Block by Block
Collaborative Strategies for Multi-Agent Robotic Construction
Workshop at ICRA 2025 - May 23, 2025

A. Degani[1,*], J. Werfel[2], H. Choset[3], B. Vundurthy[4], G. Gutow[5], F. Oliva[6,*]

Abstract

The Block by Block workshop brought together researchers working at the intersection of robotics, construction, and multi-agent systems. As infrastructure development increasingly turns to robotic solutions for efficiency and resilience, understanding how agents can act collaboratively in unstructured and dynamic environments becomes essential.

This workshop focused on strategies for multi-agent collaboration in construction tasks, including site preparation, material manipulation, and task coordination under real-world constraints such as limited communication, localization, and actuation capabilities. A particular emphasis was placed on push manipulation, modular construction, and distributed decision-making in GPS-denied or challenging environments.

Overview

The abstracts included in this collection were peer reviewed by the workshop's organizing and program committee to ensure relevance and quality. The accompanying short papers were not peer reviewed, but were proofread for clarity and formatting. All contributions were presented at the Block by Block: Collaborative Strategies for Multi-Agent Robotic Construction workshop at ICRA 2025, through a lightning talk and a poster session.

This collection also features six invited speakers, whose bios and talk abstracts are included at the beginning of the document. Their contributions framed the day's discussion and offered broad insights into the future challenges of the field.

The papers gathered here represent a wide range of approaches and perspectives, from algorithmic planning and coordination frameworks to hardware design and field deployment. These contributions reflect ongoing research and ideas shared within the community, and are published to encourage continued discussion and collaboration.

Proceedings Contents

[1]Civil and Environmental Engineering - Technion Israel Institute of Technology, adegani@technion.ac.il
[2]Harvard's School of Engineering and Applied Science, jkwerfel@seas.harvard.edu
[3]Robotics Institute - Carnegie Mellon University, choset@andrew.cmu.edu
[4]Robotics Institute - Carnegie Mellon University, pvundurt@andrew.cmu.edu
[5]Robotics Institute - Carnegie Mellon University, ggutow@andrew.cmu.edu
[6]Civil and Environmental Engineering - Technion Israel Institute of Technology, federicoo@campus.technion.ac.il
[*]Technion Autonomous Systems Program, Technion - Israel Institute of Technology
We gratefully acknowledge the support of the IEEE ICRA 2025 organizing committee for providing the platform and infrastructure that made this workshop possible.

Abstracts of the Invited Speakers

Collective Construction in Landscape Architecture

Karen-Lee Bar Sinai

Abstract

This presentation explores the emergence of collective robotic tools in landscape architecture—such as autonomous earthwork systems, extraterrestrial construction tools for in-situ resource utilization (ISRU), and swarm robotics for terrain and erosion management. We argue that these technologies represent a shift from top-down design toward situated, adaptive construction that dynamically responds to environmental conditions. Drawing inspiration from natural systems—particularly beavers as ecosystem engineers—we reframe autonomy as a distributed, ecological process. Beavers collaboratively reshape landscapes through simple, decentralized actions, resulting in complex constructs, reconfigured hydrological systems, and enriched ecologies. We will present early research into beaver-inspired tools and envision a future autonomous system composed of responsive agents working with, rather than against, ecological systems and environmental flows. By learning from natural collectives, we propose a reframing of environmental robotics as a co-evolutionary practice—integrating robotics, ecology, and design. This nascent field uses robotic tools to facilitate collaborative environmental processes and cultivate new synergies with the environment.

Biography

Dr. Karen Lee Bar-Sinai is an Assistant Professor of Materials and Design in Landscape Architecture and the Harvard University Graduate School of Design. She is a licensed architect, an urbanist, and holds a Ph.D. and postdoctoral training in robotic construction with found matter. Her GSD research group investigates the interaction between tools, materials, and the environment aiming to shift how and with what we build in the face of imminent material scarcity, environmental challenges, and climate change. Her research spans from small through territorial to planetary scales, all involving the modulation of matter in architectural or landscape construction. Current projects include Environmental Robotics – starting with 'beaver-bots' – simulating and developing beaver-inspired tools for restoring wetlands, with applications also in beaver-less sites like erosion-prone arid regions. In addition, she explores ways to deploy living materials – from fungi to root systems – as instruments for construction. She is also advancing Planetary Design Computation, testing the potential of targeted local landscape design to influence global climate-system dynamics. Karen Lee has lectured broadly on architecture, landscape architecture, and technology. She currently teaches core studios in landscape architecture and 'Eco-Machina' – a seminar on the emerging relationships between machines and landscapes.

Multi Agent Matter

Karola Dierichs

Abstract

Granular materials such as sand are modular systems: a large number of units—for example sand grains—are in loose contact with each other. This loose interaction allows for the formation of solid, liquid and gaseous states in granular materials. The characteristics of a granular material are defined by the materiality and geometry of its component units—the particles. If these particles are designed, entirely new characteristics can be programmed into a granular material. Departing from this notion of a designed granular material the talk will show how granular materials can become multi-agent matter. Moving from self-interlocking particles for architecture-scale construction to autonomously entangling ones—we will pose the question how matter itself can become a robotic system.

Biography

Karola Dierichs holds the Material and Code professorship in the Cluster of Excellence Matters of Activity. In addition, she is a researcher at the Max Planck Institute of Colloids and Interfaces. Her expertise lies in the fields of materials design and minimal machines for architectural construction. Here, the main goal is to establish architecture as sourced from and embedded in a given environment. For this, methods of science and art are integrated to establish a novel paradigm of fundamental research. Previous affiliations include the Institute for Computational Design and Construction at the University of Stuttgart, where she has conducted research in the field of designed granular materials for architecture.

Docking Mechanisms for Lunar Surface Technology

Mark Yim

Abstract

This talk will deal with the progress in the NASA TRUSSES project for docking mechanisms for robots to attach to each other in the context of Lunar regolith interaction. This project explores methods for teams of robots to jointly overcome environmental hazards on the Moon by attaching to each other to form larger and more stable, maneuverable structures. In the process, we can explore ground interactions required for building regolith structures and site preparation on the moon.

Biography

Mark Yim's research interests began with modular robots that are made up of identical active components that can be arranged to form many different configurations, ranging from a snake robot to a humanoid to a 17 legged centipede. These systems can also self-reconfigure, changing the robot's shape to suit the task. In addition to self-reconfiguring and self-assembling robots, Mark has also started work on flying robots, and task specification, working to figure out how to specify a task so that a robot configuration can optimally satisfy that task.

Planning with the TERMES Robots

Sven Koenig

Abstract

Cooperative multi-robot planning is an understudied but important area of AI planning due to the increasing importance of multi-robot systems. The Harvard TERMES robots can build 3D structures by picking up individual blocks, carrying them around, putting them down, and climbing them. Planning for single robots is already difficult due to the large number of blocks and long plans. Planning for multiple robots is even more difficult since it needs to reason about how to achieve a high degree of parallelism without agents obstructing each other, even though many robots operate together in tight spaces. In this talk, I describe previous research on a first (domain-dependent, centralized, and non-optimal) multi-agent planning method for this domain that compared favorably to off-the-shelf planning technologies.

Biography

Sven Koenig is Chancellor's Professor and Bren Chair at the University of California, Irvine and a Fellow of AAAI, AAAS, ACM, and IEEE. Additional information about him can be found on his webpage.

Decentralized Strategies for Multi-Robot Object Transport and Collision-Free navigation

Spring Berman

Abstract

Robot collectives that perform construction tasks must be able to safely navigate around changing arrangements of materials and cooperatively transport objects that are too large or heavy to be moved by a single robot. To achieve collision-free robot navigation in environments with unknown obstacles, we have designed nonlinear model predictive control methods, integrating barrier functions learned from the robot's sensor measurements, as well as virtual potential-based controllers. We have also developed decentralized robot controllers for cooperative transport that, unlike prior approaches, do not rely on inter-robot communication or prior information about the object, environment, or robot transport team; each robot only requires the target object location or velocity and measurements of its own state. Some of these controllers were designed to reproduce observed features of group food retrieval in desert ants, with the aim of producing similarly robust transport behaviors. These navigation and control strategies are demonstrated in numerical and physics-based simulations and in experiments with small mobile ground robots.

