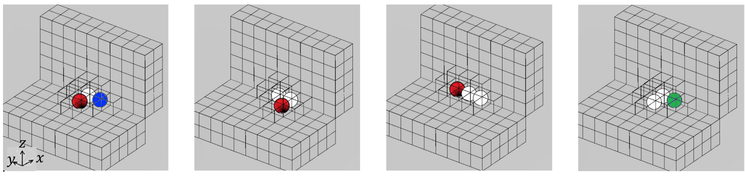

Figure 12 Move along the bottom of the wall. In each figure, the red module moves. When the blue module is in cell $(x, y, 0)$, after this move sequence, the green module reaches cell $(x, y - 1, 0)$.

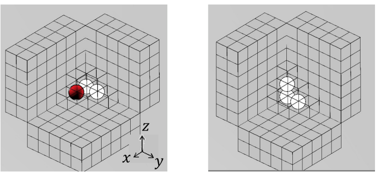

Figure 13 Move on the southwest corner. In the first figure, the red module moves.

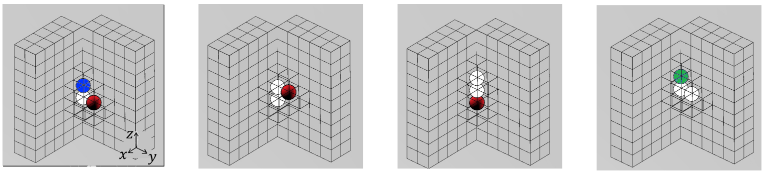

Figure 14 Move up on the southwest corner. In each figure, the red module moves. When the blue module is in cell $(0, 0, z)$, after this move sequence, the green module reaches cell $(0, 0, z + 1)$.

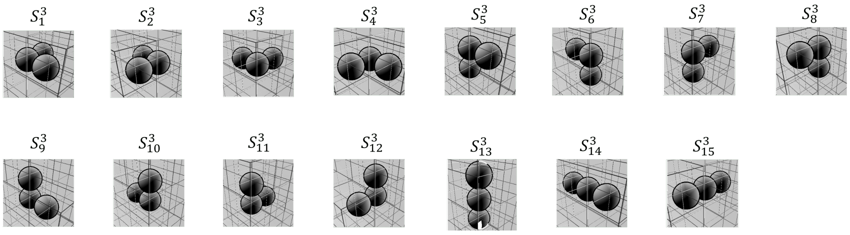

Figure 15 States of the MRS of three modules equipped with a common compass.

Step 6 When the MRS reaches the northeast corner of the top wall, the algorithm sends the MRS back to the southwest corner, where the MRS starts searching by repeating Steps 1 to 5. It moves along the northeast edge until it reaches the northeast corner of the bottom wall by M_D. Then, it moves along the east edge of the bottom wall until it reaches the south east corner of the bottom wall by $M_{NECorner}$ and repeating $M_{WallBottom}$. It moves along the south edge of the bottom wall until it reaches the southwest corner of the bottom wall by repeating $M_{WallBottom}$. . Finally, it moves along the southeast edge until it reaches the southwest corner of the top wall by $M_{SWCorner}$ and repeating M_{Up}. Then, the MRS returns to Step 4.

Table 1 and 2 show the input and the output of the proposed algorithm. Each element specifies a part of the input (especially, C_w and C_e), and the MRS does not care whether other cells than those specified are walls or not.

We briefly address the correctness of the proposed algorithm. When the MRS is on a plane $\{(x, y, z)|x + y = s\}$ for some s, it visits all cells in the horizontal line $\{(x, y, z)|x + y = s, z = t\}$ for some t by repeating Steps 1, 2, and 3. Then, it proceeds to the horizontal line $\{(x, y, z)|x + y = s, z = t + 1\}$ by Step 5. By repeating Steps 1, 2, 3, and 5, it eventually reaches the top wall. Then, it starts searching for cells in $\{(x, y, z)|x + y = s + 1\}$ by Step 4 and 6.

Repeating the above movement, the MRS eventually reaches the northeast corner of the top wall. At this point, it may have not yet visited the cells near the south west corner. Steps 6 enables the MRS visit these cells by moving it to the southwest corner of the top wall and starting Step 1 again.

