ReactiveBuild: Environment-Adaptive Self-Assembly of Amorphous Structures

Petras Swissler and Michael Rubenstein

Northwestern University, Evanston IL 60208, USA,
pswissler@u.northwestern.edu, rubenstein@northwestern.edu

Abstract. *ReactiveBuild* is an algorithm that enables swarms of robots to build a variety of robust, environment-adaptive structures without pre-planning. Robots form structures by climbing their peers until either reaching a point closest to a goal location or until a neighboring robot recruits it for structural reinforcement. This contrasts with typical approaches to robotic self-assembly which generally seek to form some a priori shape. This paper demonstrates a simulated swarm of FireAnt3D robots using *ReactiveBuild* to form towers, chains, cantilevers, and bridges in three-dimensional environments.

Keywords: self-assembly, swarm robotics, bio-inspired, climbing robot

1 Introduction

Self-assembly is a set of behaviors that enable organisms such as cells and social insects to join together and form structures [1] [2] [3]. Self-assembled structures enhance swarm capability by allowing organisms to adapt to the environment. These benefits have driven research into the field of robotic self-assembly [4].

Of particular interest are the structures formed by certain species of ant, including towers [5], bridges [6], and chains [7]. In each case, the insects form these structures by *self-climbing* (climbing over their peers) and are thought to grow the structures based on environmental conditions and insect-insect interactions. These structures are therefore not only functional, but also adaptive to dynamic conditions. In contrast to this environment-reactive approach, most robotic self-assembly algorithms seek to form a priori, prescribed shapes [8] [9] [10] [11], limiting the adaptability of self-assembled structures.

One exception to this is [12], in which self-assembly and disassembly result from robots reacting to local traffic, taking inspiration from [6]. Another exception is [13], in which robots use local force information to build cantilevers an order of magnitude longer than those built without using local force information; local forces also guide self-assembly in nature [2] [5]. In both examples robotic self-assembly follows simple, decentralized rules.

The organization of self-assembled structures also differs between those of insects and those of most robotic self-assembly platforms. Insects form and reinforce self-assembled structures by grabbing each other at seemingly arbitrary

Fig. 1. Examples in this paper use FireAnt3D, which consists of three spheres and can walk on a floor, wall, and ceiling of its peers [22].

locations using their mandibles and legs, resulting in *amorphous* (not constrained to a lattice) structures [14] [15]. Contrasting this, most examples of robotic self-assembly form discretized, latticed structures [16] [17] [18]. Although such discretization simplifies the localization of individual agents within a structure [10] [11], environment-adaptive structures do not appear to require such localization. A latticed approach is problematic for structures starting from more than one location; misalignments are likely without careful alignment of all starting locations (difficult when operating in unknown, arbitrary environments). *Amorphous* structures can also better conform to arbitrary environments.

Several robots capable of *amorphous* self-assembly have been developed. In most, operation is limited to a 2D plane [12] [19] [20] [21], and several require grippers to be properly oriented prior to attachment, limiting spontaneity [12] [20]. To our knowledge, only FireAnt3D [22] has demonstrated 3D *self-climbing* while also forming the strong connections necessary to self-assemble robust structures (see Fig. 1). Unfortunately, work on algorithms for robust self-assembly of *amorphous* structures is also sparse [12] and has been limited to 2D.

This paper presents *ReactiveBuild*, a novel algorithm for the self-assembly of *amorphous*, environment-adaptive 3D structures. We validate *ReactiveBuild* using simulated FireAnt3D robots [22] to self-assemble four types of environment-adaptive structures: towers, chains, cantilevers, and bridges. Structures arise via a single set of environment-adaptive behaviors, are not latticed, and control stresses between individual robots to maintain structural integrity throughout construction.

2 Algorithm Description

ReactiveBuild enables robots to self-assemble environment-adaptive structures without requiring these structures to conform to a lattice. *ReactiveBuild* requires robots to have the capability to do the following:

- *Self-climb* and to climb the environment.
- Send messages to contacting robots.
- Sense local forces and the direction to some goal position.

These capabilities match what we expect to be possible with updated versions of FireAnt3D, though it should be emphasized that any robotic platform with the above capabilities can execute *ReactiveBuild*.

