Swarm Construction Coordinated Through the Building Material

Yating Zheng[1,2], Michael Allwright[2(✉)], Weixu Zhu[2], Majd Kassawat[3], Zhangang Han[1], and Marco Dorigo[2]

[1] School of Systems Science, Beijing Normal University, Beijing, China
zhengyating@mail.bnu.edu.cn, zhan@bnu.edu.cn
[2] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium
{michael.allwright,weixu.zhu,marco.dorigo}@ulb.ac.be
[3] Universidad Jaume I, Castellon, Spain
majd@uji.es

Abstract. This paper demonstrates a swarm robotics construction system where the intelligence that coordinates construction has been moved from the robots to an advanced building material. This building material, that we call Stigmergic Blocks, is capable of computation and local communication. Using comprehensive simulation models based on real hardware, we investigate approaches to improving the efficiency and flexibility of a swarm robotics construction system.

Keywords: Swarm robotics · Stigmergy · Swarm construction · Stigmergic material

1 Introduction

In swarm robotics, groups of robots coordinate their actions by communicating with their neighbors and by sensing and modifying the surrounding environment [5,7]. These interactions between the robots and their environment can result in the emergence of useful collective behaviors. It is the goal of swarm robotics researchers to understand how the individual robots in these swarms can be programmed so that these collective behaviors not only perform a useful task but do so in a way that is generalizable, scalable, and robust to disturbances such as robot failures. If these characteristics can be realized in robot swarms, this approach to robotics may be well suited to automating construction in hostile environments. As an example, environments with excessive radiation are too dangerous for human workers and may result in high failure rates of robots and their supporting positioning and communication infrastructure.

From an abstract perspective, the goal of construction is to arrange materials in an environment into one or more structures with respect to a set of constraints. For example, an ordering that ensures that the structure remains stable during the entire building process. In the case of swarm robotics, these constraints can

M. Baratchi et al. (Eds.): BNAIC/Benelearn 2020, CCIS 1398, pp. 188–202, 2021.
https://doi.org/10.1007/978-3-030-76640-5_12

be realized in terms of reactive rules that instruct robots to perform construction actions in response to environmental stimuli. If these stimuli are defined in terms of the results of previous construction actions by other robots, we say that the robots are coordinating a construction task through stigmergic communication [6,16]. This approach to construction has been used by Jones and Matarić to build 2D structures from colored blocks [8,9] and by Allwright et al. to build a staircase using a single robot and a stepped pyramid using four robots [1,2]. A significant challenge in this approach, however, is finding a set of rules that unambiguously map all intermediate construction states to construction actions. The complexity of these sets of rules increases with the size of the structure and has necessitated the use of offline algorithms to generate rule sets in similar research [11]. Moreover, if we want to take advantage of the potential scalability of swarm robotics systems by building in parallel, this complexity is exacerbated since building in parallel imposes additional constraints on a rule set to guarantee that the structure is always in a valid state [4,15].

To work around these limitations, researchers have supplemented stigmergic communication in a variety of ways. For example, Werfel et al. [17,19] use the concept of extended stigmergy in their work on multi-robot construction. This approach leverages a robot's or a block's ability to localize itself to simplify the construction rules. The work by Sugawara and Doi [13,14] takes another approach and instead has the building materials guide the robots to where building material should be added. In this paper, we extend the work of Sugawara and Doi by further investigating the potential advantages of having a building material coordinate construction in a more capable multi-robot construction system, namely, the one designed by Allwright et al. [3]. This construction system consists of two components, a robot called the BuilderBot and a building material, called the Stigmergic Blocks, which the BuilderBot assembles into structures using its manipulator (see Fig. 1). We have developed plugins that provide comprehensive models of the BuilderBot and the Stigmergic Block for the ARGoS simulator [12] and used them in the experimental work presented in this paper.

The general setup of our construction system involves having the robots use computer vision to identify the configuration of a structure by observing the location of its blocks and the colors of the LEDs on those blocks. The robots then perform construction actions such as attaching another block in response to certain configurations of the structure. In the experiments where we extend the work of Sugawara and Doi, we use the building material's peer-to-peer near-field communication to allow messages to be exchanged between adjacent blocks. By enabling the routing of messages through intermediate blocks, we enable one block to monitor the structure and to communicate directly with the robots by changing the colors of the LEDs on one or more blocks.

