Collective Construction with Robot Swarms

Justin Werfel

Abstract Social insects build large, complex structures, which emerge through the collective actions of many simple agents acting with no centralized control or preplanning. These natural systems inspire the research topic of collective construction, in which the goal is to engineer artificial systems that build in a similar way, with swarms of simple robots producing desired structures. In this chapter I review work on the design and realization of such systems. Robots in these systems act independently, in unknown numbers and with no fixed timing, using only local information and no explicit communication; the system takes a high-level design as input, and is guaranteed to produce a structure matching that design, without requiring the details of the construction process to be specified. Stigmergy (indirect communication through manipulation of a shared environment) and convention (tacit agreement due to the use of a common set of rules shared by all robots) are useful principles for implicit coordination that make these collective behaviors possible. I outline current progress in this area and future directions.

1 Introduction

Termites are capable of extraordinary feats of construction. They build towering mounds several meters high (Figure 1), with architecture that not only reflects a complex layout (with features like gardens, nurseries, and a bewildering network of tunnels) but also performs functions like atmospheric regulation for the colony. The insects responsible for these mounds are millimeter-scale creatures, all acting independently, with no central supervisor directing their activities, no knowledge of what's going on beyond their immediate vicinity; most termites are even blind.

Justin Werfel
Wyss Institute for Biologically Inspired Engineering, Harvard University, Cambridge, Massachusetts, USA, e-mail: justin.werfel@wyss.harvard.edu

Fig. 1 (Left) The author with a termite mound in Namibia. (Center) The termites that build such a mound. (Right) A robot swarm inspired by such swarms of social insects (credit: James McLurkin).

And yet the colony as a whole reliably manages to accomplish the construction of a home that suits its needs.

The human approach to construction is very different. We begin with careful blueprints, making detailed plans for each step of the project. Foremen direct the work, coordinating the activities of the workers and the overall progress of the structure. Unlike nearly every other human activity geared toward producing artifacts, automation is largely absent.

Could we harness the power of the swarm? Imagine a collection of robots, individually capable of only a few simple tasks, that together can build any structure you ask them for—working from a picture or other high-level representation, not needing detailed instructions about what to do or when, and acting independently with no explicit coordination and no predictable timing—such that you are *guaranteed* to get the structure you requested. This is the goal of collective construction.

Here I review work toward this goal, describing how simple robots using specialized building material can assemble user-specified structures from large classes of possibilities, in two and three dimensions. Algorithmically, these approaches can be proven to generate the desired results; practically, they have been demonstrated with hardware prototypes for building both two- and three-dimensional structures. Ultimately, the hope is that such systems will be useful in performing human construction projects—particularly enabling exploration and settlement of places like undersea or extraterrestrial environments, where human presence is difficult or dangerous and no traditional construction process currently exists.

1.1 *Challenges and opportunities with robot swarms*

Several factors make collective construction a particular challenge. These include the limitations of mobile robots, the complications associated with a swarm, and the problem of global-to-local compilation.

The limitations faced by mobile robots are especially significant for robots intended to be simple and expendable:

- *Localization*—In general it's very difficult for robots to determine where they are in any global coordinate system. GPS is expensive, not available in all settings, and not always reliable even when it is available. Odometry (trying to estimate position by integrating estimated velocity) is notoriously unreliable, especially in messy settings like construction sites where wheels are likely to slip or other perturbations may easily occur. And yet if a swarm of robots is trying to build a single structure, all of them need to agree on a common coordinate system, or else their efforts may conflict.
- *Communication*—Establishing and maintaining ad-hoc communication networks is an open research area for mobile robots. Network structure changes as robots move; messages may be dropped; individual robots may lose contact with the rest of the network, or the network can otherwise fragment into multiple disconnected units.
- *Manipulation*—Manipulating physical objects is another major open research area for robots. In controlled settings like factories, the environment can be regulated extensively enough that robots only ever encounter a few predictable situations; but in real-world, unconstrained environments, even sophisticated robots have real difficulty manipulating objects. Of course, for construction tasks, very precise alignment of building materials is likely to be required.

Swarm systems present additional challenges. Algorithms have to be robust to unspecified and potentially variable numbers of robots, which may act with no particular detailed timing or specified order, and some of which may be lost during the course of completing the task. Central coordination or monitoring, even if available, may not be feasible for very large swarms, with a centralized agent acting as a communications bottleneck as well as a single failure point. And individual robots in a swarm will likely lack capabilities available to more sophisticated robots.

Perhaps the most significant challenge is that of connecting the low-level behavior of the individual robots with the high-level behavior of the swarm. Emergent behavior—where the local actions of many simple units collectively give rise to interesting global outcomes—is the hallmark of complex systems [3], and in general is not predictable from studying the components of the system in isolation. In the context of construction, the local-to-global problem is to predict what structure will be built given a set of rules for individual robots to follow. While it may not be possible to predict more than very general attributes of the final structure [39], it is at least straightforward to find out what low-level rules will produce by executing them. By contrast, the global-to-local problem—finding a set of low-level rules that guarantee a particular desired high-level result—has no such straightforward means of addressing it. Nevertheless, the problem can be solved: it is possible to design a set of low-level rules that, together with a high-level structure specification as input, specify individual agent behaviors that will provably generate that structure.

The swarm approach can then carry a number of potential advantages over those using one or a few more sophisticated robots. Because no robot is assigned any particular task or role, the system is robust to the loss of individual robots; many or even most robots can break without preventing the swarm from completing its task. Decentralization removes the likelihood of communication bottlenecks, and the necessity for high reliability and/or long latency in passing messages over long distances between specific senders and recipients. The large number of robots gives the opportunity for massive parallelism and very significant speedups over individual robots. Even the simplicity of the robots can be an advantage: a robot with fewer components or capabilities has less that can go wrong, and so might be less likely to malfunction than a more complex robot. Further, simpler robots could more feasibly be scaled down, a critical consideration for those interested in micro- or nanoscale robots that could be used for familiar futurist goals like assembly and disassembly of artifacts or maintenance within the human body.

1.2 Key tools from nature

Two principles used by social insects, *convention* and *stigmergy*, make it possible to realize the goals of collective construction.

Social convention lets all agents (insects or robots) be assured that the other members of the swarm follow the same rules as themselves. This guarantee lets them take certain matters for granted without having to establish them through explicit communication. An analogy from everyday experience is the convention of all cars driving on the same side of the road; it's not necessary to discuss it or to expect to encounter exceptions. The result is in general needing to be able to deal not with any conceivable situation, but only with those that can arise from the shared conventions; exceptions arising from occasional errors can be dealt with as rare special cases.

Stigmergy refers to storing information in the environment, which acts as an indirect form of communication. An analogy again from driving is the street signs that tell drivers where they are, giving street names and numbers that identify the current location. Termites use stigmergy in construction by leaving building material and chemical pheromones in the environment, which can evoke certain responses in other termites that come to that location later; for instance, a deposit of material tends to lead to the deposit of additional material to accompany it, giving a positive feedback process that contributes to the formation of pillars in the nest [11]. Robots can use stigmergy by looking at the local configuration of building material and using that to determine whether and where to add additional material [39]. Or, in *extended stigmergy*, more sophisticated building material can be used to store additional information, such as location in a shared coordinate system, or even directions to reach a place where more material is needed [50, 52]. This idea is elaborated in §2.2–2.4.