Algorithm 1: ReactiveBuild: Locomotion

```
initialize: locomote ← true, next_move_direction ← Sense(direction to goal)
while locomote do
    StepTowards(next_move_direction)
    last_move_direction ← next_move_direction
    next_move_direction ← Sense(direction to goal)
    if next_move_direction == last_move_direction then
        // Robot would reverse
        locomote ← false
    end
    comm_in ← communication from other robots
    if max(comm_in) ≥ 1 then
        // Robot has been recruited
        locomote ← false
    end
end
```

Algorithm 2: ReactiveBuild: Communication

```
while true do
    if robot is part of the structure then
        Sense(sensed_force)
        recruit_value ← min(floor(B^(sensed_force/F−1)), J)
        for all contact zones do
            in_neighbor ← max(comm_in(this_zone)) − 1
            in_others ← max(comm_in(other_zones)) − 2
            comm_out(this_zone) ← max(recruit_value, in_neighbor, in_others)
        end
    end
end
```

ReactiveBuild consists of two parts: one governing locomotion before a robot joins the structure and another governing robot-robot communication after joining the structure. During locomotion (algorithm 1) the robot moves towards some goal location (e.g. moving towards a light) then joins the structure when either its next step would move backwards, or when another robot recruits it.

After joining the structure (algorithm 2) each robot senses local forces and outputs a recruitment value based on its sensed force and on the recruitment values of neighboring robots. The calculation of *recruit_value* in the algorithm depends on three factors. J is the maximum recruitment value; this controls how far away from an area of high stress a robot can recruit structural reinforcement (i.e. the extent of its jurisdiction). F is the threshold force; when $sensed\_force$ exceeds F, the robot begins recruiting structural reinforcement. B controls the growth of *recruit_value* after $sensed\_force$ exceeds F.

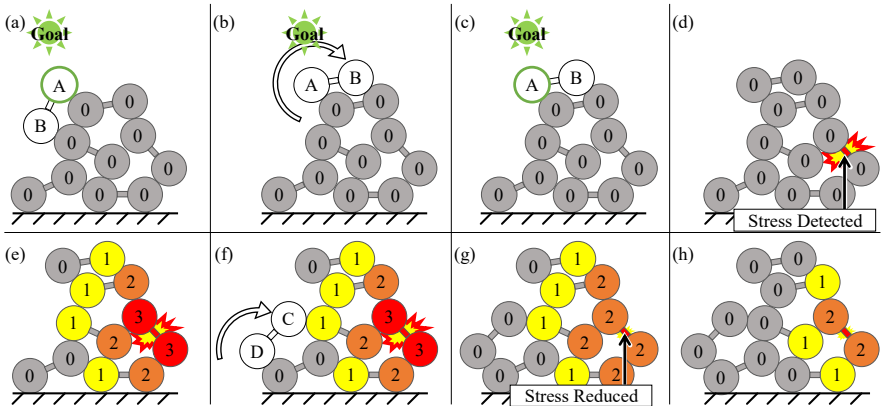

Fig. 2. An illustration of a small swarm of 2D robots executing *ReactiveBuild*.

The algorithm assumes that the robot can spatially differentiate incoming communications using discrete contact zones, such as the individual spheres of FireAnt3D, though *ReactiveBuild* still functions using only a single contact zone (removing spatial differentiation). Recruitment values decrement by one for each *hop distance* the message travels, similar to [23]. Both inter-robot contacts and intra-robot contact zones are a *hop distance* of 1 apart.

To simplify the algorithm we assume that only one robot is active at a time. This paper does not explore ways in which moving robots may interact, such as the traffic effects explored in [12]. Fig. 2 illustrates the function of *ReactiveBuild* using a 2D robot similar to [21]. The events depicted are as follows:

(a) A robot climbs the tower and must now decide in which direction to move. It determines that sphere A is closest direction to the goal.
(b) The robot therefore steps in the direction of sphere A.
(c) The robot again assesses which sphere is closest to the goal and again finds that sphere A is closest. Because a step in this direction would result in backwards motion, the robot stops and joins the structure. If the goal were further to the right and sphere B were closer, the robot would have continued moving towards the goal since this would result in forwards motion.
(d) The highlighted robot detects an increase in structural stress.
(e) This robot uses the observed stress to calculate a recruitment value of 3, which it then communicates to neighbor robots. Robots propagate this this recruitment signal through the structure, decrementing it by 1 for each *hop* of separation between spheres.
(f) Another robot climbs the structure and, eventually, sphere C contacts a sphere with a non-zero recruitment signal. The robot therefore stops and joins the structure.
(g) The addition of the new robot to the structure reduces structural stresses. The lowered stresses result in a decrease in recruitment value from 3 to 2.
(h) The decrease in recruitment value cascades through the structure, shrinking the recruitment region.

