

# Decentralized Gathering of Stochastic, Oblivious Agents on a Grid: A Case Study with 3D M-Blocks

Anıl Özdemir<sup>1,\*</sup>, John W. Romanishin<sup>2,\*</sup>, Roderich Groß<sup>1,2</sup>, and Daniela Rus<sup>2</sup>

**Abstract**—We propose stochastic control policies for gathering a group of embodied agents in a two-dimensional square tile environment. The policies are fully decentralized and can be executed on anonymous, oblivious agents with chirality, but no sense of orientation. The agents require only 4 ternary digits of information. We prove that a group of agents, irrespective of initial positions, will almost surely reach a Pareto optimal configuration in finite time. For one of the control policies, computer simulations show that groups of up to 20 agents consistently reach Pareto optimal configurations, whereas groups of 1000 agents, given the same amount of time, improve the compactness of their configurations on average by 89.20%. The policy also copes well with sensory noise up to a level of 50%. We also present an experimental validation using 6 physical 3D M-Block modules, demonstrating the feasibility of the stochastic control approach in practice.

## I. INTRODUCTION

Getting into physical proximity is often a prerequisite for groups of autonomous robots that are collaborating to accomplish a specific task. The underlying problem, referred to as robot *aggregation* [1], *gathering* [2], or *rendezvous* [3], is not only relevant for groups of loosely coupled robots, but also for the units of modular reconfigurable systems that, by physically assembling with each other, form larger connected entities [4]. In the following we consider the situation that all robots execute the same control policy, and that they are not allowed to exploit any cues from the environment, such as the intensity of ambient light [5], [6].

For gathering in continuous space, some solutions require that each robot determines the relative position of all other robots in its local neighborhood. For example, Ji and Egerstedt [7] present a solution that is guaranteed to solve the gathering problem, provided that the visibility graph corresponding to the robots’ initial spatial distribution is connected. Other solutions require that each robot determines the bearing of all other robots in its local neighborhood [2], again assuming initial connectivity. For robots using a line-of-sight sensor, it was shown that a single bit of information—whether another robot is detected or not—could be sufficient to solve the gathering problem, though only if the sensing range is unlimited [8]. Ozsoyeller *et al.* [9] present a solution guaranteeing that a pair of robots, operating in an environment with polygonal obstacles, is guaranteed to meet almost

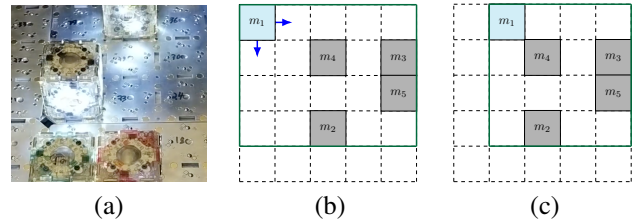


Fig. 1. Decentralized gathering in a 2D square tile environment. (a) Experiments are conducted with the modules of the 3D M-Block reconfigurable robotics platform. Each module has four sensor units, one per side, which report whether other modules are in contact (i.e., physically connected) and/or visible (i.e., their body lights are perceived). (b)–(c) Illustration of the naive stochastic control policy (Algorithm 1). An agent can move into any empty, adjacent cell if another agent is perceived in the corresponding direction (e.g., see blue arrows for agent  $m_1$ ), or remain in its current cell. The agent chooses uniformly random among all eligible actions. Parts (b) and (c) show the situation immediately before and after agent  $m_1$ ’s move. In this instance, the group’s spatial configuration became more compact—the dimensions of the corresponding bounding box reduced from  $5 \times 4$  to  $4 \times 4$  (see green frames). The agents will almost surely reach in finite time a Pareto optimal configuration, that is, a configuration with bounding box dimensions  $2 \times 3$  or  $3 \times 2$ .

surely. The solution involves repetitively tossing a coin to decide whether to rest in place, or move in a way that covers the environment. The strategy is extended to more than two robots, provided they can communicate. Barel *et al.* [10] propose a probabilistic algorithm for gathering agents with 1-bit, unlimited range sensors. At every time step, each agent assumes a random orientation, and then moves forward if no other agent is present in the half-plane behind it, and rests, otherwise. The correctness of the algorithm is proven under the assumptions that the agents act synchronously, can jump instantaneously from one pose to another, and do not have physical bodies. Moreover, to avoid deadlock situations, the binary sensor is shown to require a half-disk blind region.

For gathering in discrete space, Cord-Landwehr *et al.* [11] and Fischer *et al.* [12] present solutions for robots with constant memory and no memory (oblivious), respectively. The solutions are guaranteed to converge in linear and quadratic time, respectively. They require each robot to determine the relative position of all other robots in its local neighborhood, comprising more than 100 cells, and a visibility graph that is initially connected. The robots are not embodied; where multiple robots occupy the same cell, all but one are removed.