Biography

Spring Berman is an Associate Professor of Mechanical and Aerospace Engineering and Graduate Faculty in Computer Science, Electrical Engineering, and Exploration Systems Design at Arizona State University (ASU). She directs the Autonomous Collective Systems Laboratory and is an Associate Director of the Center for Human, Artificial Intelligence, and Robot Teaming (CHART) within the ASU Global Security Initiative. Prior to joining ASU in 2012, she was a postdoctoral researcher in Computer Science at Harvard University. She received the Ph.D. and M.S.E. degrees in Mechanical Engineering and Applied Mechanics from the University of Pennsylvania and the B.S.E. degree in Mechanical and Aerospace Engineering from Princeton University. She was a recipient of the ONR Young Investigator Award (2016) and the DARPA Young Faculty Award (2014). Her research focuses on the synthesis of scalable control strategies, including bio-inspired controllers, for robotic swarms and other types of distributed systems.

Towards Learned Cooperation at Scale in Robotic Multi-Agent Systems

Guillaume Sartoretti

Abstract

With the recent advances in sensing, actuation, computation, and communication, the deployment of large numbers of robots is becoming a promising avenue to enable or speed up complex tasks in areas such as manufacturing, last-mile delivery, search-and-rescue, or autonomous inspection. My group strives to push the boundaries of multi-agent scalability by understanding and eliciting emergent coordination/cooperation in multi-robot systems as well as in articulated robots (where agents are individual joints). Our work mainly relies on distributed (multi-agent) reinforcement learning, where we focus on endowing agents with novel information and mechanisms that can help them align their decentralized policies towards team-level cooperation. In this talk, I will first summarize my early work in independent learning, before briefly discussing my group's recent advances in convention, communication, and context-based learning. Throughout this journey, I will highlight the key challenges surrounding learning representations, policy space exploration, and scalability of the learned policies, and outline some of the open avenues for research in this exciting area of robotics.

Biography

Guillaume Sartoretti joined the Mechanical Engineering Department at the National University of Singapore (NUS) as an Assistant Professor in 2019, where he founded the Multi-Agent Robotic Motion (MARMot) lab. Before that, he was a Postdoctoral Fellow in the Robotics Institute at Carnegie Mellon University (USA), where he worked with Prof. Howie Choset. He received his Ph.D. in robotics from EPFL (Switzerland) in 2016 for his dissertation on "Control of Agent Swarms in Random Environments," under the supervision of Prof. Max-Olivier Hongler. His passion and research lie in understanding and eliciting emergent coordination/cooperation in large multi-agent systems, by identifying what information and mechanisms can help agents reason about their individual role/contribution to each other and to the team. Guillaume was a Manufacturing Futures Initiative (MFI) postdoctoral fellow at CMU in 2018-2019, was awarded an Amazon Research Awards in 2022, as well as an Outstanding Early Career Award from NUS' College of Design and Engineering in 2023.

DUA: a containerized architecture for coordinating heterogeneous multi-agent robotic teams

Roberto Masocco[1], Alessandro Tenaglia[1], Federico Oliva[2], Simone Mattogno[1], Lorenzo Bianchi[1], Alexandru Cretu[1], Giorgio Manca[1], Daniele Carnevale[1]

***Abstract*— The Distributed Unified Architecture (DUA) framework addresses a fundamental challenge in multi-agent robotic systems: enabling seamless coordination across heterogeneous platforms. As robotic construction tasks increasingly demand collaboration between diverse agents with varying capabilities, the underlying software infrastructure must support efficient interoperability without compromising performance. DUA provides a containerization-based solution that creates consistent operating environments across different hardware platforms while preserving platform-specific features and optimizations, with the ultimate goal of streamlining integration, research, and prototyping activities.**

I. Introduction

Multi-agent robotic systems are increasingly deployed in construction applications where collaborative tasks demand coordination between diverse autonomous agents with varying computational capabilities, sensor configurations, and hardware architectures [1], [2]. The complexity of modern construction scenarios requires swarms that can seamlessly integrate heterogeneous platforms, from resource-constrained single-board computers managing simple manipulation tasks to high-performance systems equipped with specialized acceleration hardware for real-time perception and planning algorithms [3].

The fundamental challenge in deploying such heterogeneous teams lies in the software infrastructure required to enable effective coordination. Each robotic platform typically requires platform-specific device drivers, customized middleware configurations, and tailored deployment procedures that create significant barriers to rapid prototyping and system integration. Traditional approaches to multi-agent coordination often assume homogeneous hardware configurations or require extensive manual configuration processes that scale poorly across diverse agent populations [4].

Existing frameworks for distributed robotic systems, while addressing specific aspects of multi-agent coordination, fail to provide comprehensive solutions for hardware abstraction across heterogeneous platforms. Solutions based on, *e.g.*, Robot Operating System 2 [5], [6] excel in providing communication primitives but require substantial manual effort to configure consistent environments across different hardware targets. Similarly, existing containerization approaches in robotics focus primarily on application isolation rather than creating unified development and deployment workflows that span diverse hardware architectures [7], [8].

Novel contribution. This paper presents the Distributed Unified Architecture (DUA) [9], [10], [11], a containerization-based framework designed to address the fundamental software infrastructure challenges in heterogeneous multi-agent robotic systems. DUA provides a systematic approach to abstracting hardware differences while preserving platform-specific optimizations, enabling developers to create modular software components that deploy consistently across diverse robotic platforms with minimal reconfiguration overhead.

The primary contributions of this work include the design and implementation of a scalable *base unit* system that creates consistent operating environments across heterogeneous hardware platforms, a template-based modular integration approach that facilitates rapid composition of complex multi-agent systems, and demonstrated applicability across real-world autonomous systems spanning multiple hardware architectures and computational capabilities.

II. System architecture

The Distributed Unified Architecture addresses the heterogeneous development and deployment challenge through a containerization-based approach that creates consistent operating environments while preserving platform-specific capabilities. The framework's design philosophy centers on providing *similar*, rather than identical, environments across diverse hardware platforms, enabling developers to leverage unique platform features while maintaining consistent development and deployment workflows.

A. Core design principles

DUA is built upon three fundamental principles that directly address the challenges of heterogeneous multi-agent development and coordination. First, the framework provides hardware abstraction without sacrificing platform-specific optimizations, ensuring that specialized capabilities such as GPU acceleration or real-time processing features remain accessible to applications. Second, the system maintains modularity at multiple levels, enabling individual software components to be developed independently and integrated seamlessly into larger project, up to multi-agent architectures. Third, the framework minimizes deployment overhead by eliminating the need for manual environment configuration across different target platforms.

The containerization approach leverages Docker Engine capabilities to create self-sufficient yet hardware-aware execu-

Corresponding author: roberto.masocco@uniroma2.it.
[1] Department of Civil Engineering and Computer Science Engineering, Tor Vergata University of Rome, Via del Politecnico 1, 00133 Rome, Italy.
[2] Department of Civil and Environmental Engineering, Technion Israel Institute of Technology, Technion City, Haifa, 3200003, Israel.

tion environments. Unlike traditional virtualization solutions that introduce significant computational overhead, DUA containers are configured to provide direct access to host hardware resources, including network stacks, communication interfaces, and specialized accelerators. This configuration enables applications to achieve near-native performance while benefiting from the consistency and portability advantages of containerized deployment.

B. Framework architecture

The DUA framework consists of two primary architectural components that work together to enable seamless deployment across heterogeneous platforms. The foundation layer comprises platform-specific *base units* that provide consistent system configurations tailored to different hardware architectures and computational capabilities. Above this foundation, a standardized project template enables rapid development and integration of modular software components that can be deployed across any supported platform configuration.

The base units serve as the core abstraction mechanism, with each unit representing a Docker image optimized for a specific combination of hardware architecture and intended use case. These images begin with platform-appropriate Ubuntu Linux distributions and systematically add common dependencies including compilation toolchains, middleware packages for distributed software communications (ROS 2, Data Distribution Service (DDS) implementations [12], [13], [14], [15], [16], Zenoh [17]), and platform-specific optimizations. The systematic construction process ensures that applications developed on one platform can be deployed on others with minimal modification while preserving access to platform-specific features.