The remainder of this paper is organized as follows. In the following section, we describe two classes of construction algorithms that we use to coordinate construction. In Sect. 3, we present three experiments that demonstrate how the efficiency and flexibility of the building process can be improved and how the need to find complex sets of construction rules can be eliminated by enabling the

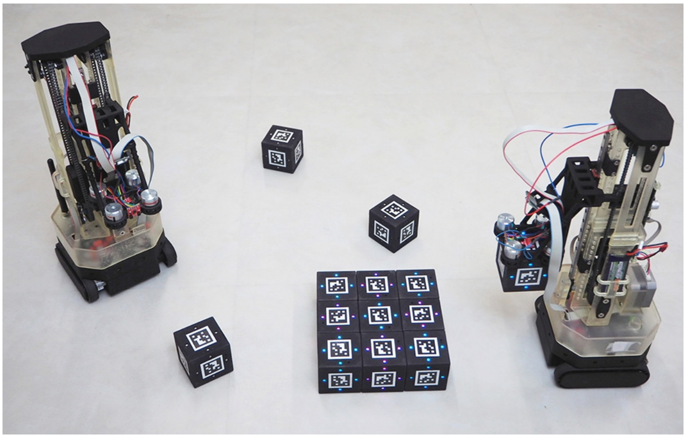


Fig. 1. The Swarm Robotics Construction System (SRoCS) consists of two components, the BuilderBot robot and the Stigmergic Block building material.

building material to coordinate its own assembly. Where possible, we compare this approach with a standard approach where the construction is coordinated exclusively by the robots. In Sect. 4, we discuss the tolerance of our system to faults and the trade-offs that are made by moving the intelligence into the blocks. We conclude the paper in Sect. 5 by suggesting several directions for future work. The results presented in this paper and the tools required to reproduce those results are open source and available as an OSF project [20].

2 Construction Algorithms

In this paper, we use two classes of algorithms for coordinating construction. The first class of algorithms, referred to as the *standard algorithms*, is a generalization of the approach used by Allwright et al. [1] and is used for comparison with the second class of algorithms. This second class of algorithms is called *block algorithms* and represents the approach where the intelligence that coordinates construction has been moved into the building material.

2.1 Standard Algorithms

In the standard algorithms, construction is coordinated exclusively through stigmergic communication. The robots perform a random walk in their environment, avoiding obstacles and searching for building material to attach to a structure. The robots perform construction actions as a response to their observations of the results of previous construction actions. In a standard algorithm, the robots are provided with a look-up table that associates intermediate construction states with construction actions. We assume here that the robots do not have access to global information and are not able to sense the complete state of larger structures. Therefore, an entry in this look-up table often does not contain the entire intermediate construction state, but rather only a partial representation of that state. This partial representation corresponds to a configuration of blocks that can be reliably detected by a robot's camera. The robots use this look-up table and their sensor readings to detect patterns of blocks in their environment and to execute the construction actions associated with them.

In our experiments with the standard algorithm, we allow robots to change the colors of the LEDs on the Stigmergic Blocks just before attaching them to a structure. Changing the LED colors on a Stigmergic Block enables a BuilderBot to detect more complex patterns of blocks with its computer vision system more reliably. After a BuilderBot has attached a Stigmergic Block to the structure, however, the block's LED colors are fixed.

2.2 Block Algorithms

In a block algorithm, the intelligence that coordinates construction is mainly in the building material. Similar to the standard algorithms, the robots perform a random walk in the environment, avoiding obstacles and searching for building material that can be added to an incomplete structure. In a block algorithm, however, the robots do not have any internal representation of the structure being built and rely on the building material for coordination.

In our system, construction starts with a single *root* Stigmergic Block in the environment. While in our experiments we assign the role of the root block statically, it would also be possible to have one or more robots assign this role to one or more blocks dynamically as a result of environmental stimuli. The root block in our current implementation of a block algorithm contains the entire target structure encoded as a rooted tree. The root block decomposes this rooted tree and sends only the required branches to its children using peer-to-peer near-field communication (NFC). This process continues until all blocks currently in the structure have received instructions from the root block. The non-root blocks in the structure continuously send data back to their parents who then forward the received data back to their parents as a single message until the root block has been reached.

Upon receiving the messages from its children, the root block can monitor construction progress, can detect incorrectly placed blocks, and can update the colors of the LEDs on the Stigmergic Blocks in the structure, triggering further construction actions by the BuilderBots. By controlling these LEDs, the root block is able to coordinate the construction of the structure by telling nearby robots where further blocks can be attached or should be removed.

Although this paper focuses primarily on results from simulation, we have successfully implemented a block algorithm using the Stigmergic Blocks, whose hardware is described in [3]. A video of this algorithm working on the hardware (with blocks being attached and detached by hand) is available online as part of the OSF project[1]. In the following section, we describe our experiments in simulation.