More broadly, exploiting the environment provides the key to overcoming the limitations of mobile robots discussed above. Localization need not be solved in general when precise location information is needed only in the immediate vicinity of the structure, and the structure itself can provide positioning cues. Explicit communication can be forgone entirely, with all communication taking place purely implicitly, through modification of the shared environment. Building material fabri-

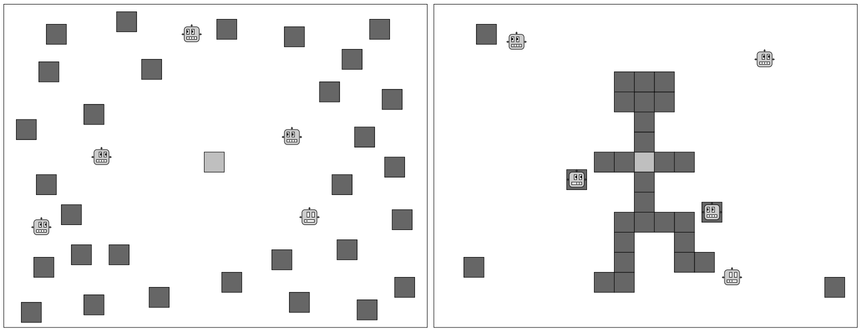

Fig. 2 In the basic problem discussed here, mobile robots collect square building blocks and use them to build some specified structure starting from a "seed" block (lighter shading).

cated with self-aligning features can be responsible for ensuring precise alignment, with robots needing only to move the material to its approximate destination and push it roughly into place. In general, it can be much more effective to take advantage of environmental elements than to try to improve the agents in a complex system directly [46].

2 Case study: two-dimensional solid structures

This section discusses a system where an arbitrary number of robots build user-specified two-dimensional structures out of identical blocks. The idea is for robots deployed in an obstacle-free workspace to collect square blocks and use them to build a specific structure requested by a user, who provides only a high-level description of the desired final result (Figure 2). The discussion covers the setup, assumptions, class of admissible structures, algorithms used, and proof-of-concept hardware prototypes. Section 3 will discuss extensions to cases such as multiple types of materials, adaptive building according to environmental conditions, and three-dimensional structures.

Appropriate assumptions made about component capabilities in the model are critical when the time comes for translating the approach from theory to practice (§2.5). *Robots* are assumed to have the following capabilities: move in any direction in the plane, avoiding collisions; find, pick up, and carry free blocks in the environment; find the structure in progress, and follow its perimeter; and attach blocks to the structure, either along a flat wall or in a corner (Figure 3). Additionally, robots should have either a compass, or the ability to identify at least one unique landmark in the structure (e.g., a colored edge of a specialized block indicating where construction is supposed to begin). These simple capabilities have been demonstrated in a variety of autonomous robotic systems.

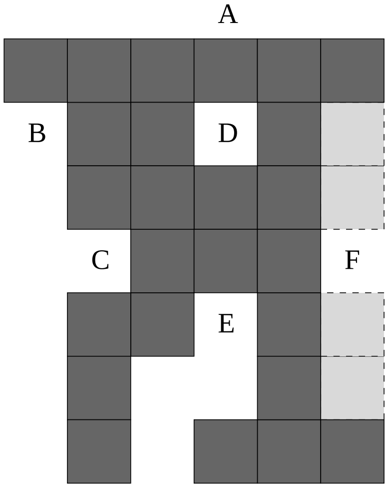

Fig. 3 Physical constraints on block placement. A block can be attached along flat walls (A) or in corners (B), but cannot be maneuvered into a constrained site like C. Preventing configurations like the latter also prevents more complicated situations like D and E. The "separation rule" says that no two blocks can be attached in the same row or column if all sites between them are supposed to be occupied (as along the right side), or else as additional blocks are added (light shading) an unfillable gap will eventually result (F).

Blocks are square and can be attached to each other on all four sides. Some form of self-aligning connectors can enable precise alignment without requiring robots to be responsible for that precision. A single block acts as the "seed" from which the structure grows; this block may, if desired, be specialized in various ways (e.g., containing a beacon to help robots find it in the workspace, having one distinct edge that can act as a unique landmark, etc.).

Figure 3 illustrates the conservative assumption about the physical constraints on block placement: a space one block wide directly between two blocks is too constrained to require robots to be able to maneuver and attach another block there. This assumption makes a physical realization of the system much easier. Moreover, preventing such configurations from occurring has the useful result of also preventing more complicated situations, where a block might have to be maneuvered down a long narrow tunnel, or where a site might become entirely closed off and inaccessible.

The assumption about one-block-wide gaps being unfillable leads directly to the "separation rule": two blocks must never be attached in the same row or column if all sites between them are ultimately intended to be occupied by other blocks. Otherwise, while it may be possible to continue to add blocks for some time, eventually an unfillable gap will result (Figure 3F).

2.1 A high-level approach

One issue in building a particular desired structure is being able to determine whether a given site should ultimately be occupied by a block, or left empty for-ever. Attaching blocks at all of the former sites and none of the latter is synonymous with successful completion of the desired structure.

A further issue is whether a block should be attached at a given site not just eventually, but right now: if blocks are attached indiscriminately without appropriate care for where others are already present, unfillable gaps can easily result, making it impossible to complete the desired structure.

It can be shown that attaching blocks freely at any sites that are supposed to be occupied, while obeying the separation rule, will *provably* lead to successful completion of any two-dimensional structure that satisfies these criteria: (1) it should be solid—the proof does not hold for structures that enclose internal spaces—and (2) because the approach depends on robots following the perimeter of the structure, any alleys in the structure need to be wide enough to permit this perimeter-following. The extent to which the latter restriction limits what structures can be built will depend on the hardware implementation.

The proof [43] runs along the following lines: (1) The structure will be built in such a way that no partially completed stage can physically restrict robots from reaching sites meant to be occupied. This is because the only ways for a site to become physically inaccessible involve breaking the separation rule or building an alley too narrow for robots to travel down, both of which are forbidden by hypothesis. (2) Deadlock—a state where robots could physically proceed with construction, but are prevented from doing so because at every site where they could potentially attach a block, attachment is forbidden by the separation rule—will never occur. This can be demonstrated by contradiction; in any situation where attachment at a given site is forbidden by the separation rule, it can be shown that another site must exist where attachment is allowed, and hence deadlock has not occurred.

Thus any structure from the permitted class will successfully be built, if robots can answer the two following questions for any site where they consider attaching a block: (1) Is this site supposed to be occupied in the final structure? (2) Are there separated blocks already present in the same row or column as this site, such that all the intervening sites are supposed to be occupied? Both of these questions are relatively high-level; in particular, both require obtaining nonlocal information.

The two questions can be addressed in a variety of different possible ways. One way to characterize different approaches is in how the sophistication of the system is apportioned between robots and building blocks. I begin with an example in which blocks are capable of passively storing information (§2.2), and later discuss alternative approaches using simpler or more complex blocks (§2.3).

2.2 A low-level approach with **writable blocks**

As stated earlier, a human using a robot construction system as envisioned should be able to specify a desired structure using some high-level representation, designating where building material should end up without requiring any information about how it gets there. Call this representation the *shape map*: a description that specifies, in some coordinate system, which sites should ultimately be occupied and which left empty. Robots can all be given a copy of this shape map; if they can then determine a site's location in a single shared coordinate system—generally not an easy task for mobile robots—then the first question about the site (whether it should be occupied in the final structure) can be trivially answered by consulting the shape map.

Stigmergy can solve the problem of localization. If robots are able to write information to blocks, then each one can serve as a unique landmark. The analogy to street numbers is apt: the landmarks need not be arbitrary and unorganized, but can stand in a predictable relationship to each other. Robots building with square blocks assemble a coordinate system as they go, and as each block is added, it can be marked with its coordinates in that single common reference frame. In this way, robots around the perimeter of the structure can always determine their location; robots further out in the workspace have no way of knowing their exact position, but neither do they need to.

One way of making blocks writable is with the use of RFID (radio frequency identification) tags. These circuits can cost on the order of pennies, measure on the order of centimeters, and store on the order of kilobytes rewritably and indefinitely without requiring a power source. A robot's transceiver provides the necessary energy to power the tag, enough for it to process a request and send back a reply.

Writable blocks thus make it possible to determine whether a given site should eventually be occupied. Next, what about enforcing the separation rule? Doing so involves information that robots can gather directly themselves. A robot following the perimeter can check the entire length of a potential row of blocks to make sure no blocks are yet present in that row. The use of shared conventions now comes into play: if all robots follow the structure perimeter in the same direction (say counterclockwise), and only start building a new row of blocks beginning at the counterclockwise end, then they can be sure when starting a new row that no conflicting blocks are already present elsewhere in the row nor will such blocks be added by other robots.

