Provably Manipulable 3D Structures using Graph Theory

Extended Abstract

John Harwell
University of Minnesota
Minneapolis, United States
harwe006@umn.edu

London Lowmanstone
University of Minnesota
Minneapolis, United States
lowma016@umn.edu

Maria Gini
University of Minnesota
Minneapolis, United States
gini@umn.edu

ABSTRACT

We identify barriers to a broader application of multi-robot systems to construction and deconstruction tasks, which represent important real-world problems, such as repairing critical infrastructure of roads and levies after a disaster. We frame these tasks as instances of the parallel bricklayer problem, where independent agents must coordinate to concurrently manipulate aspects of a 3D environment without deadlocks. We extract desirable properties of graphs representing natural 3D structures and sketch a graphical representation to model and reason about structures composed of discrete cuboid blocks. We present a sample algorithm sketch for a non-trivial structure utilizing our model.

KEYWORDS

Multi-robot systems; Swarm Robotics; Construction; Graph Theory

ACM Reference Format:
John Harwell, London Lowmanstone, and Maria Gini. 2023. Provably Manipulable 3D Structures using Graph Theory: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023,* IFAAMAS, 3 pages.

1 INTRODUCTION

We define the *Parallel Bricklayer Problem (PBP)* as a problem in which robots interact with each other to concurrently manipulate objects in their environment, much like human bricklayers work together to build something out of bricks or cinder blocks. In addition to strictly additive changes, they can create temporary scaffolds to be able to access or build certain parts of a structure by adding/removing things. All activities must be coordinated among the bricklayers to ensure overall correctness, which is non-trivial.

Two important real-world problems can be modeled as variants of the PBP: *construction* and its counterpart, *deconstruction* [8]. This dichotomy also includes their associated variants: cinderblock laying/removal, box stacking to minimize volume used, etc. The use of *individual* robots for automation in simple variants of these tasks is widespread, and robots can operate with minimal human supervision. However, for larger and more complex variants of the PBP, autonomous *Multi Robot Systems (MRSs)* are needed for effective operation without extensive human supervision.

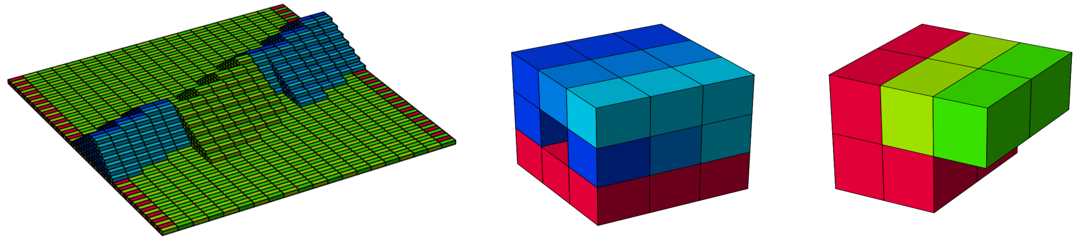
Single-robot automation in these tasks is not trivial; multi-robot automation exponentially more so. Successful robots must have the ability to: (a) sense and manipulate assembly components; (b) interact with the desired structure at all stages of the assembly/disassembly process; (c) satisfy a variety of precedence constraints to ensure assembly/disassembly correctness; and (d) ensure static stability and structural integrity throughout the assembly/disassembly process. Modern engineering can develop individual robots with these abilities; however, robot control software must also guarantee correct, deadlock-free operation with multiple robots, which is exponentially more difficult—even in idealized indoor environments. This lack of linear scaling in engineering complexity because of coordination difficulties makes MRS solutions much more difficult to develop. Because the environment of the PBP is *dynamic*, and changes as time passes, to be successful robots must model and adapt to these changes. However, there is no unified modeling framework that provides the necessary guarantees so that theoretical results developed for simpler single-robot approaches can be leveraged to quickly develop algorithms for MRS approaches. With these factors, it is not surprising that multi-robot automation in construction and deconstruction tasks is largely absent.