In this paper, we propose stochastic control policies for gathering a group of embodied agents in a two-dimensional (2D) square tile environment, as illustrated by Figure 1. The policies are fully decentralized and can be executed on

<sup>1</sup>A. Özdemir and R. Groß are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, UK, {a.ozdemir|r.gross@sheffield.ac.uk}

<sup>2</sup>J. W. Romanishin and D. Rus are, and R. Groß has been, with the Computer Science and Artificial Intelligence Lab, MIT, Cambridge, MA, 02139, {johnrom|rus@mit.edu}

\*These authors contributed equally to this work.

anonymous, oblivious agents with chirality, but no sense of orientation. Unlike previous solutions to the gathering problem with such restricted agents, our policies are not limited to specific initial positions, take the agent’s embodiment into account, and require only four trits of sensory information, though the latter comes at the expense of unlimited-range sensing.

The stochastic control policies can be viewed as an extension of the “computation-free” swarming concept [8], [13]. In particular, each agent reactively chooses its action based on a few bits of sensory information, though in the present work the choice is stochastic, rather than deterministic.

The paper is organized as follows. Section II describes the gathering problem. Section III presents a naive, stochastic control policy and proves its correctness. Section IV presents a refined variant of the naive control policy. Section V evaluates both control policies by simulation. Section VI presents an experimental validation using the physical modular robotics platform 3D M-Blocks [14]. Section VII concludes the paper.

## II. PROBLEM FORMULATION

### A. Environment and Robot Model

Consider an unbounded, obstacle-free 2D square tile environment, containing  $n$  mobile agents. The agents are anonymous, that is, indistinguishable from each other, fully autonomous, and execute an identical controller. Each agent occupies a tile<sup>1</sup>, has no orientation, but can distinguish between clockwise and counter-clockwise (chirality).

Each agent has four sensor units, one per side. Each unit provides a tuple of binary values,  $s = (c, v)$ . The first value indicates whether another agent is in physical *contact* with the sensor unit;  $c$  is true if another agent resides on the corresponding adjacent cell, and false, otherwise. The second value indicates if any other agent is *visible* from the sensor unit;  $v$  is true if at least one agent resides within the half-plane next to the unit, and false, otherwise. Formally,

$$c = \text{true if } \exists j : (x_j^{\text{rel}}, y_j^{\text{rel}}) = (1, 0), \quad (1)$$

$$v = \text{true if } \exists j : x_j^{\text{rel}} > 0, \quad (2)$$

where  $(x_j^{\text{rel}}, y_j^{\text{rel}}) \in \mathbb{Z}^2$  denotes the position of agent  $j$  in the reference frame that is (i) local to the sensing agent, and (ii) has its  $x$ -axis parallel to the sensor unit’s sensing direction.<sup>2</sup> Note that  $c = \text{true}$  implies  $v = \text{true}$ . In other words, the sensor unit provides a ternary digit (i.e., trit) of information,  $s \in \{(\text{false}, \text{false}), (\text{false}, \text{true}), (\text{true}, \text{true})\}$ .

Time is assumed to be discrete. In each round, every agent executes one action; the update order of agents is fixed.<sup>3</sup>

An agent can choose to remain in its current cell (action  $a_0$ ), or move into any adjacent cell (actions  $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ ), provided the latter is not occupied. The

sensor data,  $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ , and actions  $\mathcal{A}$  are provided in a counter-clockwise order. As the agents have no orientation, the specific starting elements are irrelevant, as long as consistent (e.g.,  $s_3$  and  $a_3$ ).

### B. Objective

A configuration of a group of  $n$  agents defines their position in space. Formally,  $\mathcal{C} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ ,  $\forall i \neq j : (x_i \neq x_j) \vee (y_i \neq y_j)$ , where  $(x_j, y_j) \in \mathbb{Z}^2$  denotes the position of agent  $j$  in the global reference frame.

Given a configuration  $\mathcal{C}$ , let  $b = (b_x, b_y)$  denote the dimensions of the corresponding bounding box. Formally,

$$b_x = 1 + \max_{i,j} |x_i - x_j|, \quad (3)$$

$$b_y = 1 + \max_{i,j} |y_i - y_j|.$$

Consider two configurations,  $\mathcal{C}$  and  $\bar{\mathcal{C}}$ , of  $n$  agents, with bounding box dimensions  $b$  and  $\bar{b}$ , respectively. Configuration  $\mathcal{C}$  is said to be *preferred* to configuration  $\bar{\mathcal{C}}$ , denoted by  $\bar{\mathcal{C}} \prec \mathcal{C}$ , if  $(b_x < \bar{b}_x) \wedge (b_y \leq \bar{b}_y)$  or  $(b_x \leq \bar{b}_x) \wedge (b_y < \bar{b}_y)$ . Configuration  $\mathcal{C}$  is said to be *Pareto optimal*, if there exists no other configuration of  $n$  agents that is preferred to  $\mathcal{C}$ .

The agents start from arbitrary cells. Their objective is to collectively reach, and remain indefinitely, in a Pareto optimal configuration.