The project template system provides a standardized workflow for developing and integrating software modules across the heterogeneous platform ecosystem. Through a combination of filesystem organization standards, automated configuration scripts, and version control integration, the template enables developers to create modular *units* that can be composed into complex multi-agent systems spanning multiple hardware platforms.

III. Implementation

The implementation of the DUA framework is centered on two integrated components: the `dua foundation` collection of platform-specific base units [10] and the `dua-template` standardized development workflow [11]. Together, these components provide a comprehensive solution for developing and deploying software across heterogeneous robotic platforms.

A. Base unit construction and organization

The `dua-foundation` implements the hardware abstraction layer through a systematic collection of Docker images, each optimized for specific platform configurations. Base units follow a hierarchical naming convention that reflects both the target architecture and the intended functionality. Architecture prefixes distinguish between x86-64 systems (`x86`), ARM64 platforms (`armv8`) including Apple silicon devices, and Nvidia Jetson System-on-Chips (SoCs) (`jetson`), while functional suffixes indicate configuration scope: `base` units provide minimal configurations suitable for resource-constrained Single-Board Computers (SBCs), `dev` units include comprehensive development toolchains, and specialized variants such as `cudev` integrate CUDA libraries and AI frameworks for GPU-accelerated computation.

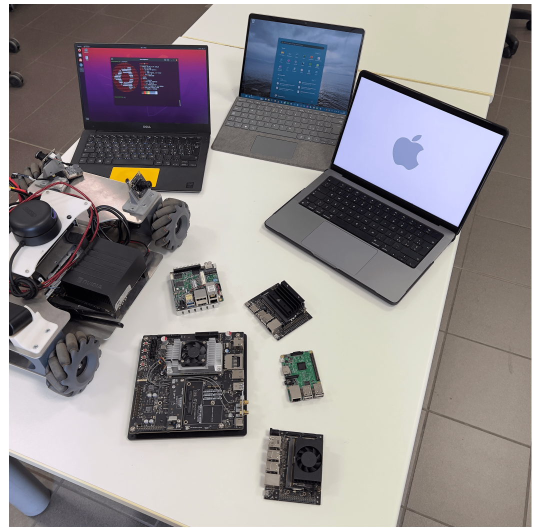


Fig. 1. Collection of DUA-supported platforms.

Each base unit construction process begins with a platform-appropriate Ubuntu Linux base image and systematically adds software dependencies through carefully ordered installation stages. Common dependencies include GCC compilation toolchains, Python development environments, and comprehensive middleware packages supporting robot communication protocols and software development tools. The ROS 2 middleware provides standardized communication primitives, while included DDS implementations (eProsima Fast DDS, Eclipse Cyclone DDS) and the new Zenoh protocol enable flexible communication patterns suited to different network conditions and performance requirements.

Platform-specific optimizations are integrated throughout the construction process to leverage unique hardware capabilities. `x86` images are configured for high-performance computing workloads, and `jetson` images integrate CUDA toolkits, cuDNN libraries, and machine learning frameworks including PyTorch, TensorFlow, and YOLO implementations. This manual configuration approach ensures that specialized hardware features remain accessible while maintaining consistent development interfaces across platforms. Figure 1 depicts a collection of hardware platforms currently supported by the DUA framework, while Table I lists the compressed sizes of the Docker images implementing the DUA base units.

B. Template system and modular integration

The `dua-template` provides a standardized structure for developing software modules (units) that can be seamlessly integrated across heterogeneous platforms. Implemented

Base unit	Image size
`armv8-base`	3.83 GB
`armv8-dev`	4.28 GB
`jetson6`	10.64 GB
`jetson5`	11.36 GB
`jetsonnano`	3.60 GB
`jetsontx2`	3.60 GB
`x86-base`	3.99 GB
`x86-cudev`	24.49 GB
`x86-dev`	4.42 GB

TABLE I

SIZES OF THE COMPRESSED DOCKER IMAGES OF THE BASE UNITS.

as a GitHub template repository, it establishes consistent filesystem organization and automated workflow management that significantly reduces configuration overhead in multi-agent system development.

Each unit follows a standardized filesystem layout, depicted in Figure 2, with dedicated directories for source code (`src`), configuration files (`config`), Docker specifications (`docker`), and other tools (`tools`). The template integrates with modern development environments, particularly Visual Studio Code, while remaining IDE-agnostic to accommodate diverse development preferences. Configuration management is handled through the `dua_setup.sh` script, which creates platform-specific targets by generating appropriate Dockerfiles and configuration files based on selected base units.

The modular integration approach enables seamless composition of sophisticated distributed architectures from independently-developed components. Units can be integrated into larger projects through Git submodules for active development scenarios or subtrees for stable dependency management. The template automatically merges system configurations and dependencies from multiple units into unified target containers, eliminating manual configuration conflicts that typically arise in complex architectures with multiple layers of integration.

Automated synchronization mechanisms maintain consistency across development teams through GitHub workflows that propagate template updates to derived projects. This approach ensures that improvements to the base framework automatically benefit all dependent projects while preserving project-specific customizations and configurations.

IV. APPLICATIONS AND PLATFORM SUPPORT

The practical viability of the Distributed Unified Architecture is demonstrated through platform coverage across diverse hardware architectures and successful deployment in real-world autonomous systems. This section provides empirical evidence of the framework's versatility and effectiveness in facilitating heterogeneous multi-agent coordination by the presentation of some prototypes developed and deployed with the framework.

A. Platform coverage and resource requirements

The DUA framework provides comprehensive support across a broad spectrum of hardware platforms that represent a wide range of computational capabilities encountered in contemporary robotic systems. The supported platforms encompass general-purpose development systems, resource-constrained SBCs, and specialized platforms featuring dedicated acceleration hardware. This extensive platform coverage enables seamless integration of autonomous agents possessing fundamentally different computational characteristics within unified multi-agent architectures.

The variation in base unit image sizes directly reflects the systematic adaptation to distinct platform requirements and computational capabilities. Minimal configurations designed for resource-constrained platforms, such as x86- or ARM64-based SBCs, require approximately 4 GB of storage capacity, while development-oriented images incorporate comprehensive toolchains and debugging capabilities. Specialized configurations for GPU-accelerated platforms integrate extensive machine learning frameworks and CUDA libraries, resulting in substantially larger images that provide complete development environments for artificial intelligence-enabled robotic applications. Platform-specific optimizations, discussed previously, ensure that unique hardware capabilities remain fully accessible while maintaining consistent programming interfaces across the framework. Finally, it is worth noting that, although the size of a base unit may be relatively large for some platforms, the Docker OverlayFS functionality allows to download it only a single time, sharing the image layers among an arbitrary number of development and deployment containers that may run on a single machine, effectively minimizing the amount of storage space required to use the framework for active development activities.

B. Early deployment scenarios

The framework's practical effectiveness has been validated through successful deployment across fundamentally different autonomous systems that demonstrate heterogeneous coordination capabilities. These implementations provide evidence of DUA's ability to enable rapid prototyping and maintain consistent deployment workflows across diverse robotic platforms while preserving platform-specific optimizations.

An autonomous Unitree Go2 quadruped robot, depicted in Figure 3 and equipped with an NVIDIA Jetson AGX Orin and a Stereolabs ZED 2i camera, demonstrating the framework's applicability to legged mobile platforms that require real-time perception and navigation capabilities. The agent performs autonomous environmental exploration for target object detection, integrating computer vision algorithms with locomotion control through the consistent development workflow provided by the DUA template system.

A complementary validation involves an autonomous coaxial octocopter, presented in Figure 4, that integrates an Nvidia Jetson Orin NX, a Stereolabs ZED Mini camera, and an Intel RealSense D435i depth camera. This implementation validates the framework's effectiveness for aerial platforms operating under stringent weight and power constraints.

