
Proceedings Paper:
Bray, E. and Gross, R. (2021) Distributed self-assembly of cantilevers by force-aware robots. In: Proceedings of 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). 2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 04-05 Nov 2021, Cambridge, UK. IEEE, pp. 110-118. ISBN: 9781665429276.

https://doi.org/10.1109/MRS50823.2021.9620697

Distributed Self-Assembly of Cantilevers by Force-Aware Robots

Edward Bray and Roderich Groß

***Abstract*— Inspired by how ants construct bridges out of their bodies, this paper investigates how a swarm of autonomous robotic agents could self-assemble into cantilevers as a first step towards constructing bridges. Two distributed self-assembly algorithms are presented that consider local force information to ensure links between agents do not break: one in which agents move one at a time, and another where multiple agents move simultaneously. The algorithms are tested in simulation for a variety of allowable link strengths, and are verified to be able to construct cantilevers of a near-optimum length. A slight decrease in cantilever length is observed when multiple agents move concurrently, but construction is completed significantly faster. Prototype hardware to measure the forces in links between real agents is also presented, demonstrating how the concept could be applied to the real world.**

I. Introduction

Swarms of robots have been proposed as a possible solution to a wide range of problems, due to the advantages they offer in completing tasks that encompass a large physical space in a robust manner [1]. A closely related field to swarm robotics is modular robotics, where groups of autonomous robots are able to assemble their bodies into a synergistic collective [2]. The requirement for large numbers of robots to comprise an effective swarm or modular robotic system means their design must be cost-effective, which in turn leads to most systems comprising physically small agents with bodylengths in the order of centimeters, such as Kilobots [3], SMORES [4], 3D M-Blocks [5], or HyMod [6]; at such scales, obstacles that would be trivially overcome by large robots can become insurmountable.

Research into groups of robots is often inspired by the behaviour of insects [7]. To enable ants to thrive in a world of obstacles many times greater than their body size, they have evolved to assemble their bodies without any external overseer (referred to as *self-assembly*) into a variety of temporary structures, such as rafts [8], bivouacs [9], and towers [10]: of particular interest to this paper is how ants construct bridges [11]–[13]. Members of the colony self-assemble into these bridges, which other ants can travel over; when the bridge is no longer necessary it is dismantled and the insects that constituted it continue with other tasks.

Inspired by the behaviour of ants, this paper considers how a group of robots could self-assemble into cantilevers, as shown in Fig. 1a, as a first step towards constructing bridges. This would allow the collective to explore a larger and more complex environment without the need for specialised building materials or tools, or to permanently sacrifice individuals to such assemblies. Two distributed self-assembly algorithms are presented and tested. Four prototype force-aware construction modules are also designed and built to validate the approach (Fig. 1b). The source code and prototype hardware design files are available on GitLab [14].

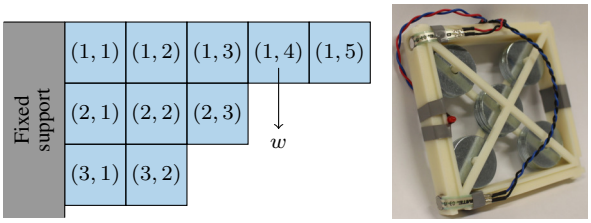


Fig. 1. (a) An example cantilever constructed from square self-assembling robotic agents, starting from a fixed support, who aim to extend as far as possible while keeping the links between agents from breaking. The numbers show the agent location coordinate system. Each agent has weight w (shown once for simplicity). (b) A proof-of-concept force-aware construction module.

A. Related Work

Bridge construction arising in robot collectives as a result of simple, local rules has been demonstrated for a group of soft-bodied robots by Malley *et al.* [15]. Other researchers have developed algorithms that construct bridges in similar scenarios while optimising bridge length and cost [16]. These works assume that connections between robots never break, potentially reducing their applicability to the real world.

Incorporating force sensing into robotic construction systems has previously been shown to be an effective way of ensuring structures do not collapse. Genetic algorithms have been used to invent structures that satisfy high-level design criteria while remaining stable, such as to build the longest possible cantilever [17], [18]. Other research shows how the assembly sequence of a wide range of predefined structures can be automatically generated to ensure stability at all stages by always choosing the most stable valid construction step to make next [19]. In these works, construction was either completed by a human or a single robot.

A team of robots has a choice when building structures: they could either use an external building material, or their own bodies. The benefits of a force-aware approach have been explored for truss-building robots, requiring only a few robots to build large structures out of force-aware components [20]–[22]. A group of self-assembling robots does not need to rely on the availability of external materials, but at the cost of requiring a larger number of autonomous

This research was funded by the Engineering and Physical Sciences Research Council through a Doctoral Training Partnership Scholarship.

The authors are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK [enbray1, r.gross]@sheffield.ac.uk

robots. A force-aware self-assembly algorithm for structures such as cantilevers is developed in [23] in which agents place as soon as they reach a position near a link that is close to failure. The work we present in this paper considers the self-assembly of cantilevers similar to this, but here we enable multiple robots to build on the structure concurrently, and also allow robots to explore more of the structure than just the first link that is close to failure.