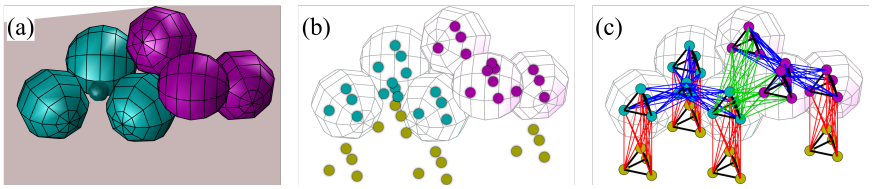

Fig. 3. Robot configurations (a) are converted into nodes arranged in tetrahedrons (b). Linear truss elements (shown as lines) connect nodes to create the FEM (c).

3 Simulator Overview

A bespoke simulator was built to validate *ReactiveBuild* using a virtual swarm of FireAnt3D robots. The simulated robots move towards a specified goal location using the locomotive method described in [22]. The simulation adds one robot at a time to the environment, waiting until the robot finishes climbing and joins the structure before adding another.

To accurately and efficiently simulate in-robot forces and connection stresses, the simulation formulates and solves a a finite element model (FEM) [24] after a robot stops and joins the structure. To do this, it first creates four nodes arranged in a tetrahedron at the location of each robot sphere and center, as well as inside the environment at the location of each environmental contact. Each node has three translational degrees of freedom with the exception of those associated with environmental contacts, which are fixed. A 0.25-unit gravitational load is applied to each robot sphere node (per-sphere weight is 1 unit).

Linear truss elements (limited to pure tension or compression) are then created between nodes. First, each tetrahedron of nodes is fully connected. Elements are then created to fully connect the nodes of each sphere to its associated robot center nodes, representing the robot structure. Finally, connections are modeled by fully connecting the nodes of contacting spheres, or between sphere and environmental contact nodes. Fig. 3 illustrates this process. The code used to formulate and run this model is adapted from [25].

The simulation solves the FEM in one step, assuming small deflections. To simulate robot force sensors, axial and shear forces, as well as bending and torsion moments are calculated from element forces. A robot's measured force, $sensed\_force$, is the sum of axial forces and bending moments averaged across all sphere-center connections; robots can only measure axial forces and bending moments. Connection stresses are calculated using all forces and moments associated with the connection and assuming a circular contact with radius 0.5. Because the focus of this paper is not the direct application of *ReactiveBuild* on real-world robots, element stiffnesses were selected based on qualitative review of the simulated robots. For completeness, the chosen stiffnesses are (in simulation units): $5e10$ for in-sphere elements, $1e10$ for sphere connection elements, and $2e9$ for robot structure elements.

4 Algorithm Validation

In this section a swarm of virtual FireAnt3D robots running *ReactiveBuild* spontaneously forms structures resulting from environmental conditions, all while

controlling stresses within these structures. Based on the position of the goal location in the environment, robots form four general types of structure: tower, chain, cantilever, and bridge. The results in this section are observational, with limited, high-level interpretation.

100 structures of 100 robots each are simulated at varying combinations of threshold force F, exponent base B, and maximum recruitment value J to understand their effect on structure formation. We chose values of F, B, and J based on their exhibition of different construction behaviors, specifically:

- The impact of F using $F = [1, 2.5, 5, 25]$, with $B = 3$ and $J = 5$.
- The impact of B using $B = [1.5, 3, 6, 10]$ at $F = [2.5, 5]$, with $J = 5$.
- The impact of J using $J = [1, 2, 5, 1000]$ at $F = [2.5, 5]$, with $B = 3$.