There exist initial configurations that satisfies no condition of Table 1 and 2. We add exceptional transformation rules from such initial configurations. Figure 15 shows all states of three modules equipped with a common compass. Observe that any configuration can be transformed to another one in one time step. (Note that more than one module can move in one time.) Hence, even if the initial state of the MRS does not match any entry of Table 1 and 2, the MRS can be transformed into one of the entries and the MRS can start search from any initial configuration.

3.2 Search with a common vertical axis

We show the following theorem by a search algorithm for the MRS of four modules equipped with a common vertical axis.

▶ **Theorem 2.** *The MRS of four modules with a common vertical axis can solve a search problem in a finite 3D grid if no pair of modules have an identical observation in an initial configuration.*

Table 1 Search algorithm for the MRS equipped with a common compass (Former part).

	C_m	C_w	C_e	Output
M_{SE}	$(0,0,1)$,$(1,0,1)$		$(2,0,0)$	$(1,0,0)$
	$(1,0,0)$,$(1,0,-1)$			$(1,-1,0)$
	$(0,0,1)$,$(0,-1,1)$		$(0,-2,0)$	$(0,-1,0)$
	$(0,-1,0)$,$(0,-1,-1)$			$(1,-1,0)$
M_{TurnSE}	$(0,0,1)$,$(1,0,1)$	$(2,0,0)$	$(0,-1,0)$	$(-1,0,1)$
	$(1,0,0)$,$(2,0,0)$		$(0,0,-1)$, $(0,0,1)$	$(1,0,1)$
	$(0,0,1)$,$(0,-1,1)$	$(0,-2,0)$		$(0,1,1)$
	$(0,-1,0)$,$(0,-2,0)$	$(0,0,-1)$		$(0,-1,1)$
M_{NW}	$(-1,0,0)$,$(-1,0,1)$		$(0,-1,0)$	$(-1,1,0)$
	$(0,0,-1)$,$(0,1,-1)$		$(0,2,0)$	$(0,1,0)$
	$(0,1,0)$,$(0,1,1)$		$(1,0,0)$	$(-1,1,0)$
	$(0,0,-1)$,$(-1,0,-1)$		$(-2,0,0)$	$(-1,0,0)$
M_{TurnNW}	$(0,-1,0)$,$(0,-1,1)$	$(0,1,0)$		$(0,0,1)$
	$(1,0,0)$,$(1,0,1)$	$(-1,0,0)$		$(0,0,1)$
M_T	$(0,0,1)$,$(1,0,1)$	$(2,0,0)$,$(0,0,2)$		$(1,0,0)$
	$(1,0,0)$,$(1,0,-1)$	$(2,0,0)$,$(0,0,1)$		$(1,1,0)$
	$(0,0,1)$,$(0,1,1)$	$(1,0,0)$,$(0,0,2)$		$(0,1,0)$
	$(0,0,1)$,$(0,-1,1)$	$(0,0,2)$,$(0,-2,0)$		$(0,-1,0)$
	$(0,-1,0)$,$(0,-1,-1)$	$(0,0,1)$,$(0,-2,0)$		$(1,-1,0)$
	$(0,0,1)$,$(1,0,1)$	$(0,0,1)$,$(0,-1,0)$		$(1,0,0)$
M_D	$(0,1,0)$,$(0,1,-1)$	$(1,0,0)$		$(0,0,-1)$
	$(0,1,0)$,$(0,1,1)$	$(1,0,0)$	$(0,0,-1)$	$(0,1,-1)$
	$(0,0,-1)$,$(0,0,-2)$	$(1,0,0)$		$(0,-1,-1)$
	$(1,0,0)$,$(1,0,-1)$	$(0,-1,0)$		$(0,0,-1)$
	$(1,0,0)$,$(1,0,1)$	$(0,-1,0)$	$(0,0,-1)$	$(1,0,-1)$
	$(0,0,-1)$,$(0,0,-2)$	$(0,-1,0)$		$(-1,0,-1)$

We prove Theorem 2 by a search algorithm.