II. Problem Formulation

This paper considers a homogeneous swarm of N robots (referred to as *agents*) that self-assemble into cantilevers as shown in Fig. 1a. Agents are squares of side length l and weight w that reside in a 2D grid for simplicity. Agents are assumed to be able to connect to each other on any face, and also to any point of a *fixed support* surface lying within the grid. Positions are referred to in $(row, column)$ format, where position $(1, 1)$ is the first cantilevering agent, row number increases downwards, and column number increases away from the fixed support, in this case to the right. The fixed support occupies positions $\{(r, c) \mid r > 0 \wedge c = 0\}$. An agent can move one step along the perimeter of the existing structure in each *timestep* by stepping into an empty space in its Moore neighbourhood, as long as this space is in the von Neumann neighbourhood of another agent or the fixed support. Construction is only permitted for positions $\{(r, c) \mid r > 0 \wedge c > 0\}$. The *length* of the structure L is the number of columns in the structure. Cantilevers are restricted to have *continuous* rows and columns, meaning each column must be filled from row 1 without any gaps, and $n_c \geq n_{c+1}$ where n_c is the number of agents in column c. Connections between agents are termed *links*: links between agents on the same row and column are called *row* and *column* links respectively, and the *links of column* c are defined as the row links on the left of c and the column links within c.

A continuous supply of agents is available at position $(0, 0)$, each of which must find its place in the structure. Each agent has the ability to sense the moment M and axial force F in the links on each of its four faces. These are compared to allowable limits $M_{allowable}$ and $F_{allowable}$ respectively to calculate the *criticalness* γ of each link as:

$$\gamma = \max\left(\underbrace{\frac{|M|}{M_{allowable}}}_{\gamma^M}, \underbrace{\frac{\max(F, 0)}{F_{allowable}}}_{\gamma^F}\right) \tag{1}$$

Each combination of $M_{allowable}$ and $F_{allowable}$ is referred to as a *limit pair*. Links with $\gamma \geq 1$ are described as *critical* and deemed to be near to but not necessarily at failure, analogous to how civil engineers consider safe stress levels in a building to be considerably lower than the failure strength of the materials. If no links are critical, the structure is defined as *stable*, else it is *unstable*. The objective of this work is to self-assemble cantilevers from a limited supply of agents that are as long as possible, while aiming to correct unstable configurations as soon as they occur during the construction.

Note that (1) implies that links are strong to moments in all directions, and strong in compression but weak in tension.

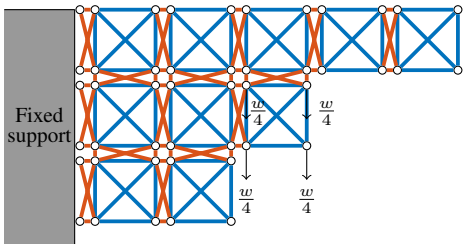


Fig. 2. Conversion of the cantilever in Fig. 1a to a truss, showing agents as blue members and links as orange members; a weight of $\frac{w}{4}$ acts downwards from each agent node (only shown for one agent). Unmade links are hidden.

We also assume that the shear strength of links is large in comparison to the axial and moment capacity, thus is not included in the calculation of γ. These strengths are similar to those of a simple mechanical connection such as a press stud, or existing modular robots such as SMORES [4]. While this work considers only M and F for simplicity, the exact failure criteria of a real robotic system will be unique to the mechanical design of its linkages.

A. Simulation Environment

The agents are simulated in a custom environment written in Python. The structure is modelled as a 2D truss as shown in Fig. 2, a method inspired by [18]. The forces within truss members are calculated using the anaStruct module for Python [24], and converted into a value of M and F for each link. This is fast to calculate, but does not take into account dynamic forces as robots move about the structure. A side length of $l = 0.1\,\mathrm{m}$ is used to reflect an estimate of existing modular robotic platforms, and the weight is conservatively calculated as if each agent were a solid cube of aluminium, giving $w = 19.3\,\mathrm{N}$, significantly greater than existing modular robots [4]–[6].

III. Structural Optimisation

Optimal cantilevers should be calculated to measure the performance of the self-assembly algorithms against. A configuration of N agents and length L is considered optimal if it is stable and no other stable configuration of N agents exists that has length $> L$; there may be multiple optimal configurations of N agents. A different optimisation procedure is used for small $(N \leq 50)$ and large $(N > 50)$ structures.

A. Small Structures

For small structures, the process of finding optimal structures begins by listing all the cantilever configurations for a given N using the number theory concept of integer partitioning [25]. A cantilever configuration may be represented by the number of agents in each column as $[n_1, n_2, ..., n_L]$, hence $\sum_c n_c = N$, and this vector is a partition of N. Since rows and columns are assumed to be continuous, this vector uniquely describes a cantilever configuration. For example, the structure in Fig. 1a can be represented as $[3, 3, 2, 1, 1]$, and this is the only valid configuration for this partition.

All cantilever configurations of $N \leq 50$ agents were listed, and each modelled to find the maximum M and F in all

links. A limit pair was then specified and these precomputed data analysed to find the configurations that result in the longest stable structures of N agents for all $N \leq 50$ for this limit pair. This exhaustive search guarantees optimality.

B. Large Structures

The previous approach becomes intractable for large structures as the number of configurations $\propto \exp(\sqrt{N})$. Another method is therefore used to reduce the number of configurations that need to be tested. This is based on the observation that the optimal arrangement of agents at the tip remains constant as N and L increase.

The optimisation procedure for large structures begins by specifying a limit pair and a number of agents N_{test} that is believed might be able to build a stable cantilever $L_{max}+1$ agents long, where L_{max} is the length of the longest optimal configuration that has been previously found for this limit pair. Exploiting the observation about the configurations at the tip, we first consider all known optimal configurations of $N < N_{test}$ agents and $L > (L_{max} - 5)$. The number of agents in each column of each stable configuration are compared to find the longest portion at the tip that is common across all these configurations. All configurations of N_{test} agents are enumerated as for the small structures, but those that do not include this shape at the tip are removed from this list. Exhaustive search again ensures these cantilevers are optimal, under the given assumption about tip configuration.

Once the reduced list of configurations has been generated, each one is modelled and any stable configurations of the specified N_{test} agents are saved. The procedure is repeated with different N_{test} until the minimum number of agents required to build a stable structure of length $L_{max}+1$ with links of the given strength is found.