2 MODEL SUMMARY

We represent cuboid objects to be manipulated uniformly as nodes within a graph, and obtain properties of objects—such as size—from the graph structure. See Fig. 1 for some examples of structures and their graph representations. Our model enables fast development of simple robot control algorithms while also allowing unparalleled expressiveness for formally representing 3D structures. By leveraging the abstract power of graph theory, our model encompasses construction, deconstruction, and multi-phase operations, allowing it to be applied across problem domains. Within our model, we take inspiration from the modular decomposability of structures built by social insects to provide a mechanism for provably correct, *parallel* manipulation of subgraphs from one state to another. We achieve $\Omega(N)$ parallelism by only requiring manipulations within each subgraph to be serial; previous works enabled only $\Omega(1)$ parallelism. That is, regardless of the number of agents working on a structure, few, if any, simultaneous attachments or placements are possible [6, 9, 10].

The provable guarantees of our model do not require inter-agent communication (except for stigmergic communication) in order to derive agent controllers. Minimal local agent knowledge is required for provable operation with the cuboid materials that are frequently used in human construction projects—agents can even be purely reactive and memory-less. Thus, by handling the complexity of coordination in the model, rather that the agents, we enable agents to be simple and reactive, and solve an important preliminary step in realizing large-scale autonomous construction systems of cost-effective robots.

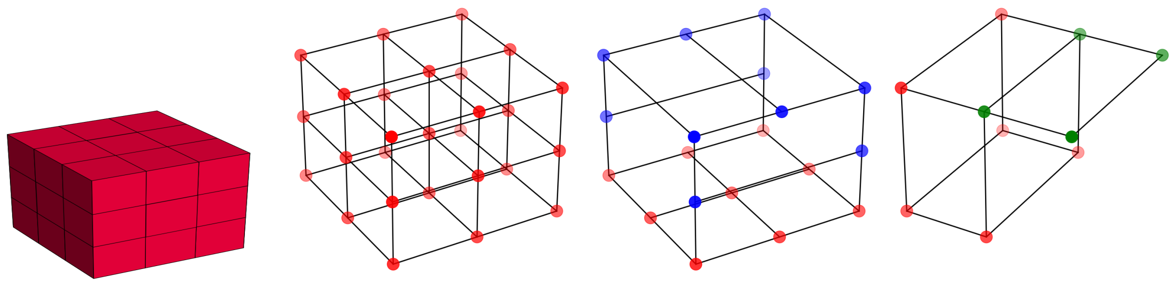
Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.



(a) Complex levee structure.

(b) A cube with a hole through it (valid structure).

(c) A cube with overhangs (invalid structure).



(d) Cube.

(e)

(f)

(g)

Figure 1: Examples of cuboid structures and graph representations. Fig. 1a sits on a bed of interlocking bricks and is heterogeneous; the middle section could be a patch for a breached wall. Validity of Fig. 1b/Fig. 1f obtained through examining planarity of graph "slices" along X or Y axis; vice versa for Fig. 1c/ Fig. 1g.

3 SAMPLE APPLICATION SKETCH

We consider the structure shown in Fig. 1a. Let $\mathcal{G}$ be the graph representing the structure. We slice $\mathcal{G}$ along Y or X every 2 units to obtain a set of construction lanes ($g \in \mathcal{G}$), each with the same properties as $\mathcal{G}$ which can be constructed independently in parallel. Each lane has *entry/exit* traffic flows within it on the left- and right-hand sides of the lane. To construct our algorithm, we consider works studying the construction process of structures found in nature [1, 2, 4, 5, 7]. Within these works, agents manipulate structures according to local *stigmergic configurations* of blocks that trigger different behaviors [3]. Stigmergic configurations provide inter-agent coordination, and are generally small and localized; large configurations may rarely be encountered during construction [1].