All distances given in this section are in sphere radii and are measured to sphere centers. Values of F do not correspond to any particular real-world values but are related to the weight of each sphere. The decision to limit experiments to 100 structures of 100 robots each was arbitrary and was not due to any simulator constraint or specific statistical reason. A simulation video is available at [26].

4.1 Tower

A tower should be as tall as possible while also being strong enough to withstand its own weight. Robots build towers by executing *ReactiveBuild* and targeting a goal 65 units above an environment consisting of a flat plane, as shown in Fig. 4. Robots are added one at a time, starting at random positions and orientations outside the tower, on the plane.

As F increases towers become taller, skinnier, and experience a higher peak connection stress (see Fig. 5). It was found that F is directly proportional to the peak stress present in the final tower ($R^2 = 0.999$), showing that F directly influences the maximum stresses present in the tower. We also found that the final height is proportional to the square root of F ($R^2 = 1.000$). After the structure matured (after $N = 25$), stress grew linearly with N ($R^2 = 0.870$), and height grew proportional to the square root of N ($R^2 = 0.983$).

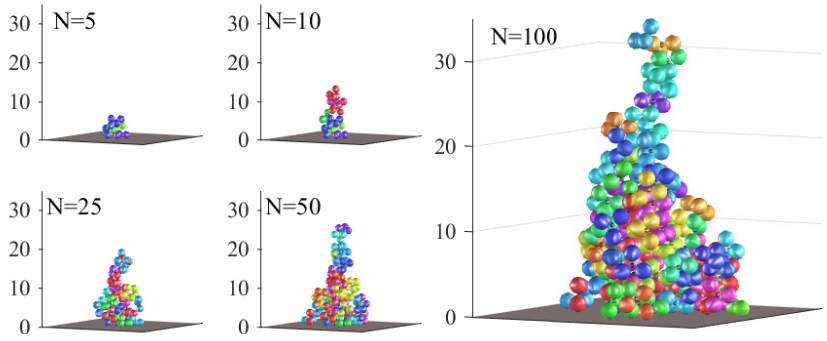

Fig. 4. Tower growth using $F = 10$, $B = 3$, $J = 5$.

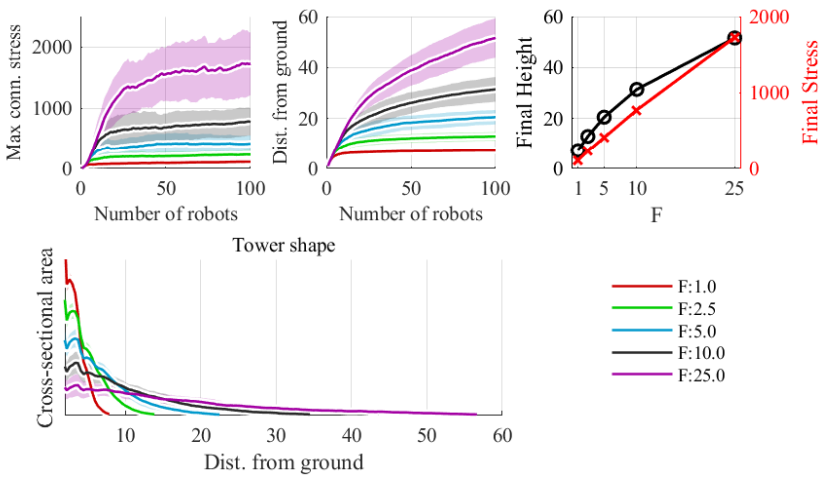

Fig. 5. Tower growth at five values of F; $B = 3$, $J = 5$. Lines represent the mean across 100 runs; shaded areas represent a region ± 1 standard deviation.

Structure shape is evaluated by summing the number of spheres within distance 1 of a given height across all final towers. The number of spheres (analogous to the cross-sectional area) is proportional to the square of the distance from the top (mean $R^2 = 0.992$ across all tested F). This indicates that the weight of the tower above any given location is proportional to the cube of distance from the top, and by extension the weight supported per unit area is linearly proportional to distance from the top. Therefore, these are not towers of constant stress, as ants appear to build [5]. This is not surprising: such towers require tighter and tighter packing of agents in the lower levels of the tower, and the towers formed by the simulated FireAnt3D robots have relatively uniform packing.