IV. Algorithm Design

This section describes the self-assembly algorithms, beginning with a discussion of the theoretical basis behind them. Two algorithms are presented, called the *sequential* and *parallel* algorithms. Both algorithms begin with an agent placed in position $(1, 1)$. Future agents initialise in position $(0, 0)$, and are referred to as *active agents* until they are placed, whereupon they are no longer able to move. In the sequential algorithm, new active agents are released only once the previous one has found its place, whereas the parallel algorithm allows for multiple active agents concurrently. Both incorporate measurements of M and F within each link to construct stable structures, and run in a distributed manner on each agent. Active agents keep track of their movements to calculate their positions in the coordinate system.

A. Theoretical Basis

Here, a result from structural mechanics which influences the design of the self-assembly algorithms is introduced.

Theorem 1: Consider a cantilever of length L whose height $h(x)$ is a continuous and monotonically decreasing function of the distance x from a fixed support, as shown in Fig. 3a. The cantilever has constant breadth b into the page, and the only load is a uniformly distributed force of magnitude ρ per unit horizontal length. For such a cantilever, increasing the height $h(x)$ at a distance $x = x_0$ from the support causes a decrease in the maximum longitudinal stress experienced in the cross-section here, $\sigma_{max}(x_0)$.

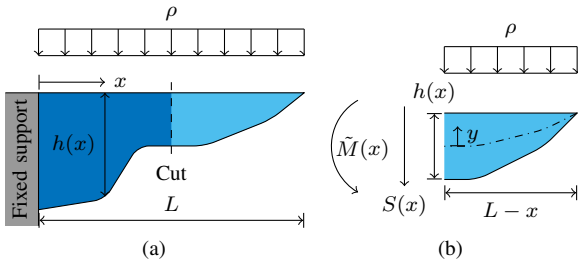


Fig. 3. (a) Profile of a beam under a uniformly distributed load of magnitude ρ per unit horizontal length, with continually-varying height $h(x)$ a distance x from the root; and (b) a cut through this cantilever at x, showing the internal shear force $S(x)$ and moment $\tilde{M}(x)$ on the exposed cross-section. The dashed line shows the neutral axis, which is assumed to be in the centroid of the cross-section.

Proof: Such a cantilever can be analysed using elastic beam theory [26]. A cut through the cross-section is made at x (Fig. 3b) to reveal the internal moment $\tilde{M}(x)$. For each x, $\sigma_{max}(x)$ occurs where the distance y from the neutral axis (assumed to be at the centroid of the cross-section) is as large as possible. Therefore, we obtain:

$$\sigma_{max}(x) = \frac{\tilde{M}(x)y}{I(x)} = \frac{\frac{\rho}{2}(L-x)^2 \cdot \frac{h(x)}{2}}{\frac{bh(x)^3}{12}} = \frac{3\rho(L-x)^2}{bh(x)^2} \quad (2)$$

where $I(x)$ is the second moment of area of the cross-section. Thus $\sigma_{max}(x_0)$ decreases as $h(x_0)$ increases. ■

These equations are derived for beams where the height is a continuous function along its length, under a uniformly distributed load. The work presented in this paper concerns cantilevers made of discrete agents under the self-weight of each agent. The exact equations are therefore not applicable, although the trends are used to inform the design of the self-assembly algorithms.

B. Sequential Algorithm

In the sequential self-assembly algorithm, the agents in the structure sample their sensor readings once an active agent places itself, and hold these values to communicate to the next active agent. Two variants of the sequential algorithm have been developed, called the *message-passing* and *local* variants, which differ only in how the active agent receives the force information from the existing structure.

The message-passing variant is shown in Algorithm 1. Lines 1 – 12 detail the operation of the agent while it is active. It first travels to the tip of the cantilever, receiving data from the agents in row 1. These data contain the maximum M and F observed for any links belonging to the corresponding columns, which allows the active agent to

create four arrays $\{\boldsymbol{\gamma}^{\alpha,\beta} \mid \alpha \in \{M, F\} \wedge \beta \in \{\textit{row, col}\}\}$ for the M and F in row and column links respectively. These arrays are arranged such that the c^{th} element of the appropriate array contains the maximum γ^M or γ^F in row or column links belonging to column c, which is given the symbol $\gamma_c^{\alpha,\beta}$.

The algorithm aims to construct cantilevers that are as long as possible, thus the active agent places at the tip if it believes the structure is stable based on the force information it has received (line 5). Otherwise, it chooses a column to reinforce at the bottom of the structure to better transfer the load onto the fixed support. Placing in a column will increase the column's height, which according to Theorem 1 causes a reduction in the maximum longitudinal stress on its left-hand face. A good approach would thus be to add to the bottom of columns that have high $\gamma^{\alpha,row}$ such that this reduced σ also causes a reduction in M and F. However, the case analysed in Theorem 1 is that of a cantilever of continually varying height and a constant weight per unit length: to exploit this information in the case of a structure built of discrete agents with their own self-weights, we use a probabilistic approach where the active agent is more likely to place in columns with links of high γ, but not guaranteed to. A probability mass function $p_{col}(c)$ describing the probability of placing in column c is calculated from the $\boldsymbol{\gamma}^{\alpha,\beta}$ arrays as follows (line 7):

1) For each member $\gamma_c^{\alpha,\beta}$ of each array $\boldsymbol{\gamma}^{\alpha,\beta}$, calculate an *urgency distribution* $\boldsymbol{\nu}_i$ where $1 \leq i \leq 4L$. This is calculated as the Gaussian distribution centred on column c with variance $(\gamma_c^{\alpha,\beta})^{-2}$, then scaled by a factor of $(\gamma_c^{\alpha,\beta})^2$. The area underneath this curve is equal to $(\gamma_c^{\alpha,\beta})^2$, so this spreads the impact of the criticalness in column c across the whole cantilever, but prioritises the region around column c the most. It is chosen to square $\gamma_c^{\alpha,\beta}$ to magnify the effect of critical links, for which $\gamma_c^{\alpha,\beta} \geq 1$.
2) For a cantilever of length L, there will be $4L$ urgency distributions, which are summed to calculate the combined urgency distribution $\hat{\boldsymbol{\nu}} = \sum_{i=1}^{4L} \boldsymbol{\nu}_i^{\alpha,\beta}$.
3) $p_{col}(c)$ is calculated by dividing each element in $\hat{\boldsymbol{\nu}}$ by the sum of all the elements.