The proposed algorithm considers planes that is parallel to the x-z plane or the y-z plane. Each plane is represented as $x = s$ for $s = 0, 1, 2, \dots$ and $y = s'$ for $s' = 0, 1, 2, \dots$. The MRS moves along each vertical line $\{(x, y, z)|x = s, y = t\}$ when it is on the plane $x = s$ for $t = 0, 1, 2, \dots$ and $\{(x, y, z)|x = t', y = s'\}$ when it is on the plane $y = s'$ for $t' = 0, 1, 2, \dots$. Figure 16 shows an execution of the proposed algorithm.

The MRS continues to search each plane perpendicular to the x-axis until it reaches the east wall. Then, it starts to search each plane perpendicular to the y-axis. Repeating the process four times, it returns to its initial position.

The proposed algorithm consists of the following move sequences. We omit the detailed description and figures due to the page limitation.

- Move sequence M_{Down}. By this move sequence, the MRS visits all the cells of the horizontal line $\{(x, y, z)|x = s, y = t\}$.
- Move sequence M_{Up}. By this move sequence, the MRS visits all the cells of the horizontal line $\{(x, y, z)|x = s, y = t\}$.
- Move sequence M_{TurnU}. By this move sequence, the MRS changes its move sequence from M_{Up} to M_{Down}.

Table 2 Search algorithm for the MRS equipped with a common compass (Latter part).

	C_m	C_w	C_e	Output
M_B	$(0,1,0)$,$(0,1,1)$	$(1,0,0)$,$(0,0,-1)$	$(0,-1,0)$, $(0,2,0)$	$(-1,1,0)$
	$(0,0,-1)$,$(-1,0,-1)$	$(1,0,0)$,$(0,0,-2)$	$(0,-1,0)$	$(-1,0,0)$
	$(1,0,0)$,$(1,0,1)$	$(0,-1,0)$,$(0,0,-1)$		$(1,1,0)$
	$(0,0,-1)$,$(0,1,-1)$	$(0,-1,0)$,$(0,0,-2)$		$(0,1,0)$
$M_{NECorner}$	$(0,0,-1)$,$(0,-1,-1)$	$(0,1,0)$,$(1,0,0)$, $(0,0,-2)$		$(-1,0,-1)$
$M_{WallBottom}$	$(1,0,0)$,$(1,-1,0)$	$(0,0,-1)$		$(0,-1,0)$
	$(1,0,0)$,$(1,1,0)$	$(0,0,-1)$	$(0,-1,0)$	$(1,-1,0)$
	$(0,-1,0)$,$(0,-2,0)$	$(0,0,-1)$		$(-1,-1,0)$
	$(0,-1,0)$,$(-1,-1,0)$	$(0,0,-1)$,$(0,-2,0)$		$(-1,0,0)$
	$(0,-1,0)$,$(1,-1,0)$	$(0,0,-1)$	$(-1,0,0)$	$(-1,-1,0)$
	$(-1,0,0)$,$(-2,0,0)$	$(0,0,-1)$		$(-1,1,0)$
$M_{SWCorner}$	$(-1,0,0)$,$(-1,1,0)$	$(0,0,-1)$,$(-2,0,0)$, $(0,-1,0)$		$(-1,0,1)$
M_{Up}	$(0,-1,0)$,$(0,-1,1)$	$(-1,0,0)$,$(0,-2,0)$		$(0,0,1)$
	$(0,-1,0)$,$(0,-1,-1)$	$(-1,0,0)$,$(0,-2,0)$	$(0,0,1)$	$(0,-1,1)$
	$(0,0,1)$,$(0,0,2)$	$(-1,0,0)$,$(0,-1,0)$		$(0,1,1)$
	Otherwise	Otherwise	Otherwise	$(0,0,0)$

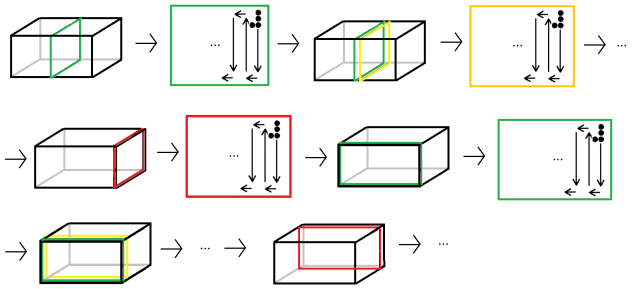

Figure 16 Example of a search by four modules.