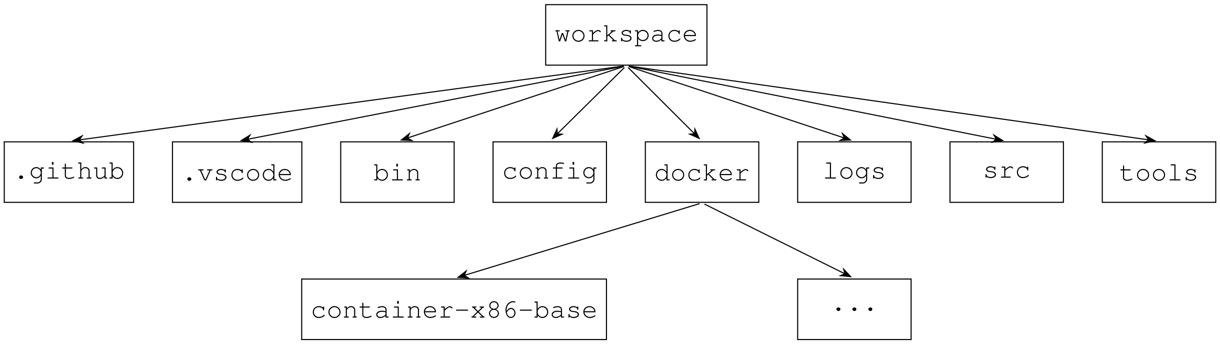


Fig. 2. Directory tree of a DUA unit based on `dua-template`.

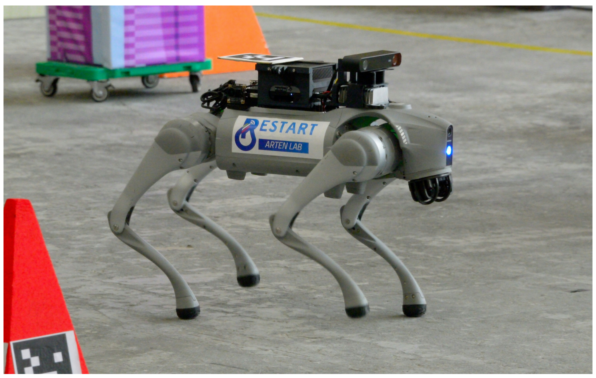


Fig. 3. Autonomous Unitree Go2 quadruped robot equipped with an NVIDIA Jetson AGX Orin and a Stereolabs ZED 2i camera that explores an environment in search of target objects.

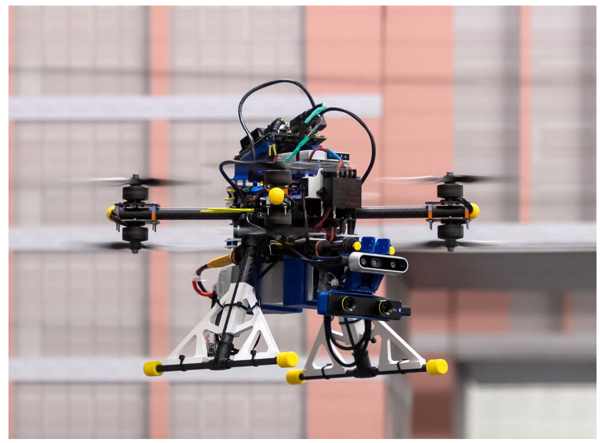


Fig. 4. Autonomous coaxial octocopter, integrating an Nvidia Jetson Orin NX, a Stereolabs ZED Mini, and an Intel RealSense D435i depth camera, explores and navigates through an unknown environment.

The agent executes autonomous exploration and navigation through unknown environments, demonstrating how the DUA containerization approach enables deployment of sophisticated perception and planning algorithms on resource-constrained aerial platforms, also easing the integration effort represented by the many different sensor and embedded hardware involved.

These diverse implementations, spanning terrestrial and aerial operational domains with distinct sensor configurations and computational requirements, provide empirical validation of DUA's fundamental value proposition of enabling seamless software deployment across heterogeneous agent teams. The consistent development and deployment workflows provided by the framework substantially reduce the integration overhead typically associated with coordinating diverse autonomous systems in multi-agent operational scenarios.

V. Conclusions

This paper presents the Distributed Unified Architecture, a comprehensive framework that addresses fundamental software infrastructure challenges in heterogeneous multi-agent robotic systems. The DUA framework successfully demonstrates that containerization-based approaches can provide effective hardware abstraction while preserving the platform-specific capabilities essential for optimal system performance. Real-world deployment validation across autonomous terrestrial and aerial platforms provides evidence of the framework's practical effectiveness. The successful integration of diverse sensor configurations and computational requirements within unified development workflows demonstrates that DUA achieves its fundamental objective of enabling seamless coordination between heterogeneous robotic teams. The framework's ability to maintain near-native performance while providing consistent interfaces represents a significant achievement in distributed autonomous system development capabilities.

Future research directions will extend the framework's applicability to encompass microcontroller-based agents, enabling comprehensive integration from high-level planning algorithms to low-level actuation systems. The implementation of cross-compilation capabilities and network-based debugging tools will further enhance development workflows for distributed multi-agent systems. Additionally, the development of automated testing frameworks specifically designed for complex collaborative behaviors will facilitate validation and verification processes essential for deploying multi-agent systems in safety-critical applications.

References

[1] Y. Meng and J. Gan, "A distributed swarm intelligence based algorithm for a cooperative multi-robot construction task," in *2008 IEEE Swarm Intelligence Symposium*, pp. 1–6, Sept. 2008.

[2] J. Werfel, "Collective Construction with Robot Swarms," in *Morphogenetic Engineering: Toward Programmable Complex Systems* (R. Doursat, H. Sayama, and O. Michel, eds.), pp. 115–140, Berlin, Heidelberg: Springer, 2012.

[3] L. Bianchi and others, "Efficient visual sensor fusion for autonomous agents," in *7th International Conference on Control, Automation and Diagnosis*, pp. 01–06, IEEE, 2023.

[4] L. Bianchi and others, "A novel distributed architecture for unmanned aircraft systems based on Robot Operating System 2," *IET Cyber-Systems and Robotics*, vol. 5, no. 1, p. e12083, 2023.

[5] S. Macenski and others, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[6] S. Macenski, T. Moore, D. V. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2," *Robotics and Autonomous Systems*, vol. 168, p. 104493, Oct. 2023.

[7] E. Cervera, "Try to Start It! The Challenge of Reusing Code in Robotics Research," *IEEE Robotics and Automation Letters*, vol. 4, pp. 49–56, Jan. 2019.

[8] F. Lumpp, M. Panato, N. Bombieri, and F. Fummi, "A Design Flow Based on Docker and Kubernetes for ROS-based Robotic Software Applications," *ACM Transactions on Embedded Computing Systems*, vol. 23, pp. 74:1–74:24, Aug. 2024.

[9] R. Masocco, *A unified software architecture for distributed autonomous systems*. PhD thesis, Tor Vergata University of Rome, Rome, Italy, 2025.

[10] R. Masocco and others, "dua-foundation." `https://github.com/dotX-Automation/dua-foundation`, 2024.

[11] R. Masocco and others, "dua-template." `https://github.com/dotX-Automation/dua-template`, 2024.

[12] S. Saxena, H. E. Z. Farag, and N. El-Taweel, "A distributed communication framework for smart Grid control applications based on data distribution service," *Electric Power Systems Research*, vol. 201, p. 107547, Dec. 2021.

[13] R. Masocco, A. Tenaglia, C. Galperti, D. Carnevale, S. Galeani, and others, "A distributed real-time diagnostic and control network for the TCV tokamak based on the Data Distribution Service," in *33rd Symposium on Fusion Technology (SOFT 2024)*, Sept. 2024.

[14] C. Galperti, F. Felici, T. Vu, and others, "Overview of the TCV digital real-time plasma control system and its applications," *Fusion Engineering and Design*, vol. 208, p. 114640, Nov. 2024.

[15] J. M. Schlesselman, G. Pardo-Castellote, and B. Farabaugh, "OMG data-distribution service (DDS): architectural update," in *Proc. of the 2004 IEEE Military Communications Conference (MILCOM)*, vol. 2, pp. 961–967, Oct. 2004.

[16] G. Pardo-Castellote, "OMG Data-Distribution Service: architectural overview," in *Proc. of the 23rd International Conference on Distributed Computing Systems*, pp. 200–206, May 2003.

[17] A. Corsaro, L. Cominardi, O. Hecart, G. Baldoni, J. E. P. Avital, J. Loudet, C. Guimares, M. Ilyin, and D. Bannov, "Zenoh: Unifying Communication, Storage and Computation from the Cloud to the Microcontroller," in *2023 26th Euromicro Conference on Digital System Design (DSD)*, pp. 422–428, Sept. 2023.