Algorithm 1 gives a low-level set of instructions for robots to follow that will provably [43] result in correct completion of the desired structure, without unfillable gaps or deadlocks, regardless of the number of robots or the order or timing of their actions. Intuitively, robots follow the perimeter and either continue building an existing row of blocks, or start a new row after checking that no blocks are already present in that row. The structure is built up by layers, on all sides at once (Figure 4).

2.3 *Building with* **inert** *or* **communicating blocks**

The above scheme can be adapted to the case where blocks do not have the ability to store information, so that the construction materials are in effect identical bricks. The simplest way to achieve localization in this case of *inert blocks* is to make one side of the seed block distinct, so that it provides a single unique landmark, and

Algorithm 1 Pseudocode procedure for assembly of a structure of writable blocks. An ‘end-of-row’ site is one where the robot is either about to turn a corner to the left, or the site directly ahead is not supposed to have a block according to the shape map.

```
while structure not complete do
  fetch new block
  go to structure
  read position from neighboring label
  row-ok ← false
  while still holding block do
    if (site should have a block) and
        ((site just ahead has a block) or
        (row-ok and (at end-of-row))) then
      attach block here
      write coordinates to that block
    else
      if at end-of-row then
        row-ok ← true
      end if
      follow perimeter counterclockwise
    end if
  end while
end while
```

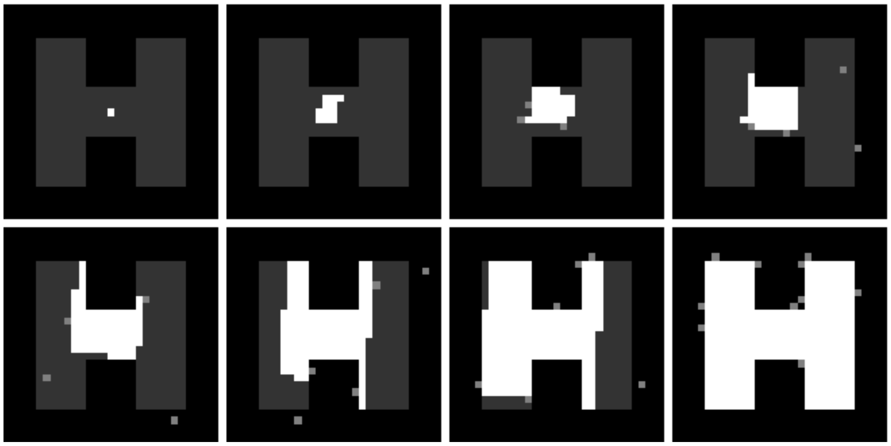

Fig. 4 Simulated construction of a sample structure of writable blocks, showing successive snapshots during the process of construction by ten robots. White: blocks; light gray: robots carrying blocks; dark gray: empty cells where blocks should be attached; black: empty cells that should be left empty.

to have the shape map specify the seed’s location such that the marked edge lies along an edge of the final desired structure. Then robots following the perimeter of the structure at any stage of completion will eventually encounter that landmark, giving them initial location information.[1] Thereafter, they can keep track of their position as they move along the perimeter, by reference to the structure: as before, square blocks embody a coordinate system; by noting the edges of blocks as they pass them, a robot can update its location in this coordinate system as it goes. The separation rule can be enforced by having robots run Algorithm 1 exactly as in the case of writable blocks.

Rather than simpler blocks, we can consider more complex ones. Computation is cheap nowadays; we can envision putting a processor into every block, with the physical connections between blocks being the basis for a data line, so that blocks can reliably and unambiguously communicate with their physically attached neighbors. In this case of *communicating blocks*, the blocks rather than the robots can be responsible for determining which potential attachment sites are valid. Blocks store the shape map and enforce the separation rule (details in [43, 48]); robots only need to bring new blocks to the structure and follow its perimeter, until the structure itself indicates to them that attachment is permitted.

2.4 Extended stigmergy

In the most basic use of stigmergy in such a system, robots building with the simplest (inert) blocks use cues based on the configuration of already-present building material in order to determine where to add more material. The idea of *extended stigmergy* is that increasing the capabilities of environmental elements (in this case, building blocks) can improve the performance of a system more easily and effectively than trying to increase the capabilities of the robots. For instance, equipping the blocks with self-aligning connectors is far more feasible than making simple mobile robots capable of very precise manipulation.

Upgrading inert blocks to writable or communicating ones lets them take over some of the building responsibilities from the robots. With inert blocks, robots keep track of their location, store the shape map, enforce the separation rule, and transport blocks. With writable blocks, the location information is stored instead in the environment. With communicating blocks, the structure is responsible also for the shape map and the separation rule, and robots do nothing but move blocks around.[2]

These different variants can be compared quantitatively, in terms of measures like cost and performance. More complicated blocks will be more expensive than simpler ones. They might not be much more expensive—RFID tags, as discussed, cost as little as a few cents each. Processors for each block would increase the cost more, but how significantly they would do so depends on the application: for a terrestrial construction project with tens of thousands of bricks, adding a computer to

---

[1] The seed block can even be identical to all other blocks, so that no such distinct landmark is required, if robots have a compass (details in [43]).

[2] One could imagine moving even further down this path and making the blocks responsible also for their own movement—that is, erasing the distinction between robots and blocks, and building the structure out of robots directly. This idea will be discussed in §4.

Block algorithm	Best case	Worst case	Average case
Inert	$2n^3-n-1$	$5n^3+3n^2-4n-5$	$\sim(2.99\pm0.04)n^{2.986\pm0.003}$
Writable	n^2+n-1	$O(n^3)$	$\sim(0.99\pm0.06)n^{2.954\pm0.014}$
Communicating	0	$O(n^3)$	$\sim(1.3\pm0.4)n^{2.56\pm0.08}$

Table 1 Summary of best, worst, and average cases for total distance robots travel along the structure perimeter while building an $n\times n$-block square. Average-case results are based on simulation experiments with values of n from 10 to 200.

each one would be prohibitive; for construction in settings like outer space or underwater, where building materials are likely to be highly specialized and expensive to begin with, a processor may represent an insignificant additional cost.

More capable blocks can lead to significantly improved performance, in a number of respects. One measure of performance is the total distance robots need to travel in the course of building a given structure, which is closely related to the total time required. Because the distance robots travel to fetch new blocks and bring them to the structure will be roughly the same regardless of the kind of blocks used, it makes sense to focus on the distance robots travel along the perimeter of the structure, searching for valid attachment sites. The best and worst case can be calculated for a given structure, and the average case determined experimentally. For square structures of $n\times n$ blocks, Table 1 shows that more capable blocks allow for much more efficient construction [43, 50].

Another measure of performance is the opportunity to exploit the parallelism of the swarm [50]. At any given moment, there will be some number of simultaneously eligible sites where robots could attach blocks. The larger this number, the more tasks that different robots could be performing at any given time, so that a larger swarm can potentially be more useful. With inert blocks, because of the need to find the single landmark before attaching a block, a structure grows in such a way that only one site is eligible for attachment at a time. With writable blocks, work can take place simultaneously on one row along each edge of the structure, which can mean many sites available at once for complicated shapes. With communicating blocks, the structure can grow in a still less constrained way, such that many more sites are typically available; moreover, unlike the cases with noncommunicating blocks, the number of such sites increases with the size of the structure. The result is that when building with communicating blocks, a larger construction task can be effectively addressed by putting a larger swarm to work on it.