In separate trials, we found that both stress and height for the final structure are proportional to the inverse of B (across both F, $R^2 = 0.993$ and 0.976). B did not obviously change the shape of the structure. Similarly, towers became taller and experienced higher stresses as J decreased, with minimal differences between $J = 5$ and $J = 1000$. This suggests that *ReactiveBuild* controls structural stresses such that recruitment values beyond 5 are rarely used.

4.2 Chain

A chain consists of robots linked together, dangling from an edge to reach the lowest possible point while reinforcing against gravitationally induced stresses. To demonstrate *ReactiveBuild* achieving such a chain, simulated robots target a goal 600 units below the edge of the plane. As seen in Fig. 6, the environment terminates in a cylinder at the edge of the plane. Robots are added to the chain one at a time, at random distances and orientations behind the furthest-back above the highest robot.

The results for the chain are similar to the tower, as seen in Fig. 7; F is linearly proportional to the stress in the final structure ($R^2 = 0.998$). When ignoring the portion near the edge, the resulting structure has a similar shape to

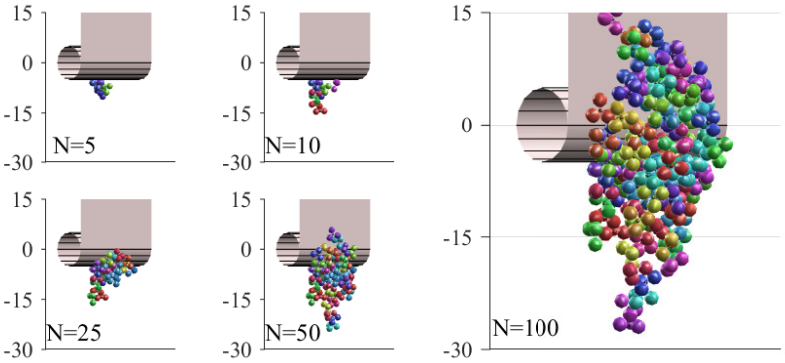

Fig. 6. Chain growth using $F = 10$, $B = 3$, $J = 5$.

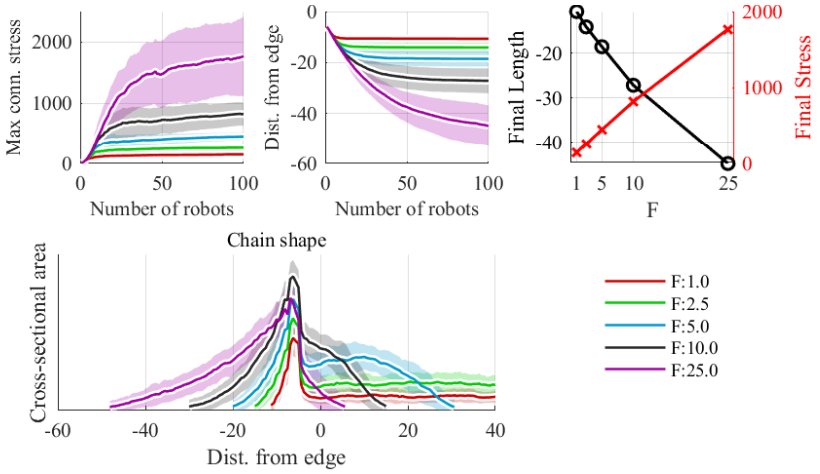

Fig. 7. Chain growth at five values of F; $B = 3$, $J = 5$. Lines represent the mean across 100 runs; shaded areas represent a region ± 1 standard deviation.

that of the tower, but is proportional to the distance from the tip raised to the 1.75 power (mean $R^2 = 0.992$ across all values of F). Another difference is that the final height is proportional to F raised to the 0.75 power ($R^2 = 0.999$). These differences may result from the chain having flatter shape relative to the tower, due to environmental geometry. Differences are also seen in the growth of the structure, as both stress and length are proportional to $log(N)$ after maturation (after $N = 25$) ($R^2 = 0.970$ and 0.901).

Trends for varying B match those from the tower: both stress and length are inversely proportional to B. Similarly, smaller values of J correspond to higher stresses and longer chains, with minimal difference between $J = 5$ and $J = 1000$.