The distribution is sampled from without replacement in line 9 to produce a target column c_{target}. The active agent travels to the bottom of column c_{target}, and if this placement is locally determined to be valid (meaning row and column continuity is maintained) it places here. If not, another sample is drawn from the distribution. This repeats until a valid location is chosen: column 1 is always valid, thus the `while` loop terminates after at most L iterations.

Once the active agent has placed itself, it remains stationary and runs the message-passing procedure in lines 13 – 16. Each agent receives the M and F that the agent below either measures in the links on its own bottom and left faces, or larger equivalent values from an agent below. These measurements are compared to measurements in the equivalent links of the agent in question, and the maximum values are passed to the agent above. If any link is not currently made it reads a value of zero. The agent in the top row thus receives the maximum M and F in all row and column links of this column. These messages are passed after one active agent places itself and before another initialises, therefore the weight of the active agent is excluded.

Algorithm 1: The message-passing variant of the sequential self-assembly algorithm.

```
1  while row = 0 and column < L do
       // Active (gathering data)
2      Step one right;
3      Record M and F from element below (γ_c^{α,β});
4  if No link is critical then
       // Active (placing at tip)
5      Place at tip;
6  else
       // Active (placing to reinforce)
7      Calculate p_col(c) from γ^{α,β};
8      while Not placed do
9          c_target ← sample p_col without replacement;
10         Move to column c_target;
11         if Valid location then
12             Place here;
13 while True do
       // Placed
14     M_l, F_l, M_b, F_b ← M & F in left & bottom links;
15     for α in [M, F] and β in [l, b] do
16         Send upwards max(α_{β,self}, α_{β,element beneath});
```

The message-passing variant requires coordination between agents that have been placed to transmit information within the structure. The local variant was developed to reduce the coordination required by incorporating more knowledge from Section IV-A. For the beam considered in Theorem 1, $\sigma_{max}(x)$ occurs on the top and bottom of the cross-section as $|y|$ is greatest. In the case of a beam of discrete agents with their own self-weights, we therefore predict that the maximum M and F in links of a given column is likely to occur in a link connected to an agent on the edge of the cantilever. This means that the active agent can reasonably approximate how the maximum M and F vary along the length of the structure by only receiving the sensor measurements of these *exterior* agents, without the need for passing messages.

The local variant is similar to Algorithm 1, but in lines 1 – 3 the active agent must traverse the full perimeter of the beam and communicate with all exterior agents. It compares the M and F values measured by agents at the top and bottom of each column and saves the maximum ones to $\boldsymbol{\gamma}^{\alpha,\beta}$. This makes the message-passing behaviour described in lines 13 – 16 obsolete, but requires the active agent to traverse a longer distance to obtain the necessary data. Agents are still required to coordinate to sample their sensors once an active

Algorithm 2: The parallel self-assembly algorithm.

```
1  switch mode do
2      case gathering do
3          Record M and F from placed element
             above or below (γ_c^{α,β});
4          if Below cantilever in column 1 then
5              Calculate p_col(c) from γ^{α,β};
6              c_target ← sample from p_col without
                 replacement;
7              mode ← placing;
8          else
9              Make step*;
10     case placing do
11         if In column c_target then
12             Attempt placement;
13             if Placement succeeded then
14                 mode ← placed;
15             else
16                 c_target ← sample from p_col without
                     replacement;
17         if mode is placing then
18             Make step*;
19     case placed do
20         Do nothing;
21     case swapping do
22         mode ← previous mode;
23 if Agent stationary for > δ steps then
24     Attempt placement;
25     if Placement succeeded then
26         mode ← placed;
```

* Steps made if the agent is not another's sole support

agent has placed itself and hold this value to communicate with the next active agent. It would be possible to read the sensors as the active agent traverses instead, but this is not done so that the two variants are more comparable. The local variant is important as it validates the assumption that only the sensors of exterior agents are salient, and is built upon in the parallel algorithm.

C. Parallel Algorithm

The sequential algorithm has two drawbacks. Firstly, construction takes a long time as only one agent is active at once. Secondly, the 'sample and hold' procedure used in communicating the force information to the active agent increases the inter-agent coordination required.

In the parallel algorithm, agents follow a procedure based upon the local variant of the sequential algorithm, but with allowances to enable multiple agents to move at once. Active elements are released at fixed intervals δ. If the initialisation position is still occupied, they are released at the next possible timestep. The agents *advance* in a random order

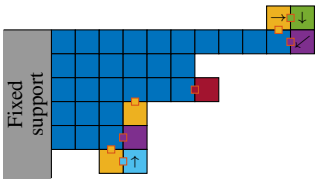


Fig. 4. Examples of special cases when the active link is set to the left face. Blue agents are stationary and yellow agents are active with the default active link (lower face when above row 1, and upper face otherwise). Agents of other colours are active with an active link on their left face. Links have orange borders and are filled the same colour as the active agent they are controlled by. Arrows show salient directions of motion.

each timestep to simulate synchronisation inconsistencies in a real system, and we assume sensing is instantaneous.