3 Experiments

In this section, we present three experiments that we have completed using the models of the BuilderBot and Stigmergic Block in the ARGoS simulator. We model the behavior of the Stigmergic Block firmware in ARGoS using a Lua controller that allows callbacks to be executed while messages are being exchanged. This model reflects the actual hardware with the exception that the firmware for the real block is written in C++ and is interrupt-driven, while the code used in simulation is written in Lua and uses polling to detect if a neighboring block is attempting to exchange messages. The control software for the BuilderBot robot is also written in Lua and uses a behavior tree architecture. An API for the BuilderBot has been developed, which provides a library of behavior trees for obstacle avoidance, picking up unused blocks, and attaching them to structures following rules that have been defined in terms of patterns of blocks that can be detected by the robot's computer vision system.

Our first experiment demonstrates a concept called dynamic construction paths, where the root block is able to adjust the target structure as it is being built. In the second experiment, we show how the blocks can be used to guide a robot towards a vacant construction site. Finally, the third experiment demonstrates how using a block algorithm allows for a more flexible construction process where robots can attach blocks to any vacant construction site in any order. These experiments aim to demonstrate the potential advantages of moving the intelligence that coordinates construction from the robots to the blocks.

3.1 Dynamic Construction Paths

In the standard algorithms, the Stigmergic Blocks are unable to communicate with each other, they can only have their LEDs configured by a robot to display a certain color before they are attached to a structure. The robots change the color of the blocks as part of executing a construction action. The set of rules that maps the intermediate construction states to these construction actions is prepared offline and is loaded into the memory of the robots before an experiment is started. In contrast, the block algorithms only require the root block to have the internal representation of the structure, which can also be modified during construction. This capability enables a feature called dynamic construction paths. The concept of dynamic construction paths is realized when two or more sequences of construction actions can be selected during construction according to a condition that can be detected by the root block (or one of the blocks with which it is in communication).

In this section, we set up an experiment with a structure that can be completed by following one of four different construction paths. This structure is

[1] Video: `hardware-demo.mp4` at https://osf.io/ve3za/.

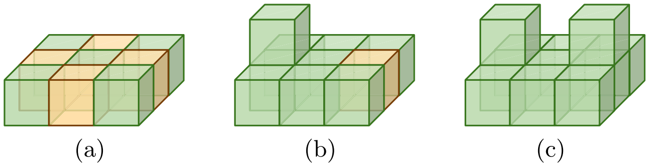


Fig. 2. Structure for demonstrating the use of a block algorithm to dynamically select construction paths. The root block in the structure indicates to nearby robots when and where a block can be attached to the structure by setting the color of a valid construction site to yellow. (Color figure online)

shown in Fig. 2. If we ignore the orientation of the structure, there are four construction paths that advance the state of the structure from what is shown in Fig. 2a to Fig. 2c. That is, we can attach blocks (i) left and then right, (ii) right and then left, (iii) front and then back, or (iv) back and then front.

In this experiment, the root block decides which path to follow by initially indicating that a block can be attached to the top face of either the left, right, front, or back block (Fig. 2a). Once a block has been attached to one of these sites (and this information has propagated back to the root block), the root block updates the illumination pattern of the structure to show nearby robots that there is one valid construction site remaining (Fig. 2b). Following the attachment of a block to this site, the root block updates the illumination pattern of the structure one last time to indicate to nearby robots that the structure is complete (Fig. 2c).

Results from Simulation. The image on the left of Fig. 3 shows a robot approaching the partially built structure. At this point, all four construction paths are possible. After the robot has placed the block on the right-hand side of the structure, the root block disables the LEDs on the right, front, and back blocks to indicate that a block can now only be added on the left. The structure is completed when the robot adds this last block to the structure, as shown on the right of Fig. 3.

To investigate the impact of using more robots, we repeated this experiment with two and four robots. Each of these configurations was repeated 25 times, with the blocks and the robots starting in random positions. Each experiment was automatically terminated when all required blocks have been deposited at the building sites. The videos and the source code for these experiments are available as part of the OSF project for this paper[2].

[2] Videos: `dcp-single-robot.mp4` and `dcp-multiple-robots.mp4` at https://osf.io/9562j/ and https://osf.io/4cpyh/.
Source code: `dcp-single-robot.zip` and `dcp-multiple-robots.zip` at https://osf.io/j2pqh/ and https://osf.io/nasf6/.