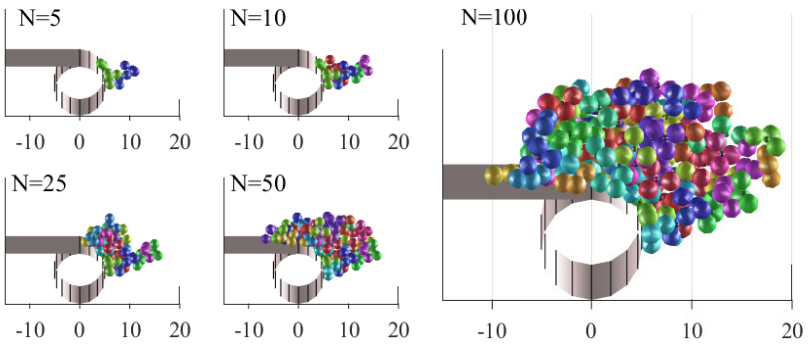

Fig. 8. Cantilever growth using $F = 10$, $B = 3$, $J = 5$.

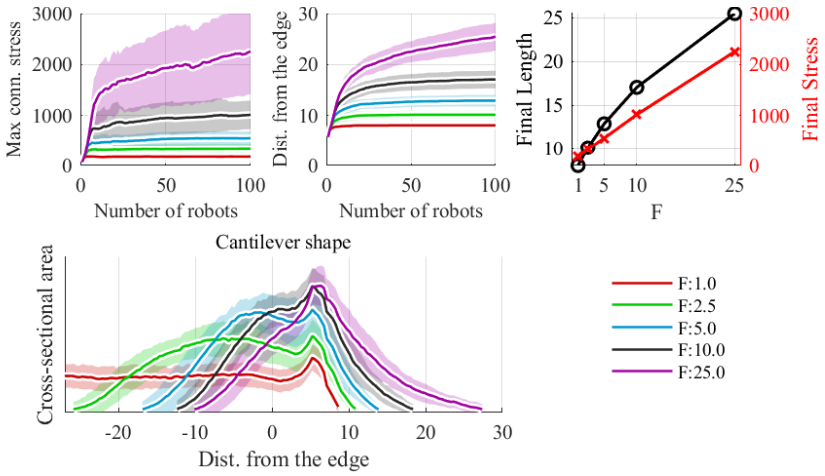

Fig. 9. Cantilever growth at five values of F; $B = 3$, $J = 5$. Lines represent the mean across 100 runs; shaded areas represent a region ± 1 standard deviation.

4.3 Cantilever

A cantilever is a structure that extends horizontally out from an edge; this structure has particularly challenging loading conditions. To demonstrate *ReactiveBuild* achieving a cantilever, simulated robots target a goal 45 units outward horizontally and 10 units upward vertically from the edge of a horizontal plane. As seen in Fig. 8, the environment terminates in a cylinder at the edge of the plane. Robots are added to the cantilever one at a time, at random distances and orientations behind the furthest-back robot.

As with the chain, a support structure emerges behind the edge. The support structure begins forming early in the construction of the cantilever and eventually becomes the primary area of construction. The support structure begins as reinforcement to an area of high stress resulting from the lengthening of the cantilever; this reinforcement eventually grows into a small tower at that location, which then requires its own support. This diminishes further lengthening

of the cantilever after about $N = [5, 17, 20]$ for $F = [1, 2.5, 5]$; the cantilevers constructed using $F = 10$ and $F = 25$ continue to lengthen for the duration of these trials. Tests with varying B and J showed similar effects as with the tower and chain and did not affect the formation of the support structure.

4.4 Bridge

The final structure is a bridge built simultaneously over a gap between two cantilever-style environments (see Fig. 10). Robots are added one at a time, alternating between the left and right side of the environment. All robots share the same set of behaviors, with the exception that the robots on the left target a goal location on the right, and vice versa; there is no coordination between sides before meeting in the middle. Robots are added to the structure until one robot successfully crosses the bridge to the other side or until $N = 200$. Bridges are considered to be unsuccessful if $N \geq 100$. Table 1 lists the parameter configurations for each of the trials run, with each trial consisting of 100 bridges built. As a consequence of environment geometry, robots would sometimes move under the environment cylinders, failing to reach the other side; these robots count towards the number of robots necessary to build a successful bridge.