CE-MRS: Contrastive Explanations for Multi-Robot Systems

Ethan Schneider[1], Daniel Wu, Devleena Das, Sonia Chernova[1]

Abstract—**As the complexity of multi-robot systems grows to incorporate a greater number of robots, more complex tasks, and longer time horizons, the solutions to such problems often become too complex to be fully intelligible to human users. In this work, we introduce Contrastive Explanations for Multi-Robot Systems (CE-MRS), a framework for generating natural language explanations that justify the validity of the system's solution to the user, or else aid the user in correcting any errors that led to a suboptimal system solution. Toward this goal, we first contribute a generalizable formalism of contrastive explanations for multi-robot systems, and then introduce a holistic approach to generating contrastive explanations for multi-robot scenarios that selectively incorporates data from multi-robot task allocation, scheduling, and motion-planning to explain system behavior. Through user studies with human operators, we demonstrate that our integrated contrastive explanation approach leads to significant improvements in user ability to identify and solve system errors, leading to significant improvements in overall multi-robot team performance.**

Index Terms—**Design and Human Factors, Human Factors and Human-in-the-Loop, and Multi-Robot Systems**

I. INTRODUCTION

Heterogeneous multi-robot systems (MRS) offer the possibility to solve complex problems in a variety of industries and environments [3] by solving the *task allocation*, *scheduling*, and *motion planning* subproblems. However, as the complexity of MRS increases, solutions to such problems becomes obscure to end-users. Despite this, operators are routinely tasked to validate solutions prior to deployment [1], [2]. The increasing complexity of black-box models is not unique to MRS and is common among machine learning and artificial intelligence research, leading to the emergence of Interpretable Machine Learning (IML) [7] and Explainable AI (XAI) [6] subfields which seek to provide a human-interpretable explanation for the decision making of complex models.

Research in social sciences has shown that humans seeking an explanation for complex phenomena typically utilize a *contrastive* approach [15], [12], in which humans probe the solution by asking about some other expected outcome. Human studies show that when providing explanations in such cases to each other, humans provide partial explanations instead of full ones. Instead humans focus explanations on the key factors that caused the given output instead of another [19].

We are interested in providing explanations that aid operators in interpreting solutions generated by complex MRS that incorporate task allocation, scheduling, and motion planning

[1]The authors are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA {eschneider32, chernova}@gatech.edu

This work was supported by Army Research Laboratory under Grants W911NF-17-2-0181 (DCIST CRA).

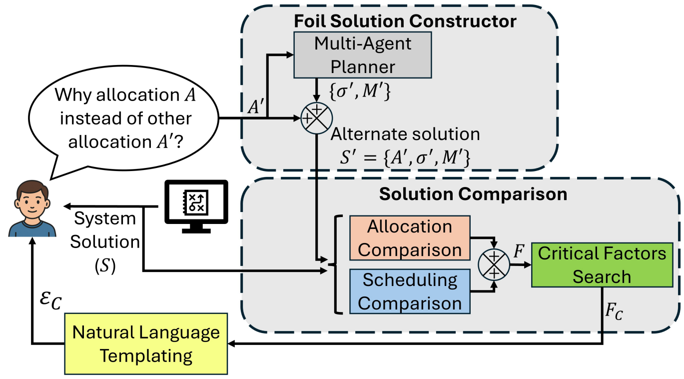


Fig. 1: CE-MRS Framework Diagram

into its decision making. While prior XAI work has addressed explanations for task allocation [21], [20], scheduling [13], [5], and motion planning [8] independently, recent work has shown the close *interdependency* between these three subproblems [18], [14], [17]. We argue that explanations must have the ability to incorporate information across subproblems in order to most accurately represent the system's decision making to the operator.

Our work makes the following contributions. First, we formalize contrastive explanations for MRS. Second, we contribute a holistic approach to generating contrastive explanations for multi-robot scenarios through our framework, Contrastive Explanations for Multi-Robot Systems (CE-MRS), which utilizes select information from the multi-robot task allocation, scheduling, and motion planning sub-problems. Finally, we validate our approach in a 22-participant in-person user study in a simulated search-and-rescue domain. Our results show that by utilizing information from all three subproblems, we can better explain the reasoning for multi-robot solutions.

II. PROBLEM FORMULATION

In this section, we formalize multi-robot contrastive explanations and the multi-robot problem and its solution.

A. Contrastive Explanations for Multi-Agent Planning

A contrastive explanation, $\mathcal{E}_S$, provides an answer to the question *"Why P and not Q?"*, where P represents the algorithmic solution, and Q represents the user's alternative suggestion or *foil* [16] (e.g. *"Why is robot1 performing task t instead of robot2?"*). We formulate $\mathcal{E}_S$ in relation to a team of heterogeneous robots, $R = \{r_1, r_2, ...r_n\}$, cooperating to execute a set of tasks, $T = \{t_1, t_2, ...t_m\}$, within a particular problem domain $\mathcal{D}$. Given $\mathcal{D}$, planning algorithm $\mathbb{A}$ solves $\mathcal{D}$ under constraints τ (time, utility, etc.), producing a solution

$\mathcal{S}$ ($\mathbb{A} : \mathcal{D} \times \tau \rightarrow \mathcal{S}$). Consistent with prior work, we define a solution to the problem as $\mathcal{S} = \langle \mathcal{A}, \sigma, \mathcal{M} \rangle$, where $\mathcal{A} : \mathcal{R} \rightarrow T$ is the allocation mapping robots to tasks, σ is the task schedule, and $\mathcal{M}$ is the set of motion plans required for task execution [18]. Our objective is to generate a natural language explanation that justifies the validity of the solution to the user, or else aids the user in correcting any errors that led to a suboptimal solution.

We consider a human operator whose task is to evaluate the quality of $\mathcal{S}$ prior to execution. Validating $\mathcal{S}$ requires the user to reason about i) robot capabilities, ii) task requirements, and iii) scheduling constraints. We posit that a meaningful explanation of $\mathcal{S}$, $\mathcal{E}_\mathcal{S}$, should consider these factors.

In order to elicit an explanation from the system, we enable a human operator to specify a contrastive example, or *foil*, that represents a foil solution, $\mathcal{S}'$. In this work, we represent the foil as an alternate task allocation $\mathcal{A}'$, from which an alternative schedule, σ', and motion plan, $\mathcal{M}'$, are derived using $\mathbb{A}$. Our work directly generalizes to foils relating to alternate schedules or motion plans, as in [4], as our explanations address each of these (Sec. III).

Given $\mathcal{S}'$, we define the contrastive explanation $\mathcal{E}_\mathcal{S}$ under two scenarios:

1) $\mathbb{A} : \mathcal{D} \times \tau \not\rightarrow \mathcal{S}'$ If $\mathcal{S}'$ is infeasible, $\mathcal{E}_\mathcal{S}$ provides information about the cause. We define a solution $\mathcal{S}$ to be infeasible if the solution: i) assigns a robot to a task without meeting the task's minimum trait requirements, ii) violates a precedence constraint, or iii) has no valid motion plan solution.
2) $\mathbb{A} : \mathcal{D} \times \tau \rightarrow \mathcal{S}'$, but $\mathcal{S} \equiv \mathcal{S}'$ or $\mathcal{S} > \mathcal{S}'$ In this case case, the user provides a feasible $\mathcal{S}'$ and $\mathcal{E}_\mathcal{S}$ utilizes elements from $\mathcal{S}$ as reasoning for why $\mathcal{S} \equiv \mathcal{S}'$ or $\mathcal{S} > \mathcal{S}'$.

B. Multi-Robot Planning Problem Description

To formulate a solution explanation $\mathcal{E}_\mathcal{S}$, we require a formal definition of the multi-robot problem and its solution. We borrow the MRS formulation described in Neville et al. [18].