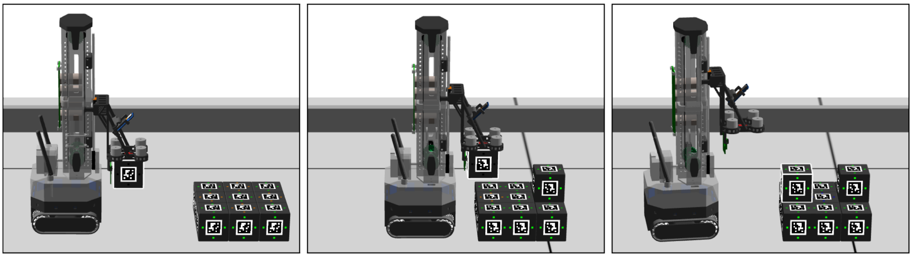


Fig. 3. Illustration of the dynamic construction paths experiment in simulation. From left to right: (i) a robot approaches a partially built structure and places a block on top of one of the orange faces, (ii) the root block responds by selecting a construction path, changing the illumination pattern, (iii) the robot places the final block in the correct location to complete the structure.

Figure 4 shows the distribution of the total experiment time with one, two, and four BuilderBots. While there is a decrease in the time taken between one and two robots, the decrease between two and four robots is not statistically significant. This diminishing return when increasing the number of robots is commonly observed in swarm robotics systems since adding more robots to a system increases the likelihood of interference between those robots. From this data, we may conclude that the use of two robots is optimal for this particular construction task in this particular environment.

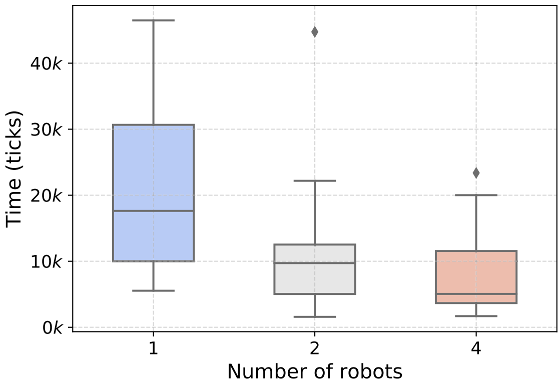


Fig. 4. Time to complete the task in simulation with different numbers of BuilderBots. Each configuration was run 25 times. Each box consists of observations ranging from the first to the third quartile. The median is indicated by a horizontal bar, dividing the box into the upper and lower parts. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outliers are shown as dots.

3.2 Guided Construction

In this experiment, we show how a block algorithm can be used to guide a robot towards a construction site. This configuration involves using the illumination pattern on the blocks to communicate the direction in which a robot should go to reach a construction site. The motivation behind implementing this mechanism is that, in the standard algorithms, the robots tend to spend a lot of time performing a random walk before locating a construction site where they can attach a block. The idea of using the building material to guide robots towards a construction site is sometimes referred to as *gradient following* and has been demonstrated before in a more abstract simulation by Werfel et al. [18].

For example, consider the partially built structure consisting of six blocks arranged in a line in Fig. 5a. To complete this structure, a robot must place one block on top of the left block and one block on top of the right block (Fig. 5b). However, since the perspective of the robot is limited, it must discover these attachment sites either through random walk or through gradient following.

In a standard algorithm, the colors of the LEDs on the blocks can not be updated once they have been attached to a structure. For this reason, the robot must rely on random walk to discover the possible attachment sites. Figure 6 shows how this construction may take place. The robots' rule set in this case is that a green block is to be attached to the top of a yellow block (unless a green block has already been attached).

The construction speed for this structure can be increased using a block algorithm that implements gradient following. In this case, the illumination pattern of the structure is under the control of the root block and can be updated in response to changes in the structure. Moreover, the robots now follow three rules: (i) when a yellow block is detected, the robot attaches a block to the top of it, (ii) if red blocks are detected, the robot biases its random walk behavior to the right, (iii) if blue blocks are detected, the robot biases its random walk behavior to the left. Figure 7 shows an example of how this construction may take place. In this example, a block is attached on top of the leftmost block, which is detected by the root block. The root block updates the illumination pattern so that a robot approaching the structure will turn to the right and find the remaining construction site.

Results from Simulation. To test our hypothesis that guided construction with a block algorithm reduces the overall construction time with respect to what is possible with a standard algorithm, we run experiments with two structures: a short line composed of six blocks and a long line composed of thirteen blocks. We run each experiment for the two structures 25 times using both the standard algorithm and the block algorithm. Figure 8 contains three screenshots of the construction of the short structure with a block algorithm in the ARGoS simulator.