- Move sequence M_{TurnD}. By this move sequence, the MRS changes its move sequence from M_{Down} to M_{Up}.
- Move sequence M_{B1}. By this move sequence, the MRS changes its move sequence from M_{Down} to M_{B2}.
- Move sequence M_{B2}. By this move sequence, the MRS visits all the cells of the horizontal line $\{(x,y,z)|y=s,z=0\}$.
- Move sequence M_{B3}. By this move sequence, the MRS changes its move sequence from M_{B2} to M_{Up}.
- Move sequence M_{Corner}. By this move sequence, the MRS changes its move sequence from M_{Down} to M_{B1}.

The proposed algorithm consists of the following six steps. We use north, south, east, and west for explanation, however each module does not need to know these directions.

Step 1 By M_{Down}, the MRS moves down along a vertical line on a plane $\{(x,y,z)|y=s\}$ for some s.

Step 2 When the MRS reaches the bottom wall, it changes the direction by M_{TurnD}.

Step 3 By M_{Up}, the MRS moves up along the same vertical line as Step 1.

Step 4 If the MRS is adjacent to the bottom of west wall, it moves to the next plane $\{(x,y,z)|y=s-1\}$ by M_{B1}. Then it moves along the bottom wall in the east direction by M_{B2}. When the MRS reaches the east wall, it performs M_{B3} and starts searching the next plane by repeating Step 1. Otherwise, it proceeds to Step 5.

Step 5 When the MRS reaches the top wall, it moves west by one row by M_{TurnU}. Then it repeats Step 1 again.

Step 6 When the MRS reaches southwest corner, it performs M_{Corner} and starts searching on a plane perpendicular to the previous search plane. Thus, a search plane is first perpendicular to the x axis and moves to east, second it is perpendicular to the y axis and moves to north, third it is perpendicular to the x axis and moves to west, and finally, it is perpendicular to the y axis and moves to south.

Depending on its initial state, MRS may start from the middle of the above track.

We omit the description like Table 1 and 2 due to the page limitation.

We briefly address the correctness of the proposed algorithm. The MRS visits all cells on a vertical line $\{(x,y,z)|x=t,y=s\}$ by Steps 1, 2, and 3. Then, it proceeds to the next vertical line by Step 5. By repeating Steps 1, 2, 3, and 5, it visits all cells on plane $y=s$. By Step 4, it starts searching the next plane $y=s-1$. By repeating Steps 1 to 5, it eventually reaches the southwest corner of the bottom wall. It starts searching the vertical line $\{\{(x,y,z)|x=1,y=d-2\}\}$ by Step 6. By repeating Step 1 to 6 four times, the MRS visits all cells of the field.

3.3 Search without a common compass

We show the following theorem by a search algorithm for the MRS of five modules not equipped with a common compass.

▶ **Theorem 3.** *The MRS of five modules not equipped with a common compass can solve a search problem in a finite 3D grid if no pair of modules have an identical observation in an initial configuration.*

We prove Theorem 3 by a search algorithm for the MRS of three modules not equipped with a common compass.

The proposed algorithm considers each plane perpendicular to one of the x, y, and z axes. The choice of the axis depends on the initial configuration of the MRS, and the modules do not need to know the global coordinate system. In the following, without loss of generality, we assume that the MRS considers planes perpendicular to the x axis, i.e., $x=s(y=s$,$z=s$, respectively) for $s=0,1,2,\ldots$. It moves along each vertical line $\{(x,y,z)|x=s,y=t,z=u\}$ or horizontal line $\{(x,y,z)|x=s,y=u,z=t\}$ for $u=0,1,2,\ldots$ on the plane. Figure 17 shows an execution of the algorithm.