A multi-robot system provides a solution $\mathcal{S}$ to a particular problem domain $\mathcal{D} = \langle \mathcal{Q}, \phi, \mathcal{T}, Y^* \rangle$, where

- $\mathcal{Q} \in \mathbb{R}_+^{N \times U}$ is the robot trait matrix describing the u^{th} trait for the n^{th} robot.
- $\phi \in \mathbb{R}_+^N$ represents the speeds for N robot types.
- $\mathcal{T}$ is the task network, defined as a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where each vertex in $\mathcal{V}$ is some task, $t_i \in T$. An edge in $\mathcal{E}$ connects two vertices in $\mathcal{V}$, shown by, $e = (t_i, t_j)$, where, $t_i, t_j \in \mathcal{V}$. This edge defines a precedence constraint, $t_i \prec t_j$, where task t_i must be completed before task t_j can start.
- $Y^* \in \mathbb{R}_+^{M \times U}$ is the desired trait matrix describing the required u^{th} trait for the m^{th} task.

Given $\mathcal{D}$, a multi-robot planner finds solution $\mathcal{S} = \langle \mathcal{A}, \sigma, \mathcal{M} \rangle$, where

- $\mathcal{A} \in \mathbb{R}^{M \times N}$ is the task allocation matrix, in which robot n is allocated to task m iff $A_m^n = 1$, otherwise $A_m^n = 0$.
- σ is the schedule which assigns a start and end time for each task in $\mathcal{T}$, in addition to ensuring precedence constraints in $\mathcal{T}$ and mutex constraints in $\mathcal{M}$ are met.
- $\mathcal{M}$ is the motion plan which represents a finite set of motion plans for each robot with their assigned tasks defined in $\mathcal{A}$.

The search space for $\mathcal{S}$ is large, spanning hundreds of potential allocations [14]. Our objective is to provide explanations to the user that reveals why $\mathcal{S}$ is a valid solution. Alternately, if errors exist in $\mathcal{D}$ (e.g. errors in the task network specification), $\mathcal{E}_\mathcal{S}$ should help the operator to identify those errors. In the following section we discuss our approach for generating such explanations.

III. CE-MRS Framework

Figure 1 presents an overview of the CE-MRS framework that provides contrastive explanations, $\mathcal{E}_\mathcal{S}$, for a multi-robot system's solution $\mathcal{S}$, given a problem domain $\mathcal{D}$. The explanation process begins with a human operator, who receives $\mathcal{S}$ for a given $\mathcal{D}$. If the solution is unclear, the user can pose a foil in the form of $\mathcal{A}'$, an alternative task allocation. CE-MRS then provides $\mathcal{E}_\mathcal{S}$ which compares the user's proposed solution $\mathcal{S}'$ (based on $\mathcal{A}'$) with $\mathcal{S}$. Below we provide further detail about each module within CE-MRS.

A. Constructing the Foil Solution

For a given solution $\mathcal{S}$, the user can ask foils about $\mathcal{A}$ in the form, *"Why is r_i not assigned to t_j?"* where r_i represents a given robot and t_j represents a particular task. The user can pose one or more of these questions, and the set of questions constitute $\mathcal{A}'$. To generate a counterfactual solution $\mathcal{S}'$, CE-MRS leverages $\mathcal{A}'$ to automatically construct foil schedule σ' and foil motion plan $\mathcal{M}'$. We utilize the ITAGS algorithm [18] to generate $\mathcal{S}$ and $\mathcal{S}'$, although CE-MRS can be adapted to work with other representations and planners. Additionally, while we restrict the foil to $\mathcal{A}'$, CE-MRS easily generalizes to schedule and motion planning foils, σ' and $\mathcal{M}'$.

B. Solution Comparison

Given $\mathcal{S}$ and $\mathcal{S}'$, CE-MRS computes the set of factors F through which $\mathcal{S}$ and $\mathcal{S}'$ differ. Specifically, CE-MRS computes differences in task allocations ($\mathcal{A}$ vs $\mathcal{A}'$) and schedules (σ vs σ'). For complex MRS, $|F|$ may be large, and it is important to ensure that $\mathcal{E}_\mathcal{S}$ only includes the most important factors to improve user understanding [11], [9]. Given these findings, the solution comparison module employs a threshold to find the subset $F_C \subseteq F$ that represents the most important factors through which $\mathcal{S}$ and $\mathcal{S}'$ differ. Below we detail the allocation comparison, scheduling and motion planning comparison, and the critical factors filtering algorithm.

Allocation Comparison: Recall that $\mathcal{A}$ represents a $M \times N$ task allocation matrix where M represents the number of tasks and N represents the number of robots. The allocation comparison algorithm performs a column-wise comparison between $\mathcal{A}_m$ and $\mathcal{A}'_m$ to find $F_\mathcal{A}$, the allocation-related factors in which $\mathcal{A}_m$ and $\mathcal{A}'_m$ differ. Specifically, when $\mathcal{A}_m \neq \mathcal{A}'_m$, $F_\mathcal{A} = (m, n, n') \forall m \in \{1, ..., M\}$, where m is the allocated task, n is the allocated robot in $\mathcal{A}$, and n' is the allocated robot in $\mathcal{A}'$.

Scheduling and Motion Planning Comparison: The scheduling comparison algorithm considers the percent differences between σ and σ' in terms of overall makespan λ, task time β_m where $m \in \{1, ..., M\}$, and robot makespan α_n where $n \in \{1, ..., N\}$. For all M tasks and N robots, the schedule-related factors are defined as $F_\sigma = \{\lambda, \beta, \alpha\}$, in which $\beta = \{\beta_0..\beta_M\}$ and $\alpha = \{\alpha_0..\alpha_N\}$.

Critical Factors Filtering: From F, we perform empirically-driven thresholding to extract the top contributing factors $F^C \subseteq F$. Recall that $F = \{F_{\mathcal{A}}, F_\sigma\}$, denoting factors related to differences between task allocation and scheduling, respectively. In this manner, $F^C = \{F^C_{\mathcal{A}}, F^C_\sigma\}$, where $F^C_{\mathcal{A}} \subseteq F_{\mathcal{A}}$ and $F^C_\sigma \subseteq F_\sigma$.

To find F^C_σ, we define a threshold, Z, such that any value of a factor $f_\sigma \in F_\sigma$ above Z is determined to be a critical factor. In our work, Z is computed through an ablation study in which we simulate multiple user foils with a varying value of Z, and find the point at which the rate of change decreases significantly (in our case, $Z = 0.1$). Additionally, we set $F^C_a = F_a$ and consider all allocation-related factors as critical factors.

C. Explanation Construction

Given F^C, the explanation construction module templates $F^C_{\mathcal{A}}$ and F^C_σ into a natural language explanation $\mathcal{E}_{\mathcal{S}}$. We template $F^C_{\mathcal{A}}$ by constructing a sentence that states a given task m is capable of being worked by robots n and n', in which the task requirement Y^*_m and robot traits $\mathcal{Q}_n$ and $\mathcal{Q}_{n'}$ are appended. Similarly, we template F^C_σ by enumerating the percent differences in $\{\lambda, \beta, \alpha\}$, in which it reveals ϕ_n as a possible reason for why β_m is significantly different between robots n and n'. Unlike prior work which explains either why $\mathcal{S} > \mathcal{S}'$ [21] (optimality) or why some agent would reject $\mathcal{S}'$ [20] (feasibility), we address both feasibility and optimality for some objective variable in $\mathcal{E}_{\mathcal{S}}$.

IV. Experimental Design

Our study objectives are to answer two research questions:

- *RQ1: Can human operators detect when a multi-robot planning solution is incorrect?* This question is critical because users are only likely to engage with the explanation system, if they can independently detect unexpected results. Thus, we first assess participant ability to correctly assess plan solution quality.
- *RQ2: How well do contrastive explanations enable human operators to identify and correct any errors within the multi-robot problem domain specification?* We examine the scenario in which some other (hypothetical) human team member encodes the domain description, $\mathcal{D}^* = \langle \mathcal{Q}, \phi, \mathcal{T}, Y^* \rangle$. In our study, we corrupt $\mathcal{D}^* \rightarrow \mathcal{D}$ in some subset of study scenarios, and then measure the participants' performance in identifying and correcting $\mathcal{D}$ such that $\mathcal{D} = \mathcal{D}^*$.