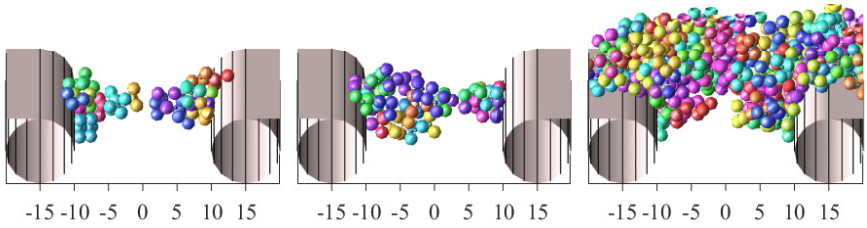

Fig. 10. Three separate bridges grown using F=10, B=3, and J=5. There are three general types of bridges: (left) two separated cantilevers, (center) two cantilevers with a narrow connection, (right) *unified support structure*.

Table 1. Table showing the number of robots to build a bridge across a gap. Medians and variances are based on a Weibull fit of simulation results to account for censored data. Success Rate (SR) is the proportion of bridges completed in under 100 robots.

Parameters			Gap Width								
			20			25			30		
F	B	J	Med.	Var	SR	Med.	Var	SR	Med.	Var	SR
5	3	5	9.5	6.3	99%	27	20	22%	-	-	0%
10			6.6	1.9	100%	17.8	14.1	100%	49.0	36.5	67%
25			6.5	1.8	100%	11.7	3.6	100%	18.6	10.7	99%
10	3	2	6.6	1.9	100%	15.9	8.4	100%	49.0	31.5	87%
		10	6.6	1.9	100%	16.2	9.9	98%	47.1	31.6	67%
10	1.5	5	6.6	1.9	100%	14.8	6.4	100%	40.4	29.1	95%
	6		6.7	2.2	100%	19.7	17.3	97%	44	37	24%

Higher F resulted in more reliable crossing of wider gaps. Lower B also increased reliability. Notably, for a 30-unit gap, runs using $F = 10$, $B = 1.5$ had a similar success rate as runs using $F = 25$, $B = 3$, but did so with a higher median number of robots (i.e. with a greater occurrence of the *unified support structure* type of bridge), which may be desirable in certain circumstances. Lower J also corresponded to higher success rates.

5 Conclusion

In most situations, the effects of B and J were indistinguishable from an adjusted value for F, but for a bridge built from two ends it appears to be advantageous to use small B and J to localize reinforcement to areas of highest stress. It therefore seems ideal to default to the use of small values of B and J, scaling F based on robot capability. Because these parameters were able to control connection stresses it will be possible to use *ReactiveBuild* for specific self-assembling robots.

This paper also showed three benefits of *amorphous* self-assembly through the simulation of FireAnt3D robots. First, robots were able to adapt the structure to the geometry of the environment. Second, robots were able to climb about the structure without consideration for proper alignment to neighbors or the environment. Finally, robots were able to build multi-origin structures without need for long-range communication or alignment.

There are several readily-apparent ways to extend the work presented in this paper. First and most importantly would be to adapt the algorithm and simulation to support multiple active robots at once. Second, to explore the impact of environment geometry on self-assembled structures. Finally, another interesting extension would be to evaluate the performance of *ReactiveBuild* on other robotic systems, both lattice-based and amorphous.

ReactiveBuild approaches self-assembly in a way distinct from traditional methods. To our knowledge, this is the first demonstration of a *amorphous*, environment-adaptive robotic self-assembly algorithm in 3D. The simulations performed in this paper validate *ReactiveBuid*, the FireAnt3D concept demonstrated in [22], as well as the benefits of *amorphous* self-assembly in general. We hope that the work outlined in this paper encourages others to explore the self-assembly of *amorphous* and environment-adaptive structures.

6 Acknowledgement

We thank Orion Kafka and Newell Moser for their help in strengthening the portions of this paper related to solid mechanics, and Thomas Bochynek for his help in strengthening the portions of this paper related to biology.