2.5 Hardware implementations

To demonstrate proof of concept for this approach to automated construction of solid 2D shapes, prototype systems with inert and writable blocks have been constructed. The first [43, 48] (Figure 5) is based on Evolution Robotics's ER1 platform, with

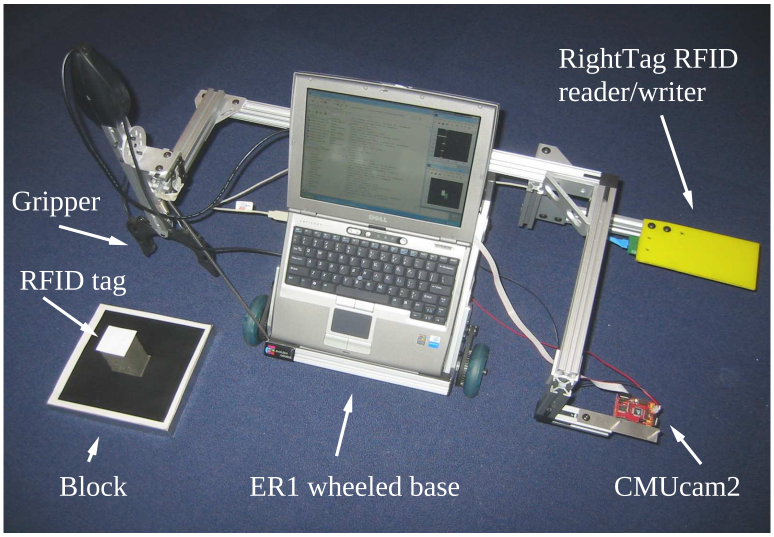

Fig. 5 ER1-based system (robot and block).

a laptop computer driving a wheeled base with a gripper. A CMUcam2 mounted pointing downward gives visual feedback about objects forward and to the left of the robot; a RightTag RFID read/write transceiver allows interaction with tags. Blocks are 8.5” square, made from sheet metal, with neodymium magnets used to achieve self-alignment and attachment; a foam column on top acts as both a handle for the gripper to grasp and a mounting point for RFID tags.

This system demonstrates the key elements of the approaches with noncommunicating blocks: the ability to maneuver to a cache, pick up a block and bring it to the structure; perimeter-following; recognition of block edges and sites where block attachment is valid; attachment of blocks at desired locations; and, for writable blocks, the ability to read and write coordinates. A user is able to specify a desired 2D structure as a bitmap, and the robot can build that structure without needing further instructions or intervention (Figure 6).

While this first system demonstrates the feasibility of the approach, it could make its point more strongly in some respects. As a system with a single robot, it lacks something as a convincing demonstration of a swarm. Morever, while the robot’s capabilities and behavior are fairly simple, it gives the appearance of substantial complexity due to its laptop and camera.

A second system addressing these shortcomings uses LEGO Mindstorms [33] (Figures 7, 8). The approach, component design, and robot behavior are very similar to the first system. The camera is replaced by three light sensors; the computer is replaced by two Mindstorms RCX control units. Blocks are made from polystyrene foam to reduce their weight, and marked with a pattern of black rectangles as visual references for the robots. This system builds only with inert blocks, and constructs

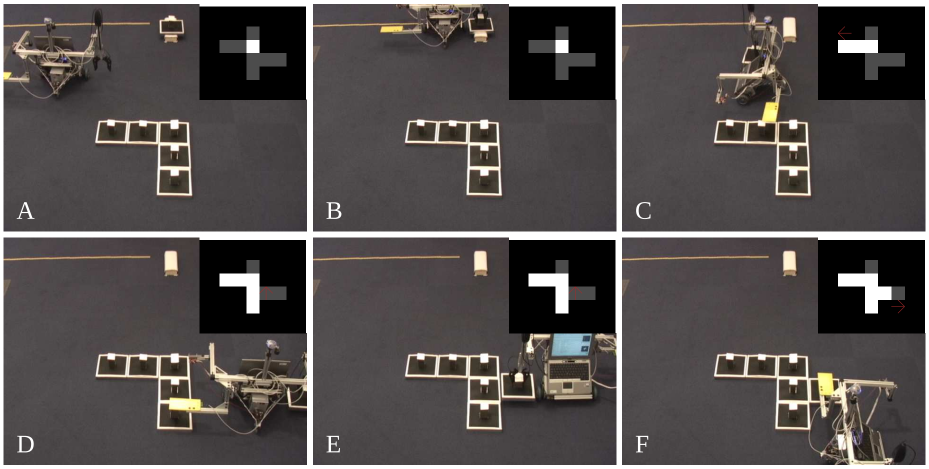

Fig. 6 Process of adding one block to the structure, using writable blocks. The cache is at top, structure in progress at bottom, with the seed block at its upper right. Inset: the robot's knowledge about the structure's progress and its own position: desired structure in gray, known existing blocks in white, robot location (if known) shown as arrow.
(A) The robot, traveling toward the cache from the vicinity of the structure, initially knows only that the seed block must be present.
(B) Using the line on the floor as a reference, it maneuvers to and picks up a block from the cache.
(C) Once at the structure, it can use its RFID reader to determine its position, and its camera to follow the perimeter. Existing blocks can be added to the robot's map as it observes them.
(D) Eventually the robot reaches an empty site where a block is desired, and where one may be attached according to Algorithm 1.
(E) It maneuvers to attach its block at that site, dropping it into place...
(F) ...and writes the block's new coordinates to its tag.

solid two-dimensional sheets rather than arbitrary user-specified structures. Robots operate within a black workspace bounded by a white border, with the structure at its center (Figure 8). Free blocks are placed around the edges of the workspace; after attaching a block, a robot heads straight outwards until it reaches the border, which it then follows in order to find its next block.

This LEGO system emphasizes that very simple robots can achieve the necessary functions. Moreover, it lets a robot be duplicated easily and inexpensively. The resulting two-robot system, while still very small from the perspective of swarm robotics, takes a critical step beyond the one-robot system: robots now must operate in a dynamic rather than static environment. As a result, they encounter changes to the environment besides those they make themselves; and they may encounter each other, and need to be able to resolve such encounters. A set of bump sensors [32] lets the robots detect collisions and react appropriately.

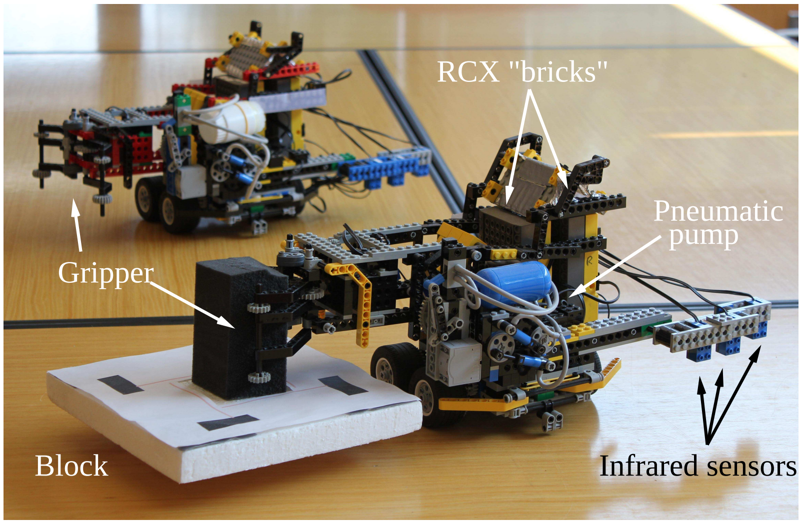

Fig. 7 LEGO-based system (robots and block).

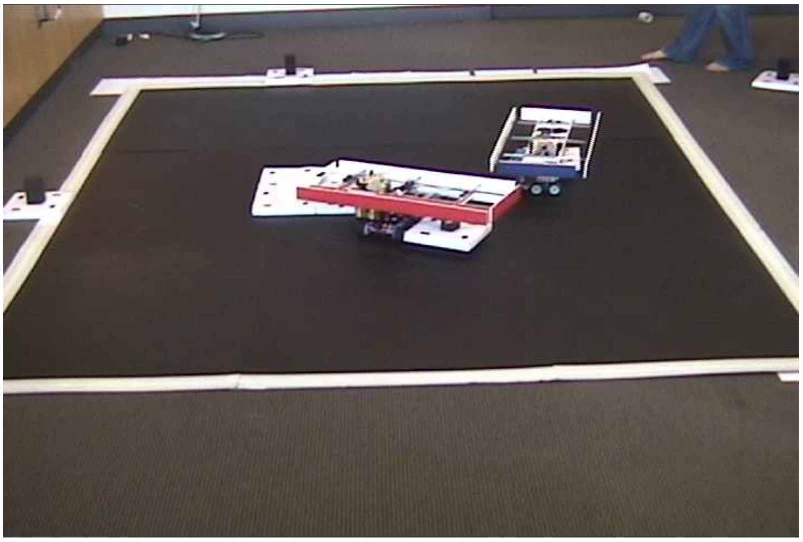

Fig. 8 Bump sensors let two robots be simultaneously active in the same workspace.