Using this inspiration, it is easy to imagine a simple algorithm suitable for parallel manipulation of structures which *exclusively* uses stigmergic rules—no memory or communication needed. We use $\Omega = \{\cup \omega_i(C)\}$ as the set of stigmergic rules which will trigger a robot action; i.e., $\omega_i(C) : C \rightarrow m(r_i)$ for some stigmergic configuration C. $\|\Omega\| = 3$: encountering a "filled" or empty lane will trigger block placement at the back of the entry flow, and encountering a gap in the exit flow will trigger a block placement in the gap, which will in turn lead to the "filled" configuration for the next robot. We assume robots (a) know the structure to build, (b) can localize accurately within the structure, (c) can perceive everything within $D = 3$ cells around them, (d) can travel over "staircases" by some mechanism. Before executing the algorithm, each robot allocates one of the construction lanes $l = (\overrightarrow{e_s}, \overrightarrow{e_e}, \overrightarrow{x_s}, \overrightarrow{x_e})$, defined by the starting and ending locations for each traffic flow ($\overrightarrow{e_s}, \overrightarrow{e_e}$ and $\overrightarrow{x_s}, \overrightarrow{x_e}$, respectively), and travels to $\overrightarrow{e_s}$.

Algorithm 1 Block placement algorithm sketch for r_i

if $detected(r_j)$ and $ahead(r_j, r_i)$ **then**
  return WAIT *// r_j is further ahead in flow*
end if
if $\omega_i(C) \in L(\overrightarrow{r_i})$ **then**
  return $m(r_i)$ *// place carried block*
end if
if $finished(m(r_i))$ **then**
  return Travel to $\overrightarrow{x_s}$ *// block placed–time to go*
end if
if $aligned(\overrightarrow{r_i}, \overrightarrow{x_s})$ **then**
  return Travel to $\overrightarrow{x_e}$ *// in exit flow*
end if
return Travel forward in entry flow *// towards $\overrightarrow{e_e}$*

We make the following observations. First, this controller is extremely simple, and can be used effectively with structures and MRSs of arbitrary size. Second, leveraging the power of graph theory, the above algorithm is *not* tied to ground or flying robots, and as long as a suitable decomposition exists for $\mathcal{G}$, correct concurrent manipulation is guaranteed, and the algorithm is reusable. Furthermore, the mechanism for guaranteeing correct concurrent operation is simple: if r_i gets too close to r_j in front, it will wait for r_j wait for it to move outside $L(\overrightarrow{r_i})$.

REFERENCES

[1] Eric Bonabeau, Sylvain Guérin, Dominique Snyers, Pascale Kuntz, and Guy Theraulaz. 2000. Three-dimensional architectures grown by simple 'stigmergic' agents. *BioSystems* 56, 1 (2000), 13–32.

[2] Seth Bullock, Dan Ladley, and Michael Kerby. 2012. Wasps, Termites, and Waspmites: Distinguishing Competence from Performance in Collective Construction. *Artificial Life* 18, 3 (2012), 267–290.

[3] Plerre-P. Grassé. 1959. La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux* 6, 1 (01 Mar 1959), 41–80. https://doi.org/10.1007/BF02223791

[4] Alexander Grushin and James A. Reggia. 2006. Stigmergic self-assembly of prespecified artificial structures in a constrained and continuous environment. *Integrated Computer-Aided Engineering* 13, 4 (2006), 289–312.

[5] Dan Ladley and Seth Bullock. 2004. Logistic Constraints on 3D Termite Construction. In *Ant Colony Optimization and Swarm Intelligence*, Marco Dorigo, Mauro Birattari, Christian Blum, Luca Maria Gambardella, Francesco Mondada, and Thomas Stützle (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 178–189.

[6] Kirstin Petersen, Radhika Nagpal, and Justin Werfel. 2011. TERMES: An Autonomous Robotic System for Three-Dimensional Collective Construction. In *Robotics: Science and Systems (RSS)*. roboticsproceedings.org, 35–42. http://www.roboticsproceedings.org/rss07/p35.pdf

[7] G. Theraulaz and E. Bonabeau. 1995. Coordination in distributed building. *Science* 269, 5224 (Aug 1995), 686–688.

[8] J. Werfel and R. Nagpal. 2006. Extended stigmergy in collective construction. *IEEE Intelligent Systems* 21, 2 (2006), 20–28.

[9] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. 2011. Distributed multi-robot algorithms for the TERMES 3D collective construction system. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 1–6.

[10] James Worcester, Joshua Rogoff, and M. Ani Hsieh. 2011. Constrained task partitioning for distributed assembly. In *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 4790–4796.