The box plot in Fig. 9 shows for the shorter structure that the approach based on the block algorithm has similar performance to the approach based on the standard algorithm. However, for the longer structure, the block algorithm shows

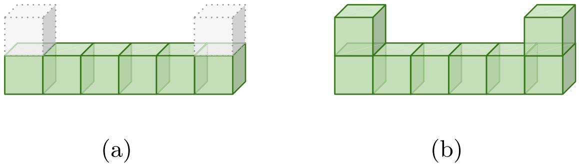


Fig. 5. Structure for demonstrating guided construction. (a) The initial state of the structure is a line consisting of six blocks. (b) The structure is completed by placing a block at each end of the structure. (Color figure online)

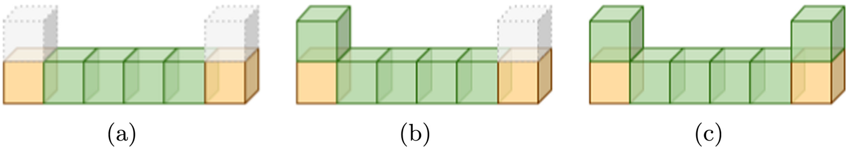


Fig. 6. Construction of the structure in Fig. 5 using a standard algorithm. (Color figure online)

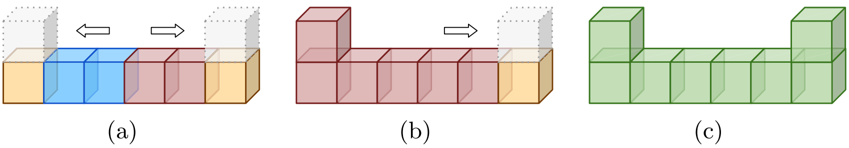


Fig. 7. Construction of the structure in Fig. 5 using a block algorithm to indicate which way a robot should turn to reach a valid construction site. (Color figure online)

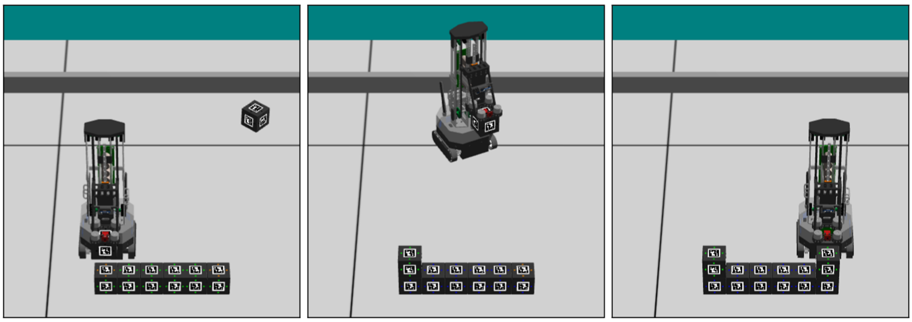


Fig. 8. Illustration of the construction of the structure in Fig. 5 with a block algorithm in the ARGoS simulator. (a) The robot attaches a block to the top of the leftmost block. (b) The illumination pattern is updated by the root block and the robot searches to the right for possible construction sites. (c) The robot attaches the last block to the top of the rightmost block.

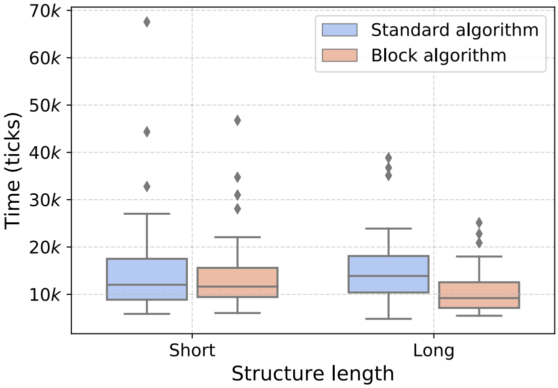


Fig. 9. Time taken to build the short and long structures in simulation using a standard algorithm and a block algorithm. Each structure was built 25 times. Each box consists of observations ranging from the first to the third quartile. The median is indicated by a horizontal bar, dividing the box into the upper and lower part. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outliers are shown as dots.

marginally better performance than the standard algorithm. From comparing the results for the two structures, it appears that the decrease in construction time is related to the size of the structure, however, further experiments with different types of structures and varying numbers of robots are needed to get a proper insight into this relationship. The videos and the source code for reproducing these experiments are available as part of the OSF project.[3]