3 Extensions

A system capable of building only two-dimensional solid structures with a single kind of material would be of limited use in real construction applications. This section outlines extensions of the system towards more general structures of greater interest.

3.1 Multiple materials

Modern human structures are generally not built from bricks alone, but from components of a variety of materials and form factors—wooden planks, metal beams, drywall sheets, etc. The approach described above does not transfer directly to the use of such varied components. However, in keeping with the idea of increasing the sophistication of environmental elements, one can imagine building not with low-level materials like these but with higher-level units—e.g., prefabricated sections of wall with plumbing and electrical wiring already incorporated into them, or even entire rooms built in a factory and then transported to a construction site and assembled into a building there. The latter approach is already used in construction projects in many countries, not only allowing significant savings over traditional methods in total cost and construction time but even winning design awards for the results [2].

With higher-level blocks of such a kind, the approaches described above can be applied directly. The rules about whether a given block can be attached at a given site now check not only whether a block is ultimately supposed to be located there and whether the separation rule is satisfied, but whether a block of that particular type can be accommodated at that location. The latter question can be handled in different ways. The most straightforward is for the shape map to specify a block type for each site in the desired structure, providing a full blueprint. Alternatively, the choice of blocks for a given site may be less constrained, allowing for adaptive building of structures that are not fully prespecified, as discussed in the next section.

3.2 Adaptive structures

In some cases, an application may not require a structure to be completely determined in advance, but instead can allow a certain amount of dynamic flexibility during the construction process. Two types of flexibility in how structures can be built are *functional adaptivity* and *shape adaptivity*.

Functional adaptivity [47] applies in cases with multiple types of blocks. Consider an example of building a temporary research station underwater, with high-level “blocks” that are complete rooms of various functional types: living quarters, laboratory space, etc. The precise layout of the different room types within the station may not be important; instead, the designers want to specify constraints on their relative locations: all living units should be located in a contiguous block, no unit should be more than three hops from an escape pod, etc. In such a case, the system can be given that set of constraints rather than a full blueprint (Figure 9). Differ-

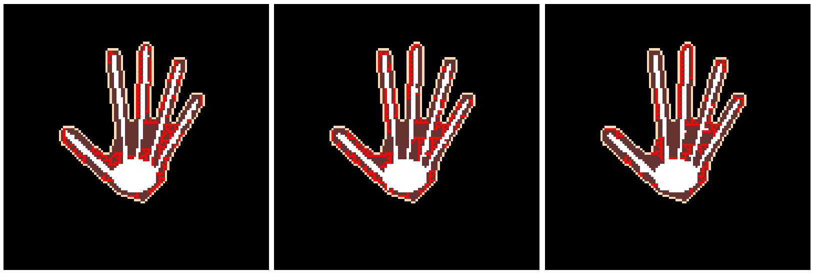

Fig. 9 Specifying functional constraints on the relative locations of multiple block types, rather than full blueprints, allows structures to be built adaptively, with the details varying from one instance to the next. In this example, one fixed set of functional constraints gives rise to variability in the placement of "blood" and "bone" blocks.

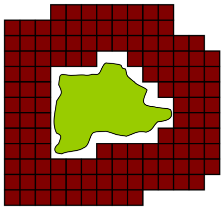

Fig. 10 The shape of a structure (e.g., a protective barrier) can be determined by environmental features (e.g., a hazardous waste spill) rather than specified in advance.

ent types of constraints can be enforced with varying levels of ease by writable or communicating blocks. Satisfying constraints dynamically during the building process in this way can lead to substantially increased efficiency and reduced construction time, compared to fully prespecified designs. Adaptive construction can also let structures be built in ways that respond to environmental conditions unknown in advance.

Shape adaptivity [49] builds on this latter point: the environment can affect the shape in which a structure is built. For instance, immovable obstacles may make it impossible to complete a structure as it was originally intended, so that the best option is to build as much of the prespecified structure as is physically possible and give up on the rest. In more extreme cases, the shape of the final structure may be entirely defined by environmental elements. For instance, a team of robots may be tasked with building a protective barrier of a given thickness around a hazardous chemical spill. Only the thickness of the barrier is set in advance; the shape is determined by that of the spill (Figure 10).

Adaptive structures more closely address the building problem solved by real termites. Insects do not have the goal of producing a particular prespecified structure; rather, they build some structure which satisfies the needs of the colony subject to the environment in which it is located. The use of adaptive building also raises the possibility of dynamic structures whose form changes after initial completion and during use, for ongoing maintenance as needed or to adjust to changing conditions.

3.3 More general shapes

One way to ease the restriction on what structures can be built is to require greater capabilities of the robots. The reason for the above limitation to structures without holes and with potentially wide alleys is the assumption that robots cannot attach one block directly between two others (Figure 3). Suppose we require instead that robots must be able to transport blocks to, and attach them at, any physically reachable location—that is, a site becomes inaccessible only if it is completely closed off, with no tunnel from outside the structure leading to it (e.g., site D in 3 is inaccessible, but E can be reached). This requirement makes things much more difficult when it comes to implementing these robots and building material in real life—robots have to be able to carry material down narrow, winding tunnels, and blocks might need to be compressible in order to ensure they can fit down such passages and still reliably be attached at any grid site. However, under these assumptions, any structure in two dimensions can be built, without restrictions as to its shape [49]. There is thus a tradeoff involving the expressivity of the system (in terms of the limitations on the class of structures that can be built) and the sophistication of the capabilities demanded of the components (with the associated challenge of implementing them).

3.4 Three dimensions

The major physical challenge in moving from two to three dimensions is dealing with gravity. As a simplifying first step, then, consider construction not in terrestrial settings but in the underwater or outer-space environments highlighted early in this chapter as places where automated construction systems will be of particular value. These environments, of course, have the attractive feature of weightlessness, allowing us to ignore the problem of gravity at first.

3.4.1 Weightless environments

Sidestepping the physical challenge of gravity for now still leaves us with an algorithmic challenge: how to program robots (and potentially blocks) such that they will provably build arbitrary user-specified structures, working from a three-dimensional shape map. This problem can be decomposed into two subproblems: (1) determining where block attachment is allowed at any partial stage of completion; (2) transporting blocks to those allowed sites. We consider each in turn.

(1) As before, the shape map specifies which sites in a cubic grid should ultimately be occupied or left empty. The problem of finding a partial ordering on block attachment that will provably let the desired structure be built, however, is more complicated in three dimensions. Now the separation rule is not sufficient to ensure that construction cannot get stalled (Figure 11A). Adding a second constraint, the

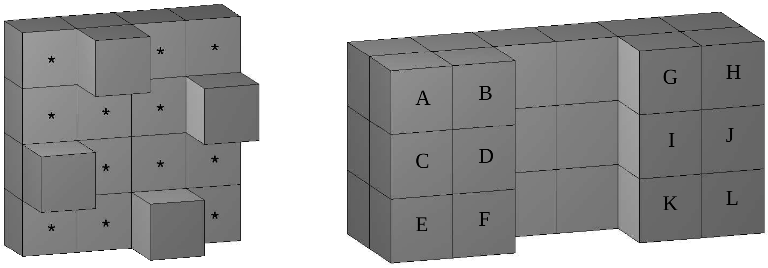

Fig. 11 In three dimensions, obeying the separation rule alone can result in situations where no further blocks can legally be attached. The example on the left shows a structure that has been built while obeying the separation rule, where no additional blocks can now be attached at the starred sites without violating that rule. These situations can be prevented by adding the "plane rule": in any planar slice of the shape map, blocks meant to be part of a contiguous group can only be attached contiguously with already-present blocks in the same group. For instance, the twelve labeled blocks in the example structure on the right form two contiguous groups in the same planar slice. In building this structure, if block A were the first in this plane to be attached, attaching block F next would violate the plane rule, because the two are in the same group but not contiguous; attaching block B or C would be allowed, because of their contiguity to the already-present A; attaching any of G through L would be allowed, because those are in a different group from A.