The procedure agents use when it is their turn to advance is presented in Algorithm 2, which shows they transition through four *modes*. On initialisation, they are in the *gathering* mode (lines 2 – 9), where they move around the perimeter of the structure and communicate with exterior agents to obtain the maximum M and F currently experienced by their links. When a gathering active agent begins advancing below the cantilever in column 1, it calculates the probability mass function $p_{col}(c)$ in the same way as the sequential algorithm, samples from it without replacement to set a target column c_{target}, and transitions to the *placing* mode.

In the placing mode (lines 10 – 18), active agents first check if they have reached c_{target}. If they have, they attempt to place here. If this placement is valid, the agent transitions to the *placed* mode where it no longer moves, and instead waits to communicate force information to passing active agents (lines 19 – 20). Otherwise it chooses another c_{target} without replacement. The agent moves towards c_{target} at the end of the advancement if it is still placing.

In the event of two agents attempting to occupy the same location, they communicate to swap internal states and 'become' one another. For two agents to swap requires that each remains stationary for one timestep, modelling the time cost this communication would have in real life: in the next timestep, each agent is in the *swapping* mode on lines 21 – 22, therefore does not move. If an agent tries to swap with another agent that is already swapping, it instead remains stationary for this timestep.

To avoid active agents blocking each other from moving, they attempt to place if they have been stationary for too many steps after advancing (lines 23 – 26). The number of steps before timeout is chosen as δ.

Instead of only updating M and F after agents place as in the sequential algorithm, the parallel version does so at the end of every timestep, thus the weight of active agents is measured. However, if active agents attach to all available contact surfaces they could reinforce the structure before their location is finalised and thus hide critical links. *Active links* are introduced to counter this by ensuring that active agents only make one link to the fixed portion of the cantilever. We restrict each active agent to only control of one of its four possible links at a time, referred to as its active

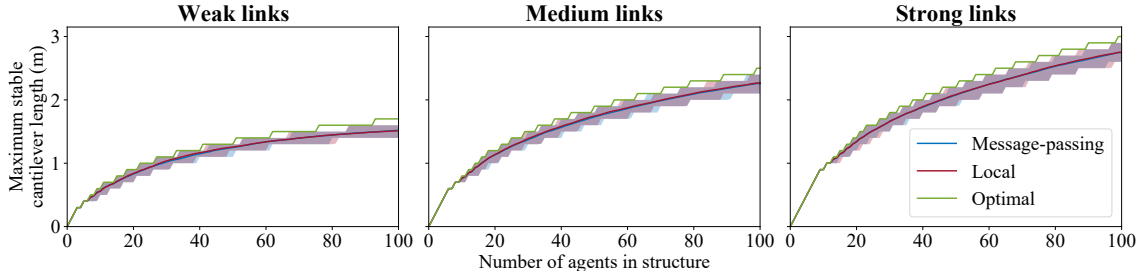


Fig. 5. Mean lengths of the longest stable structure as more agents are added produced by the message-passing (blue) and local (red) variants of the self-assembly algorithm for 400 trials, compared to the optimum achievable (green). Error bars show the 5th and 95th percentiles.

link. By default, the active link is set to be on the lower face of each active agent if the agent is above row 1, and on the upper face otherwise. Fig. 4 illustrates the exceptions to this rule where the active link is set to the left face instead. If the active agent is above row 1, this only occurs when the agent left of it is also active and there is no placed agent underneath (green agent). If the active agent is below or in row 1, this occurs if either there is no agent above (red agent), there is an active agent above and a placed agent to the left (purple agents), or there is an active agent above that is not swapping, an active agent to the left, and the active agent in question is moving upwards (cyan agent). Each active agent in Fig. 4 is only connected to the placed agents through a single chain of links, therefore not providing any additional support to the structure. In a real system, the structure would deflect and potentially cause robots to rest on each other, but this effect is not considered here. Additionally, if an active agent is connected by more than just its active link, it knows it is supporting another active agent and thus cannot move.

V. Simulation Results

A. Limit Pairs

The algorithms are tested for three limit pairs to examine their applicability in a range of scenarios. Each pair is generated by choosing a number of agents N_g, arranging $N_g + 1$ agents in a cantilever 1 row thick, and measuring M at the root, which will be the largest across all the links [26]. $M_{allowable}$ is set to be 10% less than this value, thus a single-thickness cantilever of length N_g is just stable. A single-thickness cantilever has $F = 0$ at the root, thus cannot be used to calculate $F_{allowable}$, which is instead calculated as $10N_g w$. These limits reflect how certain connections are better able to hold structures vertically than horizontally: the SMORES robots described in [4] can hang an average of 11 agents vertically from a single connector without failure, but only cantilever 3 agents. We chose to use $N_g \in \{3, 6, 9\}$ agents: the corresponding links are referred to as weak, medium, and strong, and correspond to limit pairs $[M_{allowable}, F_{allowable}] = [13.9\,\mathrm{N\,m}, 579\,\mathrm{N}]$, $[42.6\,\mathrm{N\,m}, 1159\,\mathrm{N}]$, and $[86.9\,\mathrm{N\,m}, 1738\,\mathrm{N}]$ respectively.

B. Optimal Structures

The optimisation described in Section III was carried out for $N \leq 100$ agents for each of the three limit pairs.