### C. Mathematical Analysis

**Lemma 1.** *A configuration of  $n$  agents contained in a bounding box of dimensions  $(b_x, b_y)$  is Pareto optimal, if and only if  $b_x b_y - n < \min\{b_x, b_y\}$ .*

*Proof.* First, we consider the case that a Pareto optimal configuration,  $\mathcal{C}$ , is given. Without loss of generality, we assume  $b_y \leq b_x$ . Let  $h = b_x b_y - n$ , that is,  $h$  is the total number of empty cells within the bounding box. If  $h \geq \min\{b_x, b_y\} = b_y$ , then  $b_x > 1$ . Let  $h_1 \geq 0$  and  $h_r > 0$  denote the number of empty cells within the first column and the remaining columns of the bounding box, respectively. We have  $h = h_1 + h_r$ . We can remove the  $b_y - h_1 > 0$  agents from the first column and insert them on some of the  $h_r = h - h_1 \geq b_y - h_1$  empty cells in the other columns. This would produce a configuration  $\bar{\mathcal{C}}$  that has at least one fewer column and at most the same number of rows, that is,  $\bar{\mathcal{C}} \prec \mathcal{C}$ . This however contradicts our assumption that  $\mathcal{C}$  is Pareto optimal. Consequently,  $h = b_x b_y - n < \min\{b_x, b_y\}$ . Second, we consider the case of a configuration with  $b_x b_y - n < \min\{b_x, b_y\}$ . The number of empty cells within the bounding box is  $h = b_x b_y - n < \min\{b_x, b_y\}$ . In other words, neither of the dimensions of the bounding box can be reduced, without increasing the respective other dimension. Therefore, the configuration is Pareto optimal.  $\square$

## III. NAIVE STOCHASTIC CONTROL POLICY

Algorithm 1 describes the naive stochastic control policy, hereafter referred to as  $\mathcal{P}_N$ . In each control cycle, the agent chooses uniformly random from the following set of actions:  $a_0 \cup \{a_i \in \mathcal{A} \mid i \in \{1, 2, 3, 4\} \wedge (\neg c_i \wedge v_i)\}$ . In other words, the agent can rest in place, but may move in up to four directions

<sup>1</sup>A tile can not be occupied by multiple agents.

<sup>2</sup>As the agent has no orientation, each of the four sensor units has its own local reference frame.

<sup>3</sup>Our theoretical analysis is also valid if the order changes randomly.

---

**Algorithm 1** Naive Stochastic Control Policy,  $\mathcal{P}_N$ 


---

```

1: while true do
2:    $\mathcal{A}_e \leftarrow \{a_0\}$             $\triangleright$  initialize set of eligible actions
3:   for all  $i \in \{1, 2, 3, 4\}$  do
4:     update  $c_i$                   $\triangleright$  probe Boolean contact sensor  $i$ 
5:     update  $v_i$                   $\triangleright$  probe Boolean visibility sensor  $i$ 
6:     if  $\neg c_i \wedge v_i$  then
7:        $\mathcal{A}_e \leftarrow \mathcal{A}_e \cup \{a_i\}$     $\triangleright$  add eligible action  $\{a_i\}$ 
8:     end if
9:   end for
10:   $a \leftarrow$  select uniformly random from  $\mathcal{A}_e$ 
11:  execute  $a$ 
12:  wait  $\delta$  units of time
13: end while

```

---

(see Figure 1). It is prevented from moving into a direction that is blocked by an adjacent agent (i.e.,  $c_i = \text{true}$ ), or in which no other agent is seen (i.e.,  $v_i = \text{false}$ ). The control cycle is here assumed to have some finite length  $\delta$ . Note that Algorithm 1 is fully reactive, as the agent does not store any information from the previous cycle.

#### A. Mathematical Analysis

**Lemma 2.** Consider  $n$  agents using policy  $\mathcal{P}_N$ . Let  $\mathcal{C}[k]$  and  $\mathcal{C}[k+1]$  denote the configurations at time steps  $k$  and  $k+1$ , respectively, which is immediately before and after one of the agents was considered. Then,  $b_x[k+1] \leq b_x[k]$  and  $b_y[k+1] \leq b_y[k]$ .

*Proof.* At time step  $k$  only one agent, say agent  $j_1$ , was considered. All other agents will not have moved, that is,  $\forall j_2 \neq j_1 : x_{j_2}[k+1] = x_{j_2}[k]$  and  $y_{j_2}[k+1] = y_{j_2}[k]$ . The  $x$ -coordinate of the “leftmost” agent at time  $k$  is given as  $x_{\text{left}} = \min_{j_2} \{x_{j_2}[k]\}$ . If agent  $j_1$  was at the left boundary ( $x_{j_1}[k] = x_{\text{left}}$ ), no agent would have been visible towards the “left” ( $v = \text{false}$ ), which would prevent the agent from moving in that direction. Otherwise ( $x_{j_1}[k] > x_{\text{left}}$ ), agent  $j_1$  may have moved, but at most by 1 cell. In both cases, we have  $x_{j_1}[k+1] \geq x_{\text{left}}$ . The same argument can be used for the lower, right and upper boundaries. From this, it follows that  $b_x[k+1] \leq b_x[k]$  and  $b_y[k+1] \leq b_y[k]$ .  $\square$

**Corollary 1.** Consider  $n$  agents using policy  $\mathcal{P}_N$ . Let  $\mathcal{C}[k]$  denote the configuration at time step  $k$ . Then,  $\forall l > k : b_x[l] \leq b_x[k]$  and  $b_y[l] \leq b_y[k]$ .