Fig. 12 Snapshots from a simulation of ten robots building a prespecified structure in three dimensions with two types of building blocks. Background image NASA/courtesy of nasaimages.org.

"plane rule" (a higher-dimensional analogue of the separation rule, Figure 11B), is enough to guarantee the correct completion of any admissible structure [52].

Each candidate attachment site is associated with one plane and two rows in which the presence of other blocks affects whether attachment is allowed. Collecting the corresponding nonlocal information, and ensuring it remains current while other robots act, can be a real challenge for robots; however, it is a simple matter for blocks [52]. Thus this framework lends itself best to the use of communicating blocks. In outer space or underwater applications, the building materials used are likely to be highly specialized, so that communicating blocks should be economically feasible as discussed earlier.

(2) Finding the valid attachment sites is a harder task for robots searching in three dimensions than in two. For two-dimensional structures, a stateless robot can start at any point on the structure perimeter, follow it counterclockwise, and be guaranteed to visit every candidate attachment site exactly once. For three-dimensional structures, in general no such path along the structure surface exists [44]. As a result, robots must generally either revisit previously rejected sites, or rely on the blocks to tell them not only where attachment is allowed but also which way to go in order to reach an allowed attachment site. A tradeoff again exists, this time between unnecessary robot movement and amount of communication required between blocks in the structure [52].

3.4.2 Terrestrial environments

When gravity is a factor, hardware challenges increase considerably: robots need to be able to reach higher sections of a large structure (for instance, by climbing on the structure in progress), and intermediate as well as final structures need to hold together against their weight (and that of any robots climbing on them) without falling or breaking apart. More than ever, careful consideration needs to be given to robot and block capabilities from the beginning, if a physical implementation is to be feasible. And again, there is a tradeoff between the expressivity of the system and the ease of implementing it.

A realistic starting point is to limit buildable structures to those where each block is supported by a stack of other blocks down to the ground, and where each stack of blocks has at least one adjacent stack whose height differs from its own by at most 1. The first restriction ensures that the system need not deal with cantilevers and other overhanging features; the second means that robots only need to climb a height of one block at a time.

These limitations make the problem more tractable, and a physical system for autonomous construction of such structures has been engineered [31]. Careful design of the robots and blocks allows a robot to perform key functions (like climbing on and adding blocks to the structure) with high reliability, as is necessary for successful completion of large-scale structures without human intervention. Figure 13 shows one robot autonomously building a structure 18 times its own volume.

Distributed control algorithms in the style of those described above for 2D systems allow an arbitrary number of robots to provably build user-specified structures from a large class of possibilities, even when using purely inert blocks [53]. Adopting shared conventions about travel again lets robots rely on local information only,

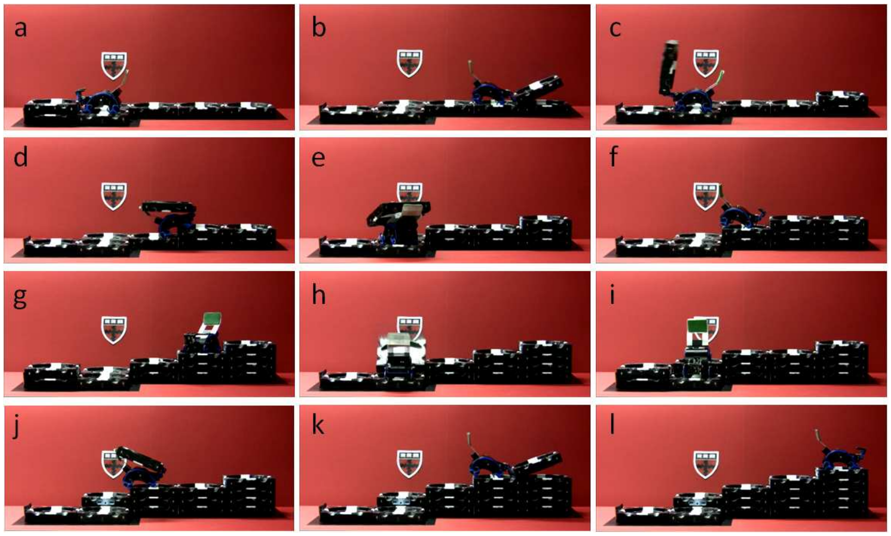

Fig. 13 Snapshots of a climbing robot building a 10-block 3D structure, collecting new blocks from a (manually reloaded) station at left.

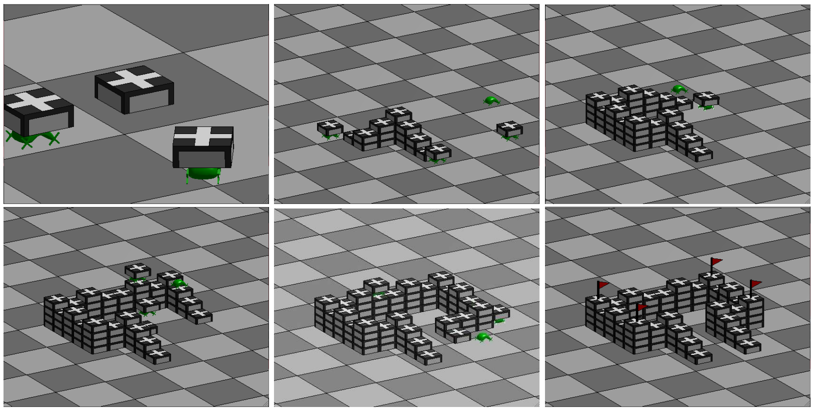

Fig. 14 Snapshots from a simulation of five climbing robots like that in Figure 13, building a prespecified 3D structure.

acting independently and without explicit communication, and still be sure of building the desired target structure (Figure 14).

4 Related work

The work reviewed in this chapter has focused primarily on algorithmic global-to-local problems within collective construction: how to program robots so that they provably build user-specified structures, subject to conservative assumptions about their physical movement and other capabilities. Other work in collective construction often focuses on other important aspects of the problem. Some studies are concerned with the hardware design of robots and building material [5, 8, 17, 37, 38]. Others focus on the use of communication between robots to improve the performance of a system [19, 41, 57], or on cooperation in teams of sophisticated, often heterogenous robots [34, 36]. Some construction-related studies do not have the goal of building any specific target structure [27, 30]. In other cases where building a particular structure is the goal, the user may need to provide a full sequence of low-level building steps as input [19, 41]. Other studies do provide methods for automated construction from high-level specifications, sometimes under assumptions that may be difficult to realize in physical robotic swarms, such as high-accuracy localization [24, 42, 57], fully centralized control [24, 56], or the ability for robots to pass through building material [26].

A number of studies consider building truss structures [9, 17, 24, 29, 56, 57]. One reason trusses are of special interest is because of their physical openness: a robot may be able to maneuver through a truss structure where a solid one would block its movement. Appropriately designed hardware may also allow truss struts to be inserted directly between already-present elements, easing the restrictions on material placement [9, 17]. The approaches described in this chapter can be applied directly to the less-restrictive task of building truss structures [43], although approaches designed for the latter case may not be applicable without modification for systems with more restrictive movement constraints.

A closely related area of inquiry looks at local-to-global questions. These studies, often more specifically concerned with modeling insect systems, start with low-level rules and examine the structures that result from their execution [4, 20, 39, 40]. Such systems use the classic notion of stigmergy in the sense of local configurations of material triggering agent actions. Evolutionary algorithms can also be used to try to find low-level rulesets that generate particular desired high-level structures [16, 25].