The MRS continues to search each plane perpendicular to the x-axis until it reaches the east wall. Then, it changes the search direction from the positive x direction to the negative x direction and it starts to search each plane perpendicular to the x-axis until it reaches the west wall.

The proposed algorithm consists of the following move sequences. We omit the detailed description and figures due to the page limitation.

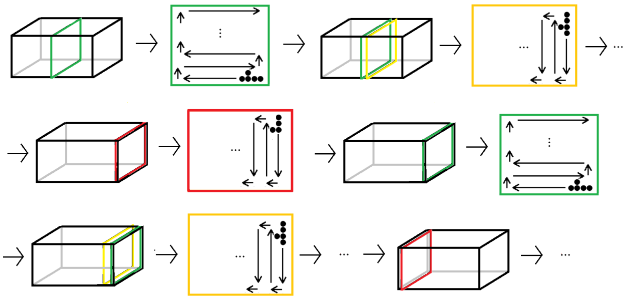

Figure 17 Example of a search with five modular robots.

- Move sequence $M_{Forward}$. By this move sequence, the MRS visits all the cells of the horizontal line $\{(x, y, z)|x = s, y = t\}$.
- Move sequence M_{Back}. By this move sequence, the MRS visits all the cells of the horizontal line $\{(x, y, z)|x = s, y = t\}$.
- Move sequence M_{TurnB}. By this move sequence, the MRS changes its move sequence from $M_{Forward}$ to M_{Back}.
- Move sequence M_{TurnF}. By this move sequence, the MRS changes its move sequence from M_{Back} to $M_{Forward}$.
- Move sequence M_{Edge}. By this move sequence, the MRS changes its move sequence from $M_{Forward}$ to M_{TurnB}.
- Move sequence M_{Corner}. By this move sequence, the MRS changes its move sequence from $M_{Forward}$ to M_{TurnB}.

The proposed algorithm consists of the following six steps. We use down direction for explanation, but each module does not need to know the down direction.

Step 1 By $M_{Forward}$, the MRS moves to the down direction along a vertical line on a plane $\{(x, y, z)|x = s\}$ for some s.
Step 2 When the MRS reaches the bottom wall, it changes the direction to up by M_{TurnB}.
Step 3 By M_{Back}, the MRS moves up along a vertical line followed in Step 1.
Step 4 If the MRS is adjacent to the bottom wall, it moves to plane $\{(x, y, z)|x = s + 1\}$ by M_{TurnF}, and starts searching the new plane by repeating Steps 1,2, and 3.
Step 5 When the MRS reaches the top or bottom wall, it moves to the southern row by M_{Edge}. Then it repeats Steps 1, 2, and 3 again.
Step 6 When the MRS reaches a corner of the field, it perform M_{Corner} and changes the search direction from the positive x direction to the negative x direction.

We omit the description like Table 1 and 2 due to the page limitation.

We briefly address the correctness of the proposed algorithm. The MRS visits all the cells on line $\{(x, y, z)|x = s, y = t\}$ by Steps 1, 2, and 3. Then, it proceeds to the next line by Step 5. By repeating Steps 1, 2, 3, and 5, it visits all cells on plane $x = s$. By Step 4, it starts searching a new plane $x = s + 1$. By repeating Steps 1 to 5, it eventually reaches the corner adjacent to the south wall and the east wall. By Step 6, it starts searching west wall. By repeating Steps 1 to 6 twice, the MRS visits all cells of the field.

You can find demonstration videos of the proposed algorithms in [18].

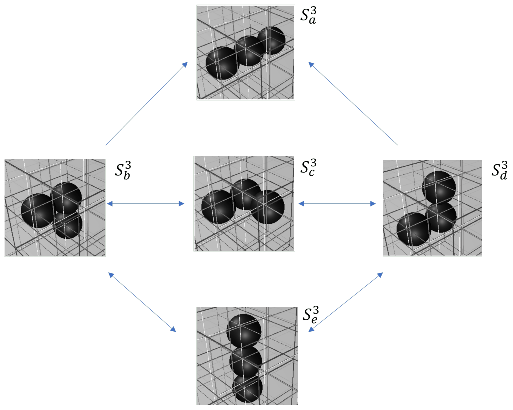

Figure 18 State transition graph for a MRS consisting of 3 modules with common vertical axis.