**Theorem 1.** Using policy  $\mathcal{P}_N$ ,  $n$  agents almost surely reach a Pareto optimal configuration in finite time.

*Proof.* Let for all  $k \geq 0$ ,  $h[k] = b_x[k]b_y[k] - n$  denote the number empty cells within the bounding box at time step  $k$ . We prove the theorem by induction.

*Base case:*  $h[k] = 0$ . As  $h[k] = b_x[k]b_y[k] - n = 0 < \min\{b_x, b_y\}$ , from Lemma 1 it follows that the configuration is Pareto optimal. From Corollary 1 it follows that the configuration remains Pareto optimal indefinitely.

*Induction step:*  $h[k] > 0$ . Without loss of generality, we assume  $b_y[k] \leq b_x[k]$ . Moreover,  $b_x[k] > 1$ , as oth-

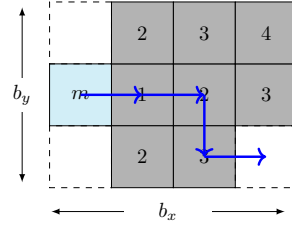


Fig. 2. Example configuration of 9 agents with a  $4 \times 3$  bounding box. The configuration is not Pareto optimal, as the same number of agents could be contained in a  $3 \times 3$  bounding box. The numbers indicate the Manhattan distance between the corresponding cell and the reference agent,  $m$ , of the first column. The blue arcs illustrate one of the shortest paths from the reference agent to the empty cell in the last column. If the four agents on this path choose to move in the indicated direction, whereas all other agents choose not to move, the empty cell is pushed into the first column, causing the new configuration to be Pareto optimal.

erwise,  $h[k] = 0$ . If  $h[k] = b_x[k]b_y[k] - n < b_y[k] = \min\{b_x[k], b_y[k]\}$ , then it follows from Lemma 1 and Corollary 1 that the configuration is Pareto optimal and remains so indefinitely. For the case  $h[k] \geq b_y[k]$ , as in the proof of Lemma 1, we consider that the agents of the leftmost column all relocate to the other columns (see Figure 2). We obtain a positive lower bound for the probability for this to happen in constant time. We assume that every round one agent chooses to move, while all others choose not to move. The probability for this to happen in a given round is at least  $\epsilon^n$ , where  $\epsilon = \frac{1}{5}$  is a lower bound for the probability for any eligible action to be chosen (note that actions are chosen uniformly random). Let us consider the shortest path from an arbitrary agent of the leftmost column to an empty cell in one of the other columns (see Figure 2). The length of path candidates are determined by the Manhattan distance, and thus reflecting how the agents may move. Multiple shortest paths may exist, but in any case only the last cell of a path is an empty cell. At any round let only the agent that is closest to the empty cell (but part of the remaining path) move. At most  $d$  rounds are required for the empty cell to reach the leftmost column, where  $d$  is the length of the shortest path. As  $d$  is bounded by  $b_x[k] + b_y[k] - 2$ , the probability for the agent relocation to have occurred after  $b_x[k] + b_y[k] - 2$  rounds is at least  $\epsilon^{(b_x[k]+b_y[k]-2)n}$ . As there could be up to  $b_y[k]$  agents in the leftmost column, the probability to reach the preferred configuration after  $U = b_y[k](b_x[k] + b_y[k] - 2)$  rounds is at least  $p = \epsilon^{b_y[k](b_x[k]+b_y[k]-2)n}$ . From Lemma 2, it follows that the bounding box dimensions,  $b_x$  and  $b_y$ , are monotonically decreasing with time,  $k$ . In other words, our lower bound,  $p$ , monotonically increases with time,  $k$ , and the number of rounds,  $U$ , required for an improvement to occur with at least probability  $p$ , monotonically decreases with time,  $k$ . If an improvement occurred, the new configuration would have at least  $b_y \geq 1$  fewer empty cells, resulting in  $h[k+Un] \leq h[k] - b_y[k] \leq h[k] - 1$ . The probability that an improved (preferred) configuration is found within  $\tau U$  rounds is at least  $p_\tau = 1 - (1-p)^\tau$ . We have  $\lim_{\tau \rightarrow \infty} p_\tau = 1$ . In other words, a preferred configuration is found almost surely in finite time, reducing  $h$  by at least 1. As  $\forall k : h[k] \geq$

0, only a finite number of improvements are possible. A Pareto optimal configuration is hence obtained almost surely in finite time.  $\square$

#### IV. OPTIMIZED STOCHASTIC CONTROL POLICY

In the previous section, we showed that a group of agents almost surely reach a Pareto optimal configuration in finite time when using the naive stochastic control policy,  $\mathcal{P}_N$ . The policy determines the set of eligible actions and then chooses uniformly random from this set.