Alternative approaches to automating construction other than the use of swarms of mobile robots have been proposed. “Contour crafting” [22] and similar approaches [6] involve erecting a gantry above a desired building site, and extruding and otherwise manipulating material to form a desired structure, in the equivalent of large-scale rapid prototyping. An automated “factory floor” [9, 29] tiled with robotic arms could manipulate material to build a structure one layer at a time, lifting it upward after the completion of each layer to work on the next, in order to construct three-dimensional artifacts without requiring robots to be mobile.

The research area of programmed self-assembly [1, 12, 13, 16, 18, 21, 23, 51, 54] is intimately related to that of collective construction. Programmed self-assembly is the problem of designing a collection of elements with (potentially dynamic) edge binding properties such that, when mixed randomly, they bind to form desired results. Its focus is typically on the microscale, where thermal energy or ambient fluid forces provide the random movement. However, algorithms for self-assembly may potentially be applied directly to construction scenarios (with robots providing the movement force) or vice versa. In particular, self-assembling tiles that can change binding properties and communicate with attached neighbors are effectively identical to communicating blocks.

One step further is the area of self-reconfigurable modular robotics [35, 55], in which units can move under their own power, subject to some set of motion constraints (e.g., always remaining connected to the group, etc.). In effect, the distinction between blocks and robots is removed, and the structure is built out of robots. While this may be considered a possible approach to the construction problem, it is unlikely to be an ideal one in practice. Units required to act as both robots and blocks will not be optimal for either role: the capacity for movement requires expense and complexity that will go to waste once a structure is complete, and will likely interfere with desirable structural properties like strength, insulation, etc. The separation into mobile and structural elements lets each type be specialized for its purpose, increasing the effectiveness and lowering the cost of the overall system. The same separation means that approaches developed for one area are typically not directly applicable to the other, due to differences in the motion constraints for system components. Nevertheless, principles and approaches developed in the context of modular robotics can be valuable in automated construction, in terms of both algorithms [14, 15] and hardware [28].

Finally, the overarching topic of this volume suggests considering engineered morphogenesis as a related area [7, 45]: programming living cells to develop into desired structures, the way an egg grows into an animal. This emerging research area involves constraints and tools significantly different from those of collective construction. For instance, passive blocks cannot be rearranged within a structure once attached, while cells can rearrange their configuration throughout morphogenesis and produce or eliminate units deep within the structure; conversely, while position information is critical in morphogenesis as it is in construction, deformable cells cannot embody a coordinate system as directly and explicitly as rigid blocks. However, as biologically inspired problems of structure formation in engineered distributed systems, the two topics are strongly similar in flavor and goals. Perhaps one day a person looking to build their dream house will have the choice of having a swarm of artificial ants assemble it from parts, or planting a seed and letting it grow directly into that form.

5 Conclusion

In this chapter, I have outlined current progress toward the goal of collective construction: algorithms for robot swarms to build user-specified structures without human intervention, extensions toward structures of greater general interest, and hardware prototypes demonstrating that such systems can in principle be built. Still, we have a long way to go before the day fully automated swarm systems are routinely building artifacts for us. A number of open questions and topics for ongoing research remain:

- The above treatment describes a tradeoff between the capabilities of the robots and the class of structures that they can build. However, this tradeoff is based on known algorithms, not necessarily on possible ones. Can an algorithm be found, for instance, that respects the conservative assumptions about block attachment of Figure 3 yet allows the construction of any structure enclosing arbitrarily-shaped holes, or conversely can it be shown that no such algorithm exists?
- While the use of high-level prefabricated building blocks is the most straightforward way to incorporate multiple building materials into the approaches discussed here, in some cases it may be preferable to work directly with lower-level materials. In the case of elements like pipes and beams, these will involve shapes other than identical squares. How can the approaches of this chapter best be extended to accommodate building blocks of different shapes?
- Robots in this chapter act independently, interacting directly only to the extent of getting out of each other's way; a single robot could complete an entire building project alone. Can explicit cooperation be used to let robot teams accomplish tasks they couldn't manage individually [10, 34, 36]—as in the way ants collectively transport food items too large for them to lift on their own—or for increased efficiency, letting N robots achieve more than an N-fold speedup?
- The topic of adaptive structures is far from exhausted. How can functional constraints involving nonlocal relationships be satisfied by robots building with non-communicating blocks? What other situations exist in which the desired shape of a structure depends on its environment, and how can they be addressed? How can robots modify a structure in an ongoing way in response to a changing environment, as termites modify their mounds over time?
- Developing hardware systems to bring automated construction not just off the page and into the physical world, but out of the lab and into the field, is a major undertaking. It's important that work in theory and in hardware go hand in hand. Real-life considerations are critical to ensure that what's studied in theory is of real-world relevance; in the work reviewed here, careful attention to what's feasible in reality during the initial problem formulation is what has made it possible to later build simple prototypes that work. Conversely, theoretical studies can provide important principles that direct hardware development, as well as quantitative estimates of how useful certain capabilities would be in practice and thus how worthwhile they would be to pursue.

Current and future work in these areas brings us closer to the day when not just this printed volume, but the library in which it is stored, are produced automatically from high-level designs by robotic systems.[3]

---

[3] Or, since you may very well be reading this chapter on your laptop, possibly outdoors: "...not just the computer displaying this electronic document, but the building to which you will eventually return to plug it in".

References

1. Arbuckle, D., Requicha, A.: Active self-assembly. In: Proceedings of 2004 IEEE International Conference on Robotics and Automation, pp. 896–901. New Orleans, Louisiana (2004)
2. Arieff, A., Burkhart, B.: Prefab. Gibbs Smith (2002)
3. Bar-Yam, Y.: Dynamics of Complex Systems. Addison-Wesley (1997)
4. Bonabeau, E., Théraulaz, G., Deneuborg, J.L., Franks, N., Rafelsberger, O., Joly, J.L., Blanco, S.: A model for the emergence of pillars, walls and royal chambers in termite nests. Philosophical Transactions of the Royal Society of London B **353**, 1561–1576 (1998)
5. Bowyer, A.: Automated construction using co-operating biomimetic robots. Tech. rep., University of Bath Department of Mechanical Engineering, Bath, UK (2000)
6. Buswell, R.A., Soar, R.C., Pendlebury, M.C., Gibb, A.G.F., Edum-Fotwe, F.T., Thorpe, A.: Investigation of the potential for applying freeform processes to construction. In: Proceedings of the 3rd International Conference on Innovation in Architecture, Engineering and Construction, pp. 141–150. Rotterdam, The Netherlands (2005)
7. Doursat, R.: Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. In: R.P. Würtz (ed.) Organic Computing, pp. 167–200. Springer-Verlag (2008)
8. Everist, J., Mogharei, K., Suri, H., Ranasinghe, N., Khoshnevis, B., Will, P., Shen, W.M.: A system for in-space assembly. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2356–2361. Sendai, Japan (2004)
9. Galloway, K., Jois, R., Yim, M.: Factory floor: A robotically reconfigurable construction platform. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation. Anchorage, Alaska (2010)
10. Gerkey, B., Matarić, M.: Pusher-watcher: an approach to fault-tolerant tightly-coupled robot coordination. In: Proceedings of 2002 IEEE International Conference on Robotics and Automation, pp. 464–469. Washington, D.C., USA (2002)
11. Grassé, P.P.: La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La théorie de la stigmergie: Essai d'interpretation du comportement des termites constructeurs. Insectes Sociaux **6**, 41–81 (1959)
12. Griffith, S., Goldwater, D., Jacobson, J.: Self-replication from random parts. Nature **437**, 636 (2005)
13. Groß, R., Dorigo, M.: Self-assembly at the macroscopic scale. Proceedings of the IEEE **96**(9), 1490–1508 (2008)
14. Grushin, A., Reggia, J.A.: Automated design of distributed control rules for the self-assembly of prespecified artificial structures. Robotics and Autonomous Systems **56**(4), 334–359 (2008)
15. Grushin, A., Reggia, J.A.: Parsimonious rule generation for a nature-inspired approach to self-assembly. ACM Transactions on Autonomous and Adaptive Systems **5**(3), 12:1–12:24 (2010)
16. Guo, Y., Poulton, G., Valencia, P., James, G.: Designing self-assembly for 2-dimensional building blocks. In: ESOA'03 Workshop. Melbourne, Australia (2003)
17. Hjelle, D.A., Lipson, H.: A robotically reconfigurable truss. In: Proceedings of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots (2009)
18. Jones, C., Matarić, M.: From local to global behavior in intelligent self-assembly. In: Proceedings of 2003 IEEE International Conference on Robotics and Automation, pp. 721–726. Taipei, Taiwan (2003)
19. Jones, C., Matarić, M.: Automatic synthesis of communication-based coordinated multi-robot systems. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 381–387. Sendai, Japan (2004)
20. Karsai, I., Pénzes, Z.: Comb building in social wasps: self-organization and stigmergic script. Journal of Theoretical Biology **161**, 505–525 (1993)
21. Kelly, J., Zhang, H.: Combinatorial optimization of sensing for rule-based planar distributed assembly. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3728–3734. Beijing, China (2006)