4 Necessary number of modules

We show that the three algorithms presented in Section 3 use the minimum number of modules for each setting.

▶ **Theorem 4.** *The MRS of less than three modules equipped with a common compass cannot solve the search problem in a finite 3D cubic grid.*

Due to the page limitation, we show a sketch of the proof.

A single module cannot perform any movement because there is no static module during the movement.

Two modules can perform rotations and we can show that the MRS can move straight to one direction by repeating rotations. Thus, when both modules cannot observe any wall (i.e., in the middle area of the field), the MRS moves straight. Assume that the straight movement of the MRS is parallel to x axis. When the field is large enough, the MRS cannot move along some lines parallel to the x axis because the MRS cannot count.

We next show the necessary number of modules equipped with a common vertical axis.

▶ **Theorem 5.** *The MRS of less than four modules equipped with a common vertical axis cannot solve the search problem in a finite 3D cubic grid.*

Proof. In the case of one module, the MRS cannot move because it cannot perform any sliding or rotation.

In the case of two modules, there are two possible states of the MRS. Let S_A be the state, where the two modules form a vertical line, and S_B be the state, where the two modules form a horizontal line. There exists only one horizontal line state because the modules do not agree on x axis or y axis. In S_A, one of the two modules can perform a rotation because they agree on a common vertical axis. Any rotation in S_A results in S_B. In S_B, both modules obtain the same observation if their local coordinate systems are symmetric against their midpoint, and if one of them moves then the other also moves. Thus, the two modules cannot perform any movement. Consequently, the two modules cannot move to any direction.

In the case of three modules, we check possible movements of the MRS by the state transition graph shown in Figure 18. State S_a^3 cannot be transformed to any other configuration because both endpoint modules obtain the same observation. Therefore, it is necessary to move only by S_b^3, S_c^3, S_d^3, and S_e^3. However, no matter which transformation of the S_b^3, S_c^3, S_d^3, and S_e^3, the MRS cannot move in the east, west, south or south direction. Therefore, when there is no wall in the visibility, the MRS cannot move, thus it cannot search the whole field. ◀

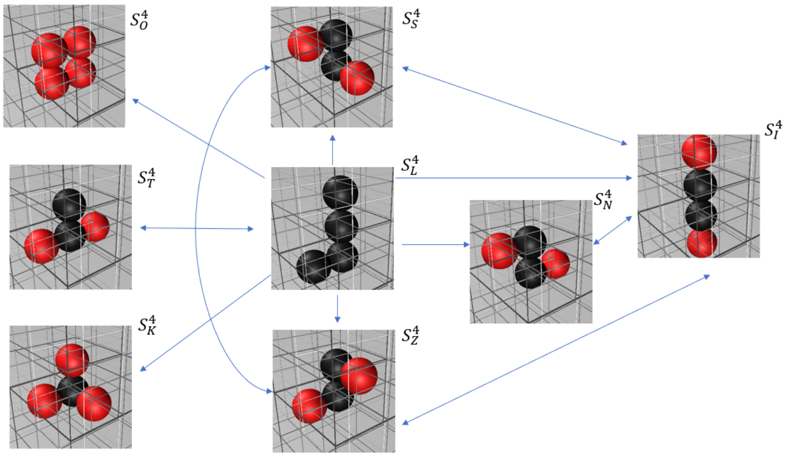

Figure 19 State transition graph for the MRS of 4 modules not equipped with a common compass.

We finally show the necessary number of modules not equipped with a common compass.

▶ **Theorem 6.** *The MRS of less than five modules not equipped with a common compass cannot solve the search problem in a finite 3D cubic grid.*

Due to the page limitation, we show a sketch of the proof.

A single module cannot perform any movement because there is no static module during the movement.

Two modules not equipped with a common compass cannot perform any movement, because if one module moves then the other module also moves.

Three modules have two states, i.e., the "L"-shape and the "I"-shape. In the L-shape, no module can perform a movement because of the symmetry. In the I-shape, two endpoint modules can perform rotations and a new state is an L-shape or I-shape. The MRS does not move after these rotations. Consequently, the MRS cannot move to any direction.