In this section, we consider an alternative stochastic control policy,  $\mathcal{P}_O$ , which is not restricted to using uniform distributions, but rather takes into account an agent's *context*. An agent can be in any of 18 contexts (see Figure 3), which is fully defined by the agent's eight sensor reading values  $(c_1, c_2, c_3, c_4, v_1, v_2, v_3, v_4)$ . Depending on the context, an agent can choose between 1 and 5 actions (note that an agent can always choose the option to rest, that is,  $a_0$ ). An agent in context  $A$  can either remain in its current position (action  $a_0$ ) or move into any direction  $(a_1, a_2, a_3, a_4)$ . As the agent has no orientation, it has to choose either direction with equal probability. An agent in context  $B$  has four possible options to choose from, and, due to chirality, for each option a dedicated probability can be chosen. In the following, we optimize the probabilities of choosing the eligible actions for each specific context.

We choose  $\epsilon$  to be the lowest probability of the  $\mathcal{P}_O$  controller. If  $\epsilon \neq 0$ , then the proof for Theorem 1 directly extends to  $\mathcal{P}_O$ .

##### A. Representation of Candidate Solutions

A candidate solution is represented by 27 real-valued parameters in range  $[0, 1]$ , which, following normalization, determine the motion probabilities. For example, for context  $A$ , all five actions are possible. However, a single parameter is sufficient, as the probabilities,  $p_1, p_2, p_3$ , and  $p_4$ , for choosing actions  $a_1, a_2, a_3$ , and  $a_4$  must be identical (due to the lack of orientation). Moreover, the probability of choosing  $a_0$  must be  $1 - p_1 - p_2 - p_3 - p_4$ .

##### B. Evolutionary Algorithm

We employ an evolutionary algorithm for the optimization process, namely Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [15]. CMA-ES is a derivation-free, black-box optimization method. It starts with a random population of  $\lambda$  candidate solutions, and uses a fitness function to select promising solutions for producing the subsequent generations.

We define the *compactness* of the configuration in round  $r$  as the number of empty cells in the corresponding bounding box, if the configuration is not Pareto optimal, and 0, otherwise. Formally, compactness  $H[r]$  is given as,

$$H[r] = \begin{cases} h[r], & \text{if } h[r] \geq \min\{b_x[r], b_y[r]\}; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

In each generation, every candidate solution is tested against the same set of  $T = 20$  starting configurations. A new

set of configurations is produced at the beginning of every generation. For each configuration, the number of agents is chosen as  $n = 2 + m$ , where  $m$  is generated randomly using the exponential cumulative distribution function given by,

$$F(m; \lambda) = \begin{cases} 1 - e^{-\lambda m}, & \text{if } m \geq 0; \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where  $\frac{1}{\lambda} = 6$  represents the expected value,  $E[m]$ . Hence, the expected number of agents is  $E[n] = 2 + E[m] = 8$ . The agents are all placed in random positions within a grid size of  $L = 2 \lceil \sqrt{n} \rceil$  and are assigned a randomly generated *fixed* update order.

The *fitness* function to be minimized by the evolutionary algorithm is,

$$f = \sum_{r=1}^R rH[r], \quad (6)$$

where  $R = 100$  is the maximum number of rounds. The performance measure  $H[r]$  is multiplied by round number  $r$  to promote faster solutions.

##### C. Controller Selection

We conducted 100 evolutionary runs with a population size of  $\lambda = 30$ . Each run was terminated after 3000 generations. Figure 4 shows the fitness of the highest-rated candidate solution per run.

To select the *best* solution, we considered the solutions that exhibited the lowest fitness value in the final generation of each run. Each of these 100 solutions was post-evaluated on  $T = 200$  random configurations using a grid size of  $L = 100$  and  $R = 10000$  rounds. In the following, we refer to the solution that exhibited the best mean performance as the optimized stochastic control policy,  $\mathcal{P}_O$ .<sup>4</sup>

## V. SIMULATION STUDIES

In this section, we analyze the performance of the control policies,  $\mathcal{P}_N$  and  $\mathcal{P}_O$ , through computer simulations using measure  $H[r]$ , as defined in Eq. (4).

##### A. Scalability Study

We investigate the scalability of the control policies. We consider a 2D square tile environment of size  $L = 100$  containing  $n \in \mathcal{N} = \{2, 5, 10, 20, 50, 100, 200, 500, 1000\}$  agents. In the beginning of a simulation trial, the agents are uniformly randomly placed. For each  $n \in \mathcal{N}$  and each control policy ( $\mathcal{P}_N$  and  $\mathcal{P}_O$ ), 100 trials of  $R = 10000$  rounds are performed.

Figure 5 reports the average performance over the number of rounds. Groups of  $n \in \{2, 5, 10\}$  agents consistently reached a Pareto optimal configuration using either of the control policies. For these group sizes, the optimized controller,  $\mathcal{P}_O$ , took on average 39.50%, 26.97% and 30.07% less time to reach a Pareto optimal configuration. For larger group sizes, it became increasingly unlikely for the agents to reach a Pareto optimal configuration in the provided

<sup>4</sup>The parameters can be found in the online supplementary material [16].