22. Khoshnevis, B., Bekey, G.: Automated construction using contour crafting—applications on earth and beyond. Journal of Rapid Prototyping **9**(2), 1–8 (2003)
23. Klavins, E., Ghrist, R., Lipsky, D.: A grammatical approach to self-organizing robotic systems. IEEE Transactions on Automatic Control **51**(6), 949–962 (2006)
24. Lindsey, Q., Mellinger, D., Kumar, V.: Construction of cubic structures with quadrotor teams. In: Proceedings of Robotics: Science and Systems. Los Angeles, CA, USA (2011)
25. von Mammen, S., Jacob, C., Kókai, G.: Evolving swarms that build 3D structures. In: Proceedings of 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1434–1441. Edinburgh, UK (2005)
26. Mason, Z.: Programming with stigmergy: using swarms for construction. In: Proceedings of Artificial Life VIII, pp. 371–374. Sydney, Australia (2002)
27. Melhuish, C., Welsby, J., Edwards, C.: Using templates for defensive wall building with autonomous mobile ant-like robots. In: Proceedings of Towards Intelligent Autonomous Mobile Robots 99. Manchester, UK (1999)
28. Murata, S., Kakomura, K., Kurokawa, H.: Toward a scalable modular robotic system: Navigation, docking, and integration of M-TRAN. IEEE Robotics & Automation Magazine **14**(4), 56–63 (2007)
29. Napp, N., Klavins, E.: Robust by composition: Programs for multi robot systems. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation. Anchorage, Alaska (2010)
30. Parker, C., Zhang, H., Kube, R.: Blind bulldozing: multiple robot nest construction. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas, USA (2003)
31. Petersen, K., Nagpal, R., Werfel, J.: TERMES: An autonomous robotic system for three-dimensional collective construction. In: Proc. Robotics: Science & Systems VII (RSS2011). Los Angeles, USA (2011)
32. Schuil, C.: Collision detection in lego robots. Senior thesis, Engineering Sciences, Harvard University, Cambridge, Massachusetts (2007)
33. Schuil, C., Valente, M., Werfel, J., Nagpal, R.: Collective construction using LEGO robots. In: Robot Exhibition, Twenty-First National Conference on Artificial Intelligence (AAAI 2006). Boston, Massachusetts (2006)
34. Sellner, B., Heger, F.W., Hiatt, L.M., Simmons, R., Singh, S.: Coordinated multi-agent teams and sliding autonomy for large-scale assembly. Proceedings of the IEEE **94**, 1425–1444 (2006)
35. Støy, K., Brandt, D., Christensen, D.J.: Self-Reconfigurable Robots: An Introduction. MIT Press (2010)
36. Stroupe, A., Okon, A., Robinson, M., Huntsberger, T., Aghazarian, H., Baumgartner, E.: Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance. Autonomous Robots **20**, 113–123 (2006)
37. Terada, Y., Murata, S.: Automatic assembly system for a large-scale modular structure: hardware design of module and assembler robot. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2349–2355. Sendai, Japan (2004)
38. Terada, Y., Murata, S.: Automatic modular assembly system and its distributed control. International Journal of Robotics Research **27**(3–4), 445–462 (2008)
39. Théraulaz, G., Bonabeau, E.: Coordination in distributed building. Science **269**, 686–688 (1995)
40. Théraulaz, G., Bonabeau, E.: Modelling the collective building of complex architectures in social insects with lattice swarms. Journal of Theoretical Biology **177**, 381–400 (1995)
41. Wawerla, J., Sukhatme, G., Matarić, M.: Collective construction with multiple robots. In: Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems. Lausanne, Switzerland (2002)
42. Werfel, J.: Building blocks for multi-agent construction. In: Proceedings of Distributed Autonomous Robotic Systems 2004. Toulouse, France (2004)

43. Werfel, J.: Anthills built to order: Automating construction with artificial swarms. PhD dissertation, Massachusetts Institute of Technology, MIT Computer Science and Artificial Intelligence Laboratory (2006)
44. Werfel, J.: Robot search in 3D swarm construction. In: Proceedings of First IEEE International Conference on Self-Adaptive and Self-Organizing Systems, pp. 363–366. Cambridge, Massachusetts, USA (2007)
45. Werfel, J.: Biologically inspired primitives for engineered morphogenesis. In: Proceedings of the 7th International Conference on Swarm Intelligence (ANTS 2010). Brussels, Belgium (2010)
46. Werfel, J., Bar-Yam, Y., Ingber, D.: Bioinspired environmental coordination in spatial computing systems. In: Workshop on Spatial Computing, at Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008). Venice, Italy (2008)
47. Werfel, J., Bar-Yam, Y., Nagpal, R.: Building patterned structures with robot swarms. In: Proceedings of Nineteenth International Joint Conference on Artificial Intelligence. Edinburgh, Scotland (2005)
48. Werfel, J., Bar-Yam, Y., Rus, D., Nagpal, R.: Distributed construction by mobile robots with enhanced building blocks. In: Proceedings of 2006 IEEE International Conference on Robotics and Automation. Orlando, USA (2006)
49. Werfel, J., Ingber, D., Nagpal, R.: Collective construction of environmentally-adaptive structures. In: Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. San Diego, USA (2007)
50. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. IEEE Intelligent Systems **21**(2), 20–28 (2006)
51. Werfel, J., Nagpal, R.: Towards a common comparison framework for global-to-local programming of self-assembling robotic systems. In: Workshop on Self-Reconfigurable Robot Systems and Applications, at IEEE Conference on Intelligent Robots and Systems (2007)
52. Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. International Journal of Robotics Research **27**(3–4), 463–479 (2008)
53. Werfel, J., Petersen, K., Nagpal, R.: Distributed multi-robot algorithms for the TERMES 3D collective construction system. In: Workshop on Reconfigurable Modular Robotics, at 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (2011)
54. White, P., Zykov, V., Bongard, J., Lipson, H.: Three dimensional stochastic reconfiguration of modular robots. In: Proceedings of Robotics: Science and Systems I. Cambridge, MA (2005)
55. Yim, M., Shen, W.M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular self-reconfigurable robot systems: Challenges and opportunities for the future. IEEE Robotics and Automation Magazine **14**(1), 43–52 (2007)
56. Yun, S., Hjelle, D.A., Lipson, H., Rus, D.: Planning the reconfiguration of grounded truss structures with truss climbing robots that carry truss elements. In: Proceedings of 2009 IEEE/RSJ IEEE International Conference on Robotics and Automation. Kobe, Japan (2009)
57. Yun, S., Schwager, M., Rus, D.: Coordinating construction of truss structures using distributed equal-mass partitioning. In: Proc. of the 14th International Symposium on Robotics Research. Luzern, Switzerland (2009)