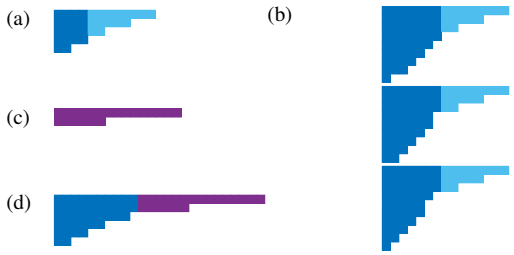


Fig. 6. Optimal structures with a given number of agents for different limit pairs. (a) A $1.2\,\mathrm{m}$ cantilever with weak links requires 33 agents. (b) Increasing length to $1.5\,\mathrm{m}$ requires 62 agents and results in three stable configurations as shown. (c) Strong links can reach $1.5\,\mathrm{m}$ with only 21 agents. (d) An optimal structure of strong links reaching a length of $2.5\,\mathrm{m}$, which requires 62 agents. The common portion at the tip for weak links is shown in cyan, and for strong links in purple.

The results can be seen in Fig. 5, which shows that higher allowable limits lead to longer possible stable structures, and the rate of increase in L slows as agents are added. Fig. 6 shows examples of these optimal structures chosen as the minimum N that can reach the specified L. In Figs. 6a and 6b, weak links are used, and short, deep cantilevers are constructed; there is only one optimal configuration for $L = 12$, $N = 33$, but three for $L = 15$, $N = 62$. Note also how the portions at the tip are the same for all these structures as observed in Section III-B. Figs. 6c and 6d show structures with strong links, where fewer agents are required to reach equivalent lengths as more slender cantilevers are possible.

C. Sequential Algorithm Performance

Both variants of the sequential algorithm were tested for all three limit pairs up to $N = 100$ agents. A total of 400 trials were performed with each limit pair to test the performance of the stochastic algorithms. The average length of the longest stable structure that was produced for a given number of agents in each trial is shown in Fig. 5. We see that both variants of the algorithm require similar numbers of agents to reach a stable structure of a given length, which on average is slightly below the optimum number, with relatively little variation between trials. In all intermediate unstable states, the two variants exceed the allowable limits by similar amounts within each limit pair. The local variant hence achieves comparable performance to the message-passing variant, despite requiring significantly less coordina-

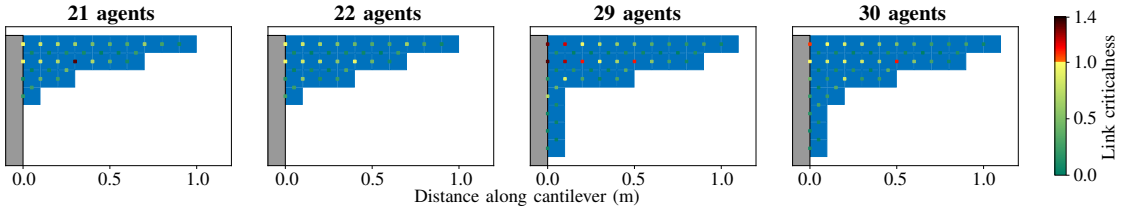


Fig. 7. Frames from a sequence produced by the local variant of the sequential algorithm for weak links. Links are coloured by their criticalness.

tion between agents. We thus predict that performance of the parallel algorithm will not be hindered by only transmitting link forces of exterior agents to active agents.

The Supplementary Material contains animations of the self-assembly sequence of a number of trials. Fig. 7 shows selected frames of the sequence generated during one trial of the local variant. At $N = 21$ agents, the only critical link is a row link in column 4. This belongs to an external agent, so the next active agent receives this information and has a high $p_{col}(4)$: the active agent places here, which makes the structure with $N = 22$ agents stable. When $N = 29$ agents, the most critical link in column 3 does not belong to an external agent, so the next active agent does not become directly aware of it. However, the link left of the agent at the top of the adjacent column 2 is critical, so the combination of the urgency distributions causes the next agent to place in a location that reduces the criticalness of the link in column 3 regardless. In all these structures, the top left corner has links of high γ compared to the rest of the structure.

An example structure with the maximum N tested of 100 agents constructed by the message-passing variant for links of medium strength is shown in Fig. 8. We can see that at this point γ^M is larger than γ^F in most columns, and that row links experience higher M and F than column links; all four traces will, however, influence $p_{col}(c)$. Theorem 1 showed how the longitudinal stress can be reduced on the left face of a given column by increasing the number of agents in that column, which will lead to a reduction in M and F in row links. An approach akin to the proof for Theorem 1 can be taken to show that M and F in column links will be similarly affected, but is not shown here as the criticalness of column links is less significant than that of nearby row links in the scenario considered in this paper.

D. Parallel Algorithm Performance

In the previous section, it was demonstrated that the sequential algorithm can construct stable cantilevers of near-optimum length, and the idea that agents should place near columns of high γ was validated: the parallel algorithm extends these concepts. Fig. 9 shows a comparison of the sequential and parallel algorithms in which 100 trials of up to $N = 100$ agents were performed for each $\delta \in \{4, 6, 8, 10, 12, 14\}$ timesteps. Two minor modifications are made to the local variant of the sequential algorithm to allow a fairer comparison to the parallel algorithm. Firstly, sensors are no longer sampled and held until the active agent passes over them: instead the active agent has an active link as in

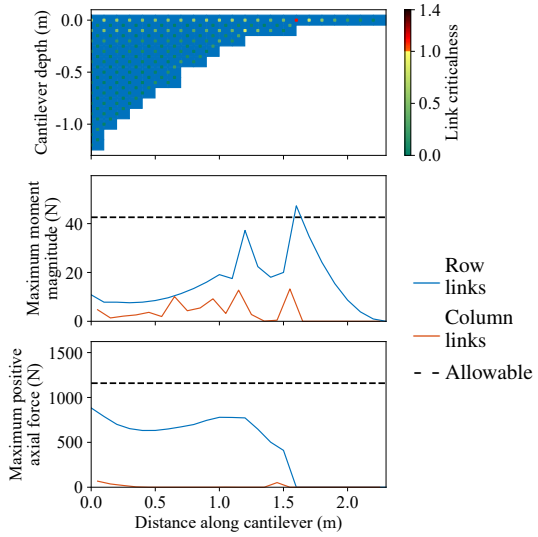


Fig. 8. A structure of 100 agents produced by the message-passing variant of the sequential algorithm with links of medium strength.

the parallel algorithm, and its own weight is included in the measurements. Secondly, the active agent remembers which locations are valid for it to place in as it moves over the structure and draws from p_{col} until a valid column is found before starting to move, so it can travel directly there.