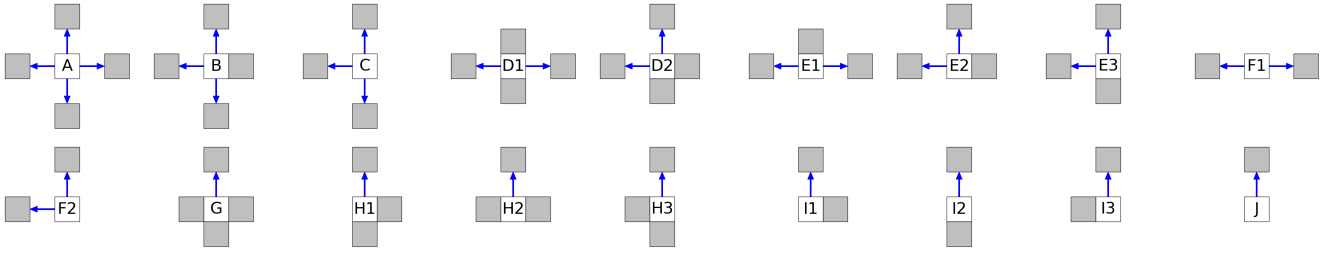


Fig. 3. Overview of the 18 unique contexts in which the focal agent (white cell) can move. A gray agent represents an agent on an adjacent cell, that is, an agent in physical contact ( $c = \text{true}$ ), or an agent that is visible ( $v = \text{true}$ , for at least one of the focal agent’s sides). Arrows indicate possible directions of movement. In addition, an agent may remain in its current position. An agent can also be in one of eight contexts (not shown), where the only possible action is to remain in its current position (action  $a_0$ ). Contexts  $A$ ,  $D_1$ , and  $F_1$  are rotation symmetric; the modules, which have chirality but no sense of orientation, have to choose every direction of movement with equal probability.

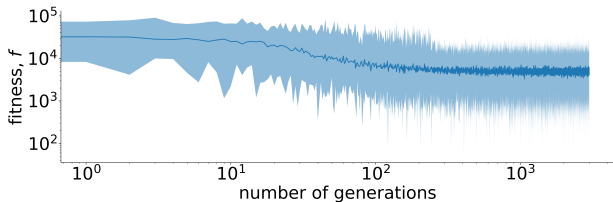


Fig. 4. Fitness dynamics showing the performance of 100 evolutionary runs over 3000 generations. The blue envelope represents the minimum and maximum average fitness values of  $\lambda = 30$  solutions for each run, and the blue line represents the median fitness values.

time period. However, for groups of  $n = 1000$  agents, the compactness,  $H[r]$ , still improved during the trial, on average by 9.26% and 89.20% for  $\mathcal{P}_N$  and  $\mathcal{P}_O$ , respectively.

### B. Sensory Noise Study

Our analysis in Section III assumed the absence of sensory noise. As a consequence, it could be shown that the modules’ bounding box dimensions were monotonically decreasing with time. We now investigate the effect of sensory noise on the performance of both control policies. As mentioned in Section II-A, each of the module’s four sensor units provides one ternary digit of information:  $s \in \{(\text{false}, \text{false}), (\text{false}, \text{true}), (\text{true}, \text{true})\}$ . In the following, we assume that each sensor unit reports a uniformly randomly chosen ternary digit with probability  $p_n$ , and reports the original reading value otherwise. We increase the noise-level from 0% to 100% by 10% increments. For each level of noise and control policy, 100 trials are performed with  $n = 100$  agents in a  $100 \times 100$  grid environment. When a module decides to move onto an occupied cell, no action is taken for the corresponding round. Each trial is run for a constant duration of  $R = 10000$  rounds.

Figure 6 shows the performance measure  $H[r]$  for all levels of noise and both policies. For a 10% noise-level, the compactness,  $H[r]$ , improved during the trial, on average by 93.68% and 99.63% for  $\mathcal{P}_N$  and  $\mathcal{P}_O$ , respectively. The system improved its compactness by at least 90% for any noise level up to 20% and 50% for  $\mathcal{P}_N$  and  $\mathcal{P}_O$ , respectively. For a noise-level of 100%, implying purely random sensor readings, the system diverged.

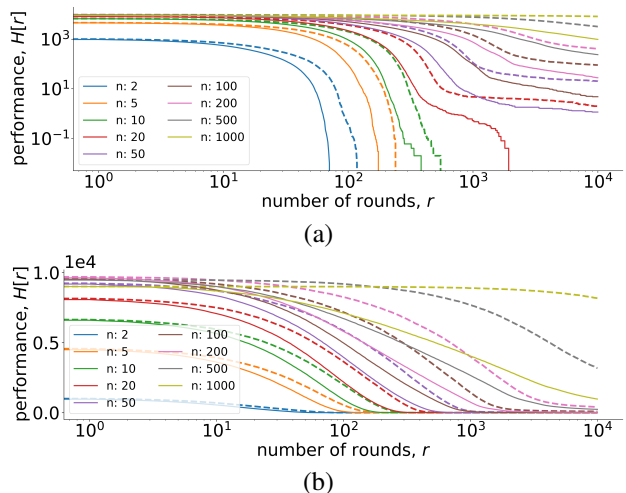


Fig. 5. Results of the scalability study with up to 1000 agents in a  $100 \times 100$  environment. Dashed and solid lines show, respectively, the performance using the naive ( $\mathcal{P}_N$ ) and optimized ( $\mathcal{P}_O$ ) control policies. Each line represents the average, across 100 trials, of  $H[r]$ , which is 0 for Pareto optimal configurations, and otherwise equals the number of empty cells in the bounding box. (a) and (b) plot the compactness using logarithmic and linear axes, respectively.