References

1. Anderson, C., Theraulaz, G., Deneubourg, J.: Self-assemblages in insect societies. Insectes sociaux 49(2), 99-110 (2002).
2. Peleg, O., Peters, J., Salcedo, M., Mahadevan, L.: Collective mechanical adaptation of honeybee swarms. Nature Physics 14(12), 1193-1198 (2018).

3. Gumbiner, B.: Cell adhesion: the molecular basis of tissue architecture and morphogenesis. Cell 84(3), 345-357 (1996).
4. Yim, M., et al..: Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robotics Automation Magazine 14(1), 43-52 (2007)
5. Phoneko, S., Mlot, N., Monaenkova, D., Hu, D.L. and Tovey, C.: Fire ants perpetually rebuild sinking towers. Royal Society open science. 4(7), 170475 (2017).
6. Reid, C. et al.: Army ants dynamically adjust living bridges in response to a cost–benefit trade-off. Proc. of the Natl. Academy of Sciences 112(49), (2015)
7. Lioni, A., Sauwens, C., Theraulaz, G., Deneubourg, J.: Chain formation in Oecophylla longinoda. Journal of Insect Behavior 14(5), 679-696 (2001).
8. Werfel, J., Petersen, K. and Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science 343(6172) 754-758 (2014).
9. Gauci, M., Nagpal, R., Rubenstein, M.: Programmable self-disassembly for shape formation in large-scale robot collectives. DARS 2018.
10. Tucci, T., et al.: A Distributed Self-Assembly Planning Algorithm for Modular Robots. AAMAS 2018.
11. Stoy, K., Nagpal, R.: Self-repair through scale independent self-reconfiguration. Intl. Conf. on Intelligent Robots and Systems, pp. 2062-2067). IEEE (2004).
12. Malley, M., et al.: Eciton robotica: Design and algorithms for an adaptive self-assembling soft robot collective. ICRA 2020.
13. Melenbrink, N., et al.: Using local force measurements to guide construction by distributed climbing robots. IROS 2017.
14. Mlot, N., et al.: Fire ants self-assemble into waterproof rafts to survive floods. Proc. of the Natl. Academy of Sciences 108(19), 7669-7673 (2011).
15. Foster, P.C., et al.: Fire ants actively control spacing and orientation within self-assemblages. Journal of Experimental Biology 217(12), 2089-2100 (2014).
16. Jorgensen, M.W., et al.: Modular ATRON: Modules for a self-reconfigurable robot. Intl. Conf. on Intelligent Robots and Systems, pp. 2068-2073. IEEE (2004).
17. Neubert, J., et al.: A robotic module for stochastic fluidic assembly of 3D self-reconfiguring structures. ICRA 2010.
18. Romanishin, J.W., et al.: 3D M-Blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions. ICRA 2015.
19. Shimizu, M., et al.: Adaptive reconfiguration of a modular robot through heterogeneous inter-module connections. ICRA 2008.
20. Mondada, F., et al.: SWARM-BOT: A new distributed robotic concept. Autonomous robots, 17(2-3), pp.193-221 (2004).
21. Swissler, P. and Rubenstein, M.: FireAnt: A modular robot with full-body continuous docks. Intl. Conf. on Robotics and Automation pp. 6812-6817. IEEE (2018).
22. Swissler, P., et al.: FireAnt3D: a 3D self-climbing robot towards non-latticed robotic self-assembly. Intl. Conf. on Intelligent Robots and Systems. IEEE (2020).
23. Espinosa, G., et al.: Using hardware specialization and hierarchy to simplify robotic swarms. Intl. Conf. on Robotics and Automation, pp. 7667-7673. IEEE (2018).
24. Fish, J., et al.: A first course in finite elements. Wiley, West Sussex, England (2007).
25. Ananthasayanam, B.: 3D truss analysis / user interface in FEM, MATLAB Central File Exchange, https://www.mathworks.com/matlabcentral/fileexchange/6832-3d-truss-analysis-user-interface-in-fem, last accessed 2020/10/16.
26. Swissler, P.: DARS_2021_Simulation_Videos, https://www.youtube.com/watch?v=YLXcj7RptPw, last accessed 2021/03/11.