Fig. 9a shows that the final L increases as δ increases, and is greatest in the sequential case. The construction time is significantly lower for the parallel algorithm (Fig. 9b). Stronger links tend to require a larger δ before they can achieve a comparable L to the sequential algorithm. This is partly because agents are more likely to timeout and place where they are due to congestion for small δ: when $\delta = 4$ steps, around $70\,\%$ of the agents place in this manner, but this drops to near zero for $\delta \geq 10$ steps, regardless of link strength. Increased agent timeout leads to short and tall structures, which as shown in Section V-B are good for weak links, but sub-optimal for stronger links.

Another consideration is the maximum γ in the structure as it is assembled (Fig. 9c). For weak links, the parallel algorithm creates structures that exceed $M_{allowable}$ and $F_{allowable}$ by a greater amount during construction than in the sequential case, but this exception decreases as δ increases. For strong and medium links, the maximum γ during the construction sequence is usually less than in the parallel case, and increases with larger δ. This is another result of the greater agent timeout at low δ creating short and tall structures. For all link strengths, the maximum γ

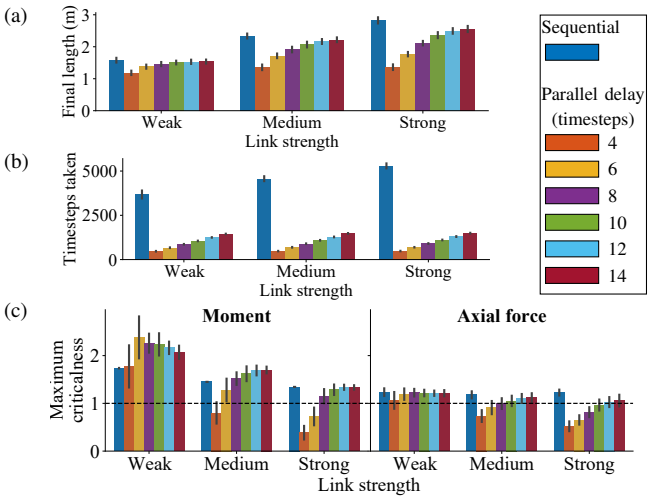


Fig. 9. Performance comparison between the local variant of the sequential algorithm and the parallel algorithm with varying delay between when agents are added, presented for 100 trials of each setup. Bars show mean values, and black lines show one standard deviation from this mean.

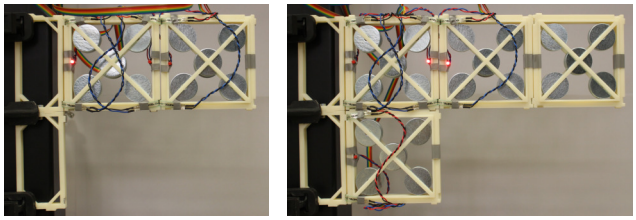
during construction tends to that of the sequential algorithm as δ increases.

VI. Hardware validation

The approach demonstrated in this paper assumes that agents are able to sense force in their links. This is not commonly employed in existing modular robotic platforms, but a method of detecting forces between connected agents is demonstrated in [21]. A proof-of-concept prototype connection mechanism inspired by this work is presented here. It is shown in Fig. 1b, and employs permanent neodymium magnets to connect modules together with force-sensitive resistors (FSRs) in between to measure force. FSRs are a relatively inexpensive component whose resistance changes as pressure is applied: they are not as accurate as strain gauges, but do not require expensive signal processing hardware to use and so are acceptable for this proof of concept.

One 3D printed frame was designed for the agents and another for the fixed support. Each agent face contains a male and female connection. Male connections have magnets that protrude from the surface of the frame and have an FSR attached to them, whereas the magnets in the female connections are recessed, and feature an FSR and a paper spacer. Protrusions of the magnets in this manner ensure the link is strong in shear but weak in tension and bending, as is the case in the simulation environment. Only one face of the fixed support has connections on it, and these connections exclude the FSRs; the depth of the magnet is adjusted so that the module spacing at agent–support interfaces is the same as at agent–agent interfaces. The frames are hollow to reduce cost, and filled with steel washers to provide enough weight such that each agent has a mass of 280 g.

The FSRs are connected to an ATmega2560 microcontroller which analyses the information from them. When the moment at a joint increases, one sensor will be compressed and the other pulled apart. A measurement of moment can



(a) (b)

Fig. 10. Cantilevers built from the hardware prototype: (a) a single-thickness cantilever 2 modules long has a critical link at the root, as indicated by the lit LED. (b) Adding support in position $(2, 1)$ allows the length to be increased by 1 module, and causes the link to the right to become critical.

therefore be made by taking the difference between these two readings. This is used to control an LED on each connection face to turn on when the moment exceeds a prespecified value, and thus the link gets closer to failure.

The Supplementary Material includes a video of the system under operation, and Fig. 10 shows the configurations with two and four agents from this demonstration; note only salient connection faces are equipped with sensors to reduce cost. This video shows that when two modules are connected in a line, the link at the root is close to failure, indicated by the lit LED in Fig. 10a, and a third module in this row will break it. Placing this module in position $(2, 1)$ instead provides support so the link is no longer critical, allowing another module to be placed in position $(1, 3)$ (Fig. 10b). This supports the idea used in the self-assembly algorithms that modules should place near critical links.