Four modules have eight states. By the state transition graph of the four modules shown in Figure 19, we can show that the MRS cannot move to any direction.

5 Conclusion and future work

In this paper, we considered search by the single MRS in the finite 3D cubic grid. We demonstrated a trade-off between the common knowledge and the necessary and sufficient number of modules for search. We finally note that the proposed algorithms depend on parallel movements, i.e., they are not designed for the centralized scheduler.

Our future goal is a distributed coordination theory for the MRS. First, reconfiguration and locomotion of a single MRS in the 3D cubic grid have not been discussed yet. Second, it is important to consider interaction among multiple MRSs such as rendezvous, collision avoidance, and collective search. Finally, the MRS is expected to solve more complicated tasks by interaction with the environment.

References

1 Abdullah Almethen, Othon Michail, and Igor Potapov. On efficient connectivity-preserving transformations in a grid. In *Proc. of ALGOSENSORS 2020*, pages 76–91, 2020.

2 Abdullah Almethen, Othon Michail, and Igor Potapov. Pushing lines helps: Efficient universal centralised transformations for programmable matter. *Theoretical Computer Science*, 830-831:43–59, 2020.

3 Dana Angluin, Zoë Diamadi James Aspnes, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *Proc. of PODC 2004*, pages 290–299, 2004.

4 Shantanu Das, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Masafumi Yamashita. Autonomous mobile robots with lights. *Theoretical Computer Science*, 609:171–184, 2016.

5 Zahra Derakhshandeh, Shlomi Dolev, Robert Gmyr, Andréa W. Richa, Christian Scheideler, and Thim Strothmann. Brief announcement: amoebot - a new model for programmable matter. In *Proc. of SPAA 2014*, pages 220–222, 2014.

6 Zahra Derakhshandeh, Robert Gmyr, Andréa W. Richa, Christian Scheideler, and Thim Strothmann. An algorithmic framework for shape formation problems in self-organizing particle systems. In *Proc. of NANOCOM 2015*, pages 21:1–21:2, 2015.

7 Keisuke Doi, Yukiko Yamauchi, Shuji Kijima, and Masafumi Yamashita. Exploration of finite 2d square grid by a metamorphic robotic system. In *Proc. of SSS 2018*, pages 96–110, 2018.

8 Adrian Dumitrescu and János Pach. Pushing squares around. *Graphs and Combinatorics*, 22:37–50, 2006.

9 Adrian Dumitrescu, Ichiro Suzuki, and Masafumi Yamashita. Formations for fast locomotion of metamorphic robotic systems. *The International Journal of Robotics Research*, 23(6):583–593, 2004.

10 Adrian Dumitrescu, Ichiro Suzuki, and Masafumi Yamashita. Motion planning for metamorphic systems: feasibility, decidability, and distributed reconfiguration. *IEEE Transactions on Robotics*, 20(3):409–418, 2004.

11 Nao Fujinaga, Yukiko Yamauchi, Hirotaka Ono, Shuji Kijima, and Masafumi Yamashita. Pattern formation by oblivious asynchronous mobile robots. *SIAM Journal on Computing*, 44(3):740–785, 2015.

12 Giuseppe Antonio Di Luna, Paola Flocchini, Nicola Santoro, Giovanni Viglietta, and Yukiko Yamauchi. Shape formation by programmable particles. *Distributed Computing*, 33(1):69–101, 2020.

13 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.

14 Junya Nakamura, Sayaka Kamei, and Yukiko Yamauchi. Evacuation from a finite 2d square grid field by a metamorphic robotic system. In *Proc. of CANDAR 2020*, pages 69–78, 2020.

15 Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

16 Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.

17 Pierre Thalamy, Benoît Piranda, and Julien Bourgeois. Distributed self-reconfiguration using a deterministic autonomous scaffolding structure. In *Proc. of AAMAS 2019*, pages 140–148, 2019.

18 Ryonosuke Yamada. MRS demonstration videos. URL: `http://tcs.inf.kyushu-u.ac.jp/~yamauchi/MRSdemonstrations.html`.