To assess the research questions, we conducted a two-way, between subjects study in which participants analyzed multiple simulated emergency-response scenarios inspired by [10] and [22]. For each scenario, the participant was first asked whether the proposed plan solution was sound (RQ1). If the participant believed the solution was sound, they were allowed to move on to the next scenario. Alternately, if the participant felt that errors were likely, they were allowed to engage with the explanation system and address any perceived errors in $\mathcal{D}$ (RQ2). Once complete, participants could either state that they believed the final solution was correct ($\mathcal{D} == \mathcal{D}^*$), or that errors remained but they did not know how to fix them. Participants were randomly split into two conditions based on the type of explanation received:

1) **CE-MRS (ours)**: participants received explanations from the CE-MRS Framework, detailing whether $\mathcal{S}'$ is feasible, and summarizing the top contributing factors through which $\mathcal{S}$ and $\mathcal{S}'$ differ.
2) **CMAoE (baseline)**: received contrastive tabular explanations, as in [21], in which a user asks "why does S not enforce property P", and CMAoE generates S' where P is enforced, while minimizing the difference between S and S'.

A. Study Design

Each participant was asked to complete 6 independent scenarios, in which we maintained the number of robots $r = 4$ and tasks $k = 7$ across the scenarios, while varying the types/number of errors injected in $\mathcal{D}$. Robots used in the study were 1 dumptruck, 2 firetrucks, and 1 ambulance, each with their own traits. Tasks included in the study were 1 large debris, 2 small debris, 2 rescue humans, 1 setup camp, and 1 defuse bomb. The initial robot and task configurations differed in each scenario, leading to different solutions.

The study was performed in two stages, familiarization and assessment. During familiarization, users received a tutorial about $\mathcal{D}$, how to read their assigned study condition explanations, and completed a 5-question quiz to validate their understanding. During assessment, participants were presented with six scenarios in a randomized order, in which users first familiarized themselves with $\mathcal{S}$ and were asked whether they thought $\mathcal{D} = \mathcal{D}^*$ (RQ1). If the participant suspected errors were present, they then leveraged the explainability interface to generate counterfactual explanations, and were able to make corrections to the underlying problem specification $\mathcal{D}$ (RQ2). Users could update $\mathcal{D}$ until they either believed $\mathcal{D} = \mathcal{D}^*$ or they gave up by indicating that errors remained but they did not know how to fix them.

B. Metrics

We utilised the following metrics to evaluate our study:

1) **User Scenario Classification Prior to Explanations (*IDP*)**: Users' accuracy score for identifying if $\mathcal{D}$ is valid prior to viewing explanations.
2) **Remaining Robot Trait Errors (*RTE%*)**: Percentage of errors in the robot trait matrix ($\mathcal{Q}$) that remain unresolved at the end of the scenario.
3) **Remaining Task Requirement Errors (*TRE%*)**: Percentage of errors within the desired trait matrix (Y^*) that remain unresolved at the end of the scenario.
4) **Remaining Robot Speed Error (*RSE%*)**: Percentage of errors within the robot speeds (ϕ) that remain unresolved at the end of the scenario.
5) **User Efficiency (*UE*)**: Ratio between the number of repair actions and the total remaining errors, where repair attempts refers to changes made to $\mathcal{D}$; this metric is an approximation for user effort.

6) **User Scenario Classification After Explanations (*IDA*)**: Users' accuracy score for identifying if $\mathcal{D}$ is valid after viewing explanations.

C. Participants

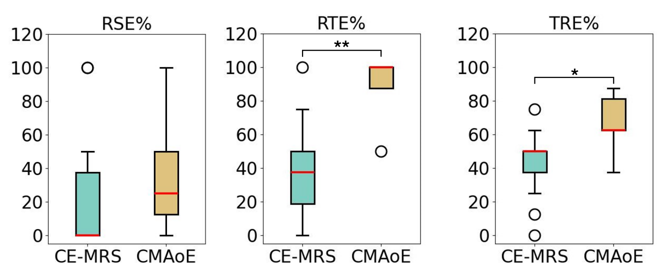
We recruited 22 participants from a US university. Participants were randomly placed into a study condition, resulting in 11 participants in each study condition. Our participants included 17 males, 9 females, and 1 non-disclosed, all of whom are over the age of 18 (M=24.7, SD=2.29). The study took 1hr - 1.5hrs, and participants were compensated $15.

V. Results

Given the *RTE%*, *TRE%*, and *RSE%* metrics do not follow a normal distribution (Shapiro-Wilk's Test, $p < 0.05$), we analyze statistical significance between study conditions using the non-parametric Wilcoxon rank-sum test. We also provide a qualitative analysis of participants' **IDP**, **UE**, and **IDA** scores.

We first evaluate the **IDP** metric to determine whether users can correctly identify $\mathcal{D} \neq \mathcal{D}^*$ when given $\mathcal{S}$. Given that $\mathcal{S}$ is presented with no accompanying explanations, we expect little difference in **IDP** scores across conditions. Our analysis confirms this hypothesis, with IDP values of 69.7% and 77.3% for CMAoE and CE-MRS, respectively, showing similar ability to assess solution accuracy across study conditions.

Next, we examine user effectiveness in resolving errors in $\mathcal{D}$. Figure 2 presents remaining error percentage in $\mathcal{D}$ for scenarios in which $\mathcal{D} \neq \mathcal{D}^*$. We examine three error types, **RSE%**, **RTE%**, **TRE%**, which relate to errors in ϕ, $\mathcal{Q}$, and Y^*, respectively. Lower values correspond to better performance, as the user has corrected more errors in $\mathcal{D}$.



(a) (b) (c)

Fig. 2: RSE%, RTE%, and TRE% metrics per study condition, in which a lower value is better. Statistical significance is reported as: * p<0.01, ** p<0.001

In Figure 2a, we observe no statistical difference in participants' ability to identify and correct errors in ϕ between CE-MRS and CMAoE. For the **RTE%** and **TRE%** metrics (Figure 2b and Figure 2c), we observe that CE-MRS significantly outperforms CMAoE in assisting users to identify and correct $\mathcal{Q}$ errors, **RTE%** ($p < 0.001$), and Y^* errors, **TRE%** ($p < 0.01$). This is because CE-MRS selectively reveals information from the task allocation, scheduling, and motion planning sub-modules to explain the system's reasoning for a particular solution, resulting in a higher user understanding of the system's solution, $\mathcal{S}$.

Figure 3 visualizes the **UE** metric separated by study condition for how efficient users are in correcting errors in $\mathcal{D}$. In this graph, the dashed line represents a perfectly efficient user that makes no redundant changes to $\mathcal{D}$. Data points above the line correspond to scenarios in which users make more repair attempts than errors corrected. Scenarios with more corrected errors have the least Final Remaining Errors (lower x-axis values). We observe that users using CE-MRS

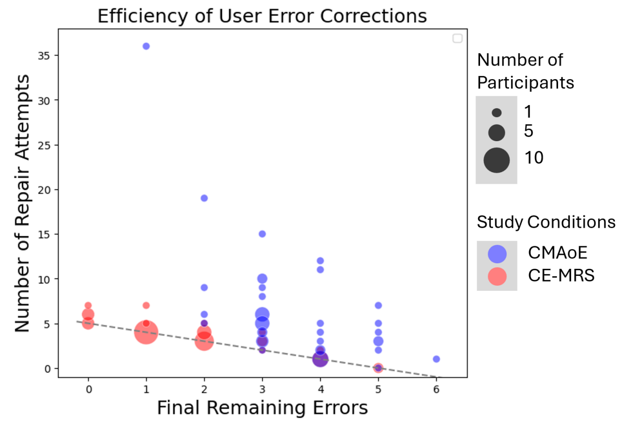


Fig. 3: Efficiency of users' error corrections of $\mathcal{D}$. Points further left are scenarios with more errors corrected; points closer to the line represent more efficient corrections.

follow the dashed line closer than participants given CMAoE explanations, indicating that our CE-MRS explanations help users correct errors in $\mathcal{D}$ more efficiently. This highlights the importance of selectively revealing information from the system input, $\mathcal{D}$, when comparing $\mathcal{S}$ and $\mathcal{S}'$ or explaining a solution's feasibility.

Lastly, we evaluate whether participants were aware that $\mathcal{D} = \mathcal{D}^*$ or $\mathcal{D} \neq \mathcal{D}^*$ as measured by the **IDA** metric. Our results show that participants showed similar ability to evaluate solution accuracy across conditions, with **IDA** values of 22.7% and 18.2% for CMAoE and CE-MRS, respectively. We conclude that this result is due to neither CE-MRS or CMAoE is designed to evaluate optimality of $\mathcal{S}$, so other techniques should be designed to address this challenge.