## VI. EXPERIMENTS

The 3D M-Blocks are  $\sim 50$  mm side length cubic modular robots which move on a cubic lattice using pulses of angular momentum from an internal reaction wheel. Each module includes a main processor, a set of IMUs, and circuit boards on each face which can (i) turn on several white LEDs (ii) identify the presence of directly adjacent modules, and (iii) measure ambient light in the half-plane of the face.

In the experiments, we evaluate the performance of the naive stochastic control policy. We specifically designed a  $6 \times 6$  testing environment, using the modular components shown in Figure 7. All experiments were conducted as follows: (i) six modules are placed at computer-generated random configurations on a  $6 \times 6$  grid; (ii) each experiment runs for 5 minutes; (iii) if a module disconnects from the grid its position is counted as the closest position to the module’s center; (iv) in contrast to the simulations, modules that are connected to more than one neighbor do not attempt to move. This limitation was added as the torque required to



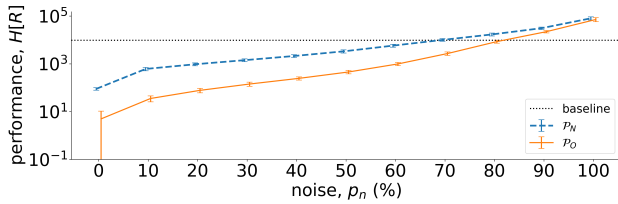


Fig. 6. Impact of sensory noise on the performance at the end of 100 simulation trials ( $H[R]$ , defined in Eq. (4), in logarithmic scale). With probability  $p_n$ , each sensor unit provides a purely random reading value. Dashed and solid lines represent, respectively, the naive ( $\mathcal{P}_N$ ) and optimized ( $\mathcal{P}_O$ ) control policies. The error bars show the standard deviation. The dotted line represents a baseline which corresponds to no movement ( $H[0]$ ).

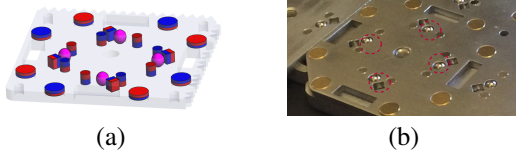


Fig. 7. Modular testing environment used in the experiments: (a) design, and (b) implementation of the lattice cells. Each cell includes 8 small magnets arrayed in a way matching that of the 3D M-Blocks connectors, as well as magnets at the boundary to serve as a temporary hinge when moving to adjacent lattice positions (which proved more effective than the rotating magnets described in [17]). Metal spheres (*purple circles*) are embedded which align with holes in the face of each module to attempt to minimize alignment errors.

move a module, when connected to multiple other modules, could not be reliably generated by the current version of the hardware; (v) no attempt was made to synchronize the modules’ movements. All trials were recorded using a camera for post-analysis.

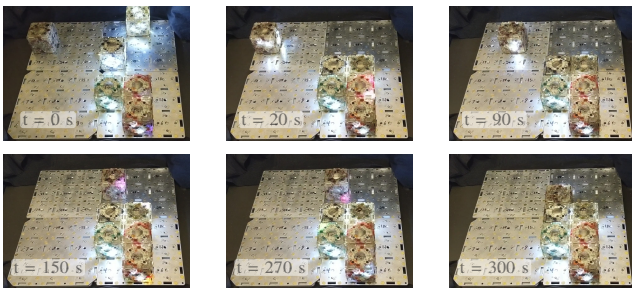


Fig. 8. In this experiment, six modules are placed at uniformly random positions on the  $6 \times 6$  testing environment. They execute the naive stochastic control policy ( $\mathcal{P}_N$ ). During the trial, the compactness of the configuration improved from  $H[0] = 24$  to  $H[R] = 2$ , yielding a 92% reduction of area, which is the largest reduction of area among the trials conducted.

Figure 8 shows a sequence of snapshots taken from one of the trials. Table I presents the results of 10 experimental trials, showing how the initial compactness,  $H[0]$ , reduced to the final compactness,  $H[R]$ . Successful moves are defined as moves where an agent begins in one lattice position and ends in another. Disconnected modules refer to modules which end up in non-lattice positions, including those that leave the grid entirely.