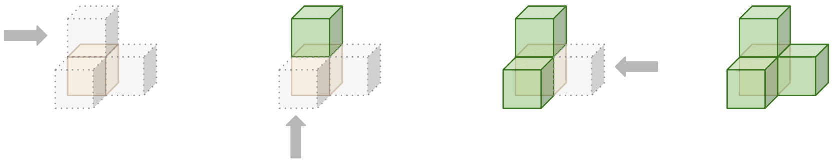
3.3 Flexible Construction

Implementing construction in a swarm robotics system using a standard algorithm puts a heavy burden on the designer to come up with a set of rules that unambiguously maps each intermediate state of a structure to a construction action. This burden is only made worse when we want to design rules that facilitate flexible construction. For example, consider the structure in Fig. 10. If we wanted to build this structure using the standard algorithm, we could constrain the building process so that there is only one construction path that can be followed, that is, there is exactly one construction action associated with each intermediate state (Fig. 11). This constrained approach, however, may be inefficient, since a robot could approach a possible construction site but be prohibited to attach a block due to the constraints of the rule set. In contrast to the constrained approach, if we allow a building process where a robot can attach

[3] Videos: `gc-standard-algorithm.mp4` and `gc-block-algorithm.mp4` at https://osf.io/5h9cs/ and https://osf.io/cdvty/.
Source code: `gc-standard-algorithm.zip` and `gc-block-algorithm.zip` at https://osf.io/we754/ and https://osf.io/3znua/.

(a) (b)

Fig. 10. Candidate structure for flexible construction. (a) initial state of the structure, (b) target state of the structure.



(a) (b) (c) (d)

Fig. 11. The structure in Fig. 10, can be built sequentially using a standard algorithm to map the possible intermediate states of the structure (configurations of blocks) to construction actions.

a block to any possible construction site at any time, the number of possible intermediate states would increase significantly. Even for the simple structure in Fig. 10, the number of intermediate states increases from three to seven. Finding the unambiguous mappings between all of these intermediate states and the possible construction actions that advance the building process while keeping the structure in a valid state is at least difficult and may in many cases be infeasible.

A block algorithm can solve this problem since the root block can detect when and where one or more blocks have been added to (or removed from) a structure and can update the illumination pattern on the blocks accordingly. Furthermore, in the case of a block being attached to an incorrect site, the root block can detect the incorrectly placed block and update the illumination pattern so that nearby robots remove it, restoring the structure to a valid intermediate state. In the final experiment for this paper, we demonstrate the construction of the structure in Fig. 10 using the ARGoS simulator.

Results from Simulation. We have implemented the construction of the structure in Fig. 10 using a standard algorithm for sequential construction with a single robot (Fig. 12) and with a block algorithm for construction of the same structure with three robots in parallel. Videos of these experiments and the related source code are available online as part of our OSF project for this research[4].

[4] Videos: `fc-standard-algorithm.mp4` and `fc-block-algorithm.mp4` at https://osf.io/ycxes/ and https://osf.io/tvhs2/.
Source code: `fc-standard-algorithm.zip` and `fc-block-algorithm.zip` at https://osf.io/gf94r/ and https://osf.io/kjhu7/.

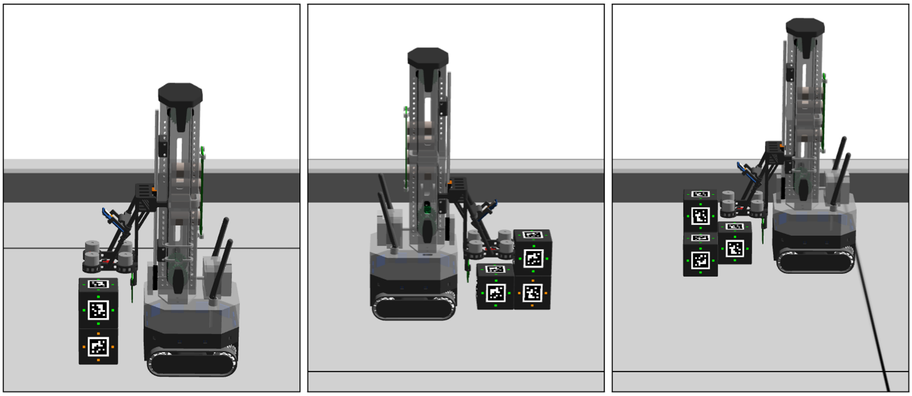


Fig. 12. Construction of the structure in Fig. 10 using a standard algorithm with a single robot in simulation.