VII. Conclusions

This paper examined how cantilevers can be constructed from force-aware robotic agents. Two distributed self-assembly algorithms that exploit local force information were presented. The algorithms are able to induce self-assembly into stable cantilevers, the length of which was shown to be near the optimal length for a given number of agents. The parallel algorithm was shown to result in cantilevers of comparable length to the sequential case, but considerably quicker. Prototype hardware was presented to demonstrate how sensors could be added to links to measure forces.

There is considerable scope for future work. This paper focused on cantilevers, but when the other side of the chasm is reached, agents should redistribute to make a more optimal bridge. The cantilevers in this paper are only attached to a vertical support surface, but could be improved by allowing construction above the support surface too. The simulator could be modified to account for dynamic forces as the agents move. The final goal of this work is to apply the self-assembly algorithm to a modular swarm robotics platform to demonstrate a fully-autonomous system, which should include more accurate force-sensing hardware, such as strain gauges, in addition to a method of autonomous locomotion.

References

[1] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *International Journal of Computer Applications*, vol. 126, no. 2, pp. 31–37, 2015.

[2] K. Stoy, D. Brandt, and D. Christensen, *Self-Reconfigurable Robots: An Introduction*. Cambridge: MIT press, 2010.

[3] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[4] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots - design of the SMORES system," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4464–4469.

[5] J. W. Romanishin, K. Gilpin, S. Claici, 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 (ICRA)*, 2015, pp. 1925–1932.

[6] C. Parrott, T. J. Dodd, and R. Groß, "HyMod: A 3-DOF Hybrid Mobile and Self-Reconfigurable Modular Robot and its Extensions," in *Distributed Autonomous Robotic Systems: The 13th International Symposium*. Springer International Publishing, 2018, pp. 401–414.

[7] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1999, no. 1.

[8] N. J. Mlot, C. A. Tovey, and D. L. Hu, "Fire ants self-assemble into waterproof rafts to survive floods," *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7669–7673, 2011.

[9] C. Anderson, G. Theraulaz, and J.-L. Deneubourg, "Self-assemblages in insect societies," *Insectes Sociaux*, vol. 49, no. 2, pp. 99–110, 2002.

[10] S. Phonekeo, N. Mlot, D. Monaenkova, D. L. Hu, and C. Tovey, "Fire ants perpetually rebuild sinking towers," *Royal Society Open Science*, vol. 4, no. 7, p. 170475, 2017.

[11] B. Hölldobler and E. O. Wilson, "The Multiple Recruitment Systems of the African Weaver Ant Oecophylla longinoda (Latreille)(Hymenoptera: Formicidae)," *Behavioral Ecology and Sociobiology*, pp. 19–60, 1978.

[12] S. Garnier, T. Murphy, M. Lutz, E. Hurme, S. Leblanc, and I. D. Couzin, "Stability and Responsiveness in a Self-Organized Living Architecture," *PLoS Computational Biology*, vol. 9, no. 3, pp. 1–10, 2013.

[13] C. R. Reid, M. J. Lutz, S. Powell, A. B. Kao, I. D. Couzin, and S. Garnier, "Army ants dynamically adjust living bridges in response to a cost–benefit trade-off," *Proceedings of the National Academy of Sciences*, vol. 112, no. 49, pp. 15 113–15 118, 2015.

[14] E. Bray, "Distributed Self-Assembly of Cantilevers by Force-Aware Robots," 2021. [Online]. Available: https://gitlab.com/natural-robotics-lab/distributed-force-aware-cantilever-self-assembly

[15] M. Malley, B. Haghighat, L. Houe, and R. Nagpal, "Eciton robotica: Design and Algorithms for an Adaptive Self-Assembling Soft Robot Collective," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4565–4571.

[16] M. Andrés Arroyo, S. Cannon, J. J. Daymude, D. Randall, and A. W. Richa, "A stochastic approach to shortcut bridging in programmable matter," *Natural Computing*, vol. 17, no. 4, pp. 723–741, 2018.

[17] P. Funes and J. Pollack, "Computer Evolution of Buildable Objects," in *Evolutionary Design by Computers*, P. J. Bentley, Ed. Morgan Kaufmann, 1999, vol. 1, pp. 387–403.

[18] L. Brodbeck and F. Iida, "Automatic real-world assembly of machine-designed structures," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1221–1226.

[19] M. McEvoy, E. Komendera, and N. Correll, "Assembly path planning for stable robotic construction," in *2014 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2014, pp. 1–6.

[20] N. Melenbrink, P. Michalatos, P. Kassabian, and J. Werfel, "Using local force measurements to guide construction by distributed climbing robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4333–4340.

[21] N. Melenbrink, P. Kassabian, A. Menges, and J. Werfel, "Towards Force-aware Robot Collectives for On-site Construction," in *Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 2017, pp. 382–391.

[22] N. Melenbrink and J. Werfel, "Local force cues for strength and stability in a distributed robotic construction system," *Swarm Intelligence*, vol. 12, no. 2, pp. 129–153, 2017.

[23] P. Swissler and M. Rubenstein, "ReactiveBuild: Environment-Adaptive Self-Assembly of Amorphous Structures," in *2021 International Symposium on Distributed Autonomous Robotic Systems (DARS)*. Springer International Publishing, 2021.

[24] R. Vink, "anaStruct," 2020. [Online]. Available: https://github.com/ritchie46/anaStruct

[25] D. Guichard, *An Introduction to Combinatorics and Graph Theory*. CC BY-NC-SA 3.0, 2020. [Online]. Available: https://www.whitman.edu/mathematics/cgt_online/cgt.pdf

[26] S. H. Crandall, N. C. Dahl, and T. J. Lardner, *An Introduction to the Mechanics of Solids*, 2nd ed. McGraw-Hill, 1978.