All but two experiments resulted in more compact configurations than those they had begun with. The experiments

TABLE I  
EXPERIMENTAL RESULTS USING SIX 3D M-BLOCK MODULAR ROBOTS.

|                             | Naive ( $\mathcal{P}_N$ ) |
|-----------------------------|---------------------------|
| Experiments conducted       | 10                        |
| Initial compactness, $H[0]$ | $22.2 \pm 6.4$            |
| Final compactness, $H[R]$   | $14.6 \pm 7.4$            |
| Area reduction percentage   | $33\% \pm 30\%$           |
| Modules disconnected        | $1.1 \pm 0.9$             |
| Successful moves            | $6.2 \pm 2.2$             |

where this was not the case involved modules which moved incorrectly and left the lattice. While the experiments do validate the concept of the algorithm, the 3D M-Block hardware presents several practical limitations which diverges from the simulation results. Lessons learned include:

- A modules may experience difficulties moving when there is a module in a cell diagonally across from it due to the edges “catching” each other.
- The current hardware system does not have closed loop control over the actuation torque and manufacturing tolerances lead to a certain percentage of moves to fail—either by moving with too much power and disconnecting from the grid, or not being able to move.
- The face LEDs were not designed for long distance illumination and only reach roughly 3 grid cells (given an ambient light level that is typical for an office environment).

The algorithms are challenging to implement on modular robotic systems due to need to constrain the motion on a regular lattice. However, insights from these experiments suggest several hardware and algorithmic improvements which could help guide future work aggregating grid based modular robots.

## VII. CONCLUSIONS

This paper presented two control polices for gathering oblivious, embodied agents in a 2D square tile environment. The agents used binary sensors that indicate whether other modules are physically in contact or otherwise visible. The agents were proven to assume a Pareto optimal spatial arrangement almost surely in finite time. Simulations examined the performance for different number of agents, or in the presence of sensory noise. Experiments, conducted with six modules of 3D M-Blocks, a self-reconfigurable robotic system, showed the overall feasibility of the concept, and provided insights into possible improvements that could increase the effectiveness of the control policies in practical situations. Future work directions include testing the optimized stochastic control policy on 3D M-Blocks, upgrading the capabilities of the physical modules in regards to movement precision and light projection, or studying more complex tasks, such as assembling 3D shapes.

## ACKNOWLEDGMENT

This work is supported by Amazon Robotics, the NSF through grants 1240383 and 1138967 and the NDSEG fellowship.

## REFERENCES

- [1] N. Correll and A. Martinoli, "Modeling and designing self-organized aggregation in a swarm of miniature robots," *The International Journal of Robotics Research*, vol. 30, no. 5, pp. 615–626, 2011.
- [2] N. Gordon, I. A. Wagner, and A. M. Bruckstein, "Gathering multiple robotic a(gen)t(s) with limited sensing capabilities," in *Ant Colony Optimization and Swarm Intelligence*. Springer, 2004, pp. 142–153.
- [3] S. Alpern, "The rendezvous search problem," *SIAM Journal on Control and Optimization*, vol. 33, no. 3, pp. 673–683, 1995.
- [4] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.
- [5] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim, "Get in touch: cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, pp. 133–155, 2009.
- [6] S. Claiici, J. W. Romanishin, J. I. Lipton, S. Bonardi, K. W. Gilpin, and D. Rus, "Distributed aggregation for modular robots in the pivoting cube model," in *2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 1489–1496.
- [7] M. Ji and M. Egerstedt, "Distributed coordination control of multi-agent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
- [8] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß, "Self-organized aggregation without computation," *The International Journal of Robotics Research*, vol. 33, no. 8, pp. 1145–1161, 2014.
- [9] D. Ozsoyeller, A. Beveridge, and V. Isler, "Rendezvous in planar environments with obstacles and unknown initial distance," *Artificial Intelligence*, vol. 273, pp. 19–36, 2019.
- [10] A. Barel, R. Manor, and A. M. Bruckstein, "Probabilistic gathering of agents with simple sensors," Technion – Israel Institute of Technology, Tech. Rep. CIS-2017-04, 2017.
- [11] A. Cord-Landwehr, M. Fischer, D. Jung, and F. Meyer auf der Heide, "Asymptotically optimal gathering on a grid," in *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*. ACM, 2016, pp. 301–312.
- [12] M. Fischer, D. Jung, and F. Meyer auf der Heide, "Gathering anonymous, oblivious robots on a grid," in *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*. Springer, 2017, pp. 168–181.
- [13] T. Wareham and A. Vardy, "Putting it together: The computational complexity of designing robot controllers and environments for distributed construction," *Swarm Intelligence*, vol. 12, no. 2, pp. 111–128, 2018.
- [14] J. W. Romanishin, K. Gilpin, S. Claiici, and D. Rus, "3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 1925–1932.
- [15] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma on Github," 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.2559634>
- [16] A. Özdemir, J. W. Romanishin, R. Groß, and D. Rus, "Online supplementary material," 2019. [Online]. Available: <https://doi.org/10.6084/m9.figshare.8527148>
- [17] J. W. Romanishin, K. Gilpin, and D. Rus, "M-Blocks: Momentum-driven, Magnetic Modular Robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4288–4295.