4 Discussion

In this section, we compare the ability of the standard algorithms and the block algorithms to recover from faults during construction. Furthermore, we consider the trade-offs that we have made as a result of introducing a degree of centralization into our system and as a result of requiring more capable building materials.

4.1 Fault Tolerance

For this part of our discussion, we consider two types of faults and how the standard and block algorithms can recover from them. The first type of fault is when a robot attaches a block to an incorrect site. This fault can be caused by a sensor error on the behalf of the robot or can be due to unfortunate timing. For example, when two or more robots attach blocks to valid attachment sites but where the combination of those attachments puts the structure into an incorrect state. The second type of fault is when a block stops working correctly. This fault may be the result of a bad power source, corrupted firmware, or damaged hardware.

The standard algorithms can handle the first type of fault, where a block has been incorrectly attached to a structure, at the cost of increasing the complexity of the rule set. That is, in addition to the rules necessary to advance the construction, it would be possible to add rules that match the structure when it is in an incorrect state and that trigger the removal of one or more blocks until the structure is back in a state from which the construction can continue. The second type of fault is difficult to solve with the standard algorithm and relies on the robots being able to infer that a block is faulty, e.g., the LEDs are displaying the wrong color. If the robots detect a faulty block in the structure, it can be ignored or removed if it is disruptive to the building process.

For the block algorithms, the first type of fault, where a block has been attached to an incorrect site, can be resolved since the root block can detect the presence of this block by exchanging messages with other blocks in the structure and can update the illumination pattern of the structure so that the robots remove it. A demonstration of a block algorithm recovering from this fault has been implemented for the dynamic construction paths discussed in Sect. 3.1. A video of the recovery from this fault is available as part of the OSF project along with the source code to reproduce the experiment[5].

The second type of failure, that is, if the block has (i) a bad power source, (ii) corrupted firmware, or (iii) damaged hardware, is more problematic for block algorithms than standard algorithms since the block algorithms currently rely on the accurate propagation of information through the structure. In some cases, it may be possible to work around these malfunctioning blocks by communicating through other blocks; however, thin sections of the structure where there is only a single path through which information can flow remain problematic and will require further research.

4.2 Trade-Offs

Although the experiments in this paper show that the block algorithms can make construction more flexible and efficient and can put less of a burden on the system designer, there are some important trade-offs that must be addressed. The first trade-off is the increase in complexity of the building materials, which can no longer be passive but now have to be capable of computation and local communication, which increases the cost and necessitates a source of power. This trade-off, however, is not so unreasonable considering recent developments in smart label technology where NFC communication, small micro-controllers, and lithium batteries can be combined into cheap flexible tags that could be attached to building materials in an automated construction system.

The second trade-off that must be considered is that a block algorithm uses a root block in the structure to coordinate its construction, introducing a form of centralized control which may be undesirable since (i) it is a potential bottleneck in terms of computational and communication throughput and (ii) it creates a single point of failure in the system. We believe, however, that it is feasible to use centralized control in a swarm robotics construction system without negating the benefits of decentralized control as long as the following conditions can be met: (i) the role of the centralized controller can be transferred to another unit in the case of hardware failure, and (ii) the centralized controller can partially delegate its authority to other units so that it is not a computational/communication bottleneck in the system (see [10] for recent research in these directions).

[5] Video: `dcp-fault-tolerance.mp4` at https://osf.io/mvhk6/.
Source code: `dcp-fault-tolerance.zip` at https://osf.io/scm7q/.

5 Conclusion

In this paper, we demonstrated the advantages of moving the intelligence that coordinates a building process in a swarm robotics construction system from the robots and into the building material. We referred to these algorithms as block algorithms and compared them against solutions where the intelligence that coordinates construction was in the robots, namely the standard algorithms.

In future work, we intend to investigate the scalability and fault tolerance of the block algorithms and to validate the experiments presented in this paper using real robots.

Acknowledgments. This work is partially supported by the Program of Concerted Research Actions (ARC) of the Université libre de Bruxelles, by a Research Credit (CDR – Crédit de Recherche) grant from the Belgian F.R.S.-FNRS, and by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 846009. Yating Zheng and Weixu Zhu would like to acknowledge their support from the China Scholarship Council (grant numbers 201806040106 and 201706270186). The research in this paper was partly undertaken at the UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Ministerio de Economía y Competitividad (DPI2015-69041-R) and by Universitat Jaume I (UJI-B2018-74). Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Director.

References

1. Allwright, M., Bhalla, N., Dorigo, M.: Structure and markings as stimuli for autonomous construction. In: Proceedings of the Eighteenth International Conference on Advanced Robotics, pp. 296–302. IEEE (2017). https://doi.org/10.1109/icar.2017.8023623
2. Allwright, M., Bhalla, N., Pinciroli, C., Dorigo, M.: Simulating multi-robot construction in ARGoS. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Reina, A., Trianni, V. (eds.) ANTS 2018. LNCS, vol. 11172, pp. 188–200. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00533-7_15
3. Allwright, M., Zhu, W., Dorigo, M.: An open-source multi-robot construction system. HardwareX **5**, e00050 (2019). https://doi.org/10.1016/j.ohx.2018.e00050
4. Bonabeau, E., Guérin, S., Snyers, D., Kuntz, P., Theraulaz, G.: Three-dimensional architectures grown by simple 'stigmergic' agents. BioSystems **56**(1), 13–32 (2000). https://doi.org/10.1016/s0303-2647(00)00067-8
5. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: a review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013). https://doi.org/10.1007/s11721-012-0075-2
6. Grassé, P.P.: Reconstruction of the nest and coordination between individuals in terms. Bellicositermes Natalensis and Cubitermes sp. the theory of stigmergy: Test interpretation of termite constructions. Insectes Sociaux 6(1), 41–80 (1959). https://doi.org/10.1007/bf02223791
7. Hamann, H.: Swarm Robotics: A Formal Approach. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74528-2

8. Jones, C., Matarić, M.J.: From local to global behavior in intelligent self-assembly. In: 2003 IEEE International Conference on Robotics and Automation, pp. 721–726. IEEE (2002). https://doi.org/10.1109/robot.2003.1241679
9. Jones, C., Matarić, M.J.: Automatic synthesis of communication-based coordinated multi-robot systems. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 381–387. IEEE (2004). https://doi.org/10.1109/iros.2004.1389382
10. Mathews, N., Christensen, A.L., O'Grady, R., Mondada, F., Dorigo, M.: Mergeable nervous systems for robots. Nat. Commun. **8**(439), 1–7 (2017). https://doi.org/10.1038/s41467-017-00109-2
11. Petersen, K., Nagpal, R., Werfel, J.: TERMES: An autonomous robotic system for three-dimensional collective construction. In: Proceedings of Robotics: Science and Systems, pp. 257–264. RSS Foundation (2011). https://doi.org/10.15607/rss.2011.vii.035
12. Pinciroli, C., et al.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. Swarm Intell. **6**(4), 271–295 (2012). https://doi.org/10.1007/s11721-012-0072-5
13. Sugawara, K., Doi, Y.: Collective construction by cooperation of simple robots and intelligent blocks. In: Kubota, N., Kiguchi, K., Liu, H., Obo, T. (eds.) ICIRA 2016. LNCS (LNAI), vol. 9834, pp. 452–461. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43506-0_40
14. Sugawara, K., Doi, Y.: Collective construction of dynamic equilibrium structure through interaction of simple robots with semi-active blocks. In: Chong, N.-Y., Cho, Y.-J. (eds.) Distributed Autonomous Robotic Systems. STAR, vol. 112, pp. 165–176. Springer, Tokyo (2016). https://doi.org/10.1007/978-4-431-55879-8_12
15. Theraulaz, G., Bonabeau, E.: Coordination in distributed building. Science **269**(5224), 686–688 (1995). https://doi.org/10.1126/science.269.5224.686
16. Theraulaz, G., Bonabeau, E.: A brief history of stigmergy. Artif. Life **5**(2), 97–116 (1999). https://doi.org/10.1162/106454699568700
17. Werfel, J., Nagpal, R.: Extended stigmergy in collective construction. IEEE Intell. Syst. **21**(2), 20–28 (2006). https://doi.org/10.1109/mis.2006.25
18. Werfel, J., Nagpal, R.: Three-dimensional construction with mobile robots and modular blocks. Int. J. Robot. Res. **27**(3–4), 463–479 (2008). https://doi.org/10.1177/0278364907084984
19. Werfel, J., Petersen, K., Nagpal, R.: Designing collective behavior in a termite-inspired robot construction team. Science **343**(6172), 754–758 (2014). https://doi.org/10.1126/science.1245842
20. Zheng, Y., Allwright, M., Zhu, W., Kassawat, M., Han, Z., Dorigo, M.: Hybrid coordination for swarm construction (2020). https://doi.org/10.17605/osf.io/vh2k6