VI. Conclusions & Future Work

In summary, our results demonstrate that CE-MRS significantly improves a human operator's ability to identify and resolve errors in $\mathcal{D}$ relating to robot capabilities (RTE%) and task requirements (TRE%), while performing on-par with prior methods on resolving errors in robot capabilities (RSE%). Participants exposed to CE-MRS explanations were also more efficient in resolving errors, with CE-MRS participants making only 1.33 ± 0.60 extraneous corrections, on average, compared to 5.24 ± 5.63 extraneous corrections in CMAoE.

While we demonstrate strong advances in improving the interpretability of MRS, there are numerous opportunities for future work. First, we determined the threshold Z through an ablation study for our domain; future work should explore more generalizable methods for filtering the factors F to the critical factors F^C. Next, future work should investigate how to improve operator ability to assess the correctness of a solution. Finally, more longitudinal field studies are needed to assess explanability techniques for multi-robot systems.

REFERENCES

[1] J. Adams, J. Hamell, and P. Walker, "Can A Single Human Supervise A Swarm of 100 Heterogeneous Robots?" *JFR*, 2023.

[2] J. A. Adams, J. Scholtz, and A. Sciarretta, "Human–Robot Teaming Challenges for the Military and First Response," *Annual Review of Control, Robotics, and Autonomous Systems*, 2023.

[3] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, "Multiple Mobile Robot Task and Motion Planning: A Survey," *ACM Computing Surveys*, 2023.

[4] M. Brandao, M. Mansouri, A. Mohammed, P. Luff, and A. Coles, "Explainability in multi-agent path/motion planning: User-study-driven taxonomy and requirements," in *AAMAS*, 2022.

[5] K. Čyras, D. Letsios, R. Misener, and F. Toni, "Argumentation for explainable scheduling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 2752–2759.

[6] M. Fox, D. Long, and D. Magazzeni, "Explainable Planning," *CoRR*, 2017.

[7] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," in *DSAA*, 2018.

[8] B. Gyevnar, C. Wang, C. G. Lucas, S. B. Cohen, and S. V. Albrecht, "Causal explanations for sequential decision-making in multi-agent systems," ser. AAMAS, 2024, p. 771–779.

[9] A. Hudon, T. Demazure, A. Karran, P.-M. Léger, and S. Sénécal, "Explainable Artificial Intelligence (XAI): How the Visualization of AI Predictions Affects User Cognitive Load and Confidence," in *NeuroIS Retreat*, 2021.

[10] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada, "RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research," in *SMC*, 1999.

[11] V. Lai, Y. Zhang, C. Chen, Q. V. Liao, and C. Tan, "Selective Explanations: Leveraging Human Input to Align Explainable AI," *PACHMCI*, 2023.

[12] P. Lipton, "Inference to the Best Explanation," *A Companion to the Philosophy of Science*, 2017.

[13] J. Ludwig, A. Kalton, and R. Stottler, "Explaining Complex Scheduling Decisions," ser. CEUR Workshop Proceedings, 2018.

[14] A. Messing, G. Neville, S. Chernova, S. Hutchinson, and H. Ravichandar, "GRSTAPS: Graphically Recursive Simultaneous Task Allocation, Planning, and Scheduling," *IJRR*, 2022.

[15] T. Miller, "Explanation in Artificial Intelligence: Insights from the Social Sciences," *AIJ*, 2019.

[16] ——, "Contrastive Explanation: A Structural-Model Approach," *KER*, 2021.

[17] G. Neville, S. Chernova, and H. Ravichandar, "D-ITAGS: A Dynamic Interleaved Approach to Resilient Task Allocation, Scheduling, and Motion Planning," *RA-L*, 2023.

[18] G. Neville, A. Messing, H. Ravichandar, S. Hutchinson, and S. Chernova, "An Interleaved Approach to Trait-Based Task Allocation and Scheduling," in *IROS*, 2021.

[19] D.-H. Ruben, *Explaining Explanation*. Routledge, 2015.

[20] Z. Zahedi, S. Sengupta, and S. Kambhampati, "'Why didn't you allocate this task to them?' Negotiation-Aware Explicable Task Allocation and Contrastive Explanation Generation," in *AAMAS*, 2023.

[21] P. Zehtabi, A. Pozanco, A. Bolch, D. Borrajo, and S. Kraus, "Contrastive Explanations of Centralized Multi-agent Optimization Solutions," in *ICAPS*, 2024.

[22] W. Zhao, Q. Meng, and P. W. H. Chung, "A Heuristic Distributed Task Allocation Method for Multivehicle Multitask Problems and Its Application to Search and Rescue Scenario," *IEEE Transactions on Cybernetics*, 2016.

Decentralized Multi-Agent Task Assignment for Resource-Constrained Robotic Operations

Rui de Gouvea Pinto[1] and Jonathan Rogers[2]

Abstract— **We present Resource-Aware CBBA, an extension to the Consensus-Based Bundle Algorithm (CBBA) that explicitly incorporates operational constraints such as limited battery life and payload capacities. By embedding predictive resource consumption models and replenishment planning into the task bidding process, our method generates feasible, efficient task allocations under real-world conditions. Simulations show that Resource-Aware CBBA significantly outperforms naive approaches in scenarios with tight resource constraints and high replenishment delays. In particular, task environments with distance-correlated rewards or trade-offs between base station proximity and recharge efficiency benefit substantially. Additionally, our method consistently achieves near-optimal performance in small-scale scenarios, highlighting its potential for real-world deployment.**

I. INTRODUCTION

Autonomous robotic systems are increasingly being considered for structured field applications such as construction site preparation, where tasks like grading, trench shaping, and material redistribution must be performed reliably and efficiently over large, partially known environments. The use of multi-robot teams in these contexts offers the potential for faster execution and improved fault tolerance. However, coordinating such teams under real-world constraints remains a significant challenge.

In practical scenarios, robotic agents must divide tasks while simultaneously considering operational constraints like finite battery life, limited payload capacity, and downtime associated with resource replenishment. These factors become critical in sustained or long-range operations, requiring agents to alternate between task execution and visits to base stations for recharging or payload offloading.

The Consensus-Based Bundle Algorithm (CBBA) [1] is a widely-used decentralized task allocation method known for its scalability, minimal communication needs, and proven convergence guarantees. However, classical CBBA assumes that agents can continuously execute tasks without interruptions for resource replenishment—an assumption that fails under realistic deployment conditions.

Prior works have partially addressed specific constraints, such as task complexity [2] and timing requirements [3], [4], or considered resource constraints without integrated replenishment planning [3]. Solutions addressing replenishment typically handle it separately through additional planning layers or market-based mechanisms [5], [6], leaving a significant gap for fully decentralized, integrated approaches.

To bridge this gap, we propose *Resource-Aware CBBA*, which extends classical CBBA by incorporating predictive resource consumption models directly into the bidding phase. Our method enables agents to proactively integrate replenishment events into their task plans, generating more feasible and efficient multi-agent assignments under realistic resource constraints.

II. PROBLEM FORMULATION

We consider a multi-agent robotic system tasked with autonomous site preparation for modular construction. We assume a fleet of N homogeneous robotic agents assigned to complete a set of M tasks, where typically $M > N$. Tasks represent specific construction activities such as trench shaping, grading, or bedding layer preparation, and are characterized by their location, payload requirements, and execution times.

Formally, task assignment is represented by a mapping $\pi : \{1, \ldots, M\} \rightarrow \{1, \ldots, N\}$, allocating tasks to agents. Each agent i executes an ordered sequence of tasks $T_i = (t_{i,1}, t_{i,2}, \ldots)$, referred to as its *path*, which specifies the order of task execution. This path is interspersed with resource replenishment at predefined base stations. When an agent performs a resource replenishment, they must stay at the base station for some fixed amount of time, hereafter referred to as the replenishment delay. This delay represents the time it would take for payload to be refilled or emptied, or the time for the agent's battery to be swapped.

Each task is associated with a nominal reward r_{task}. To encourage rapid task completion, the reward earned by an agent, r_{earned}, is discounted based on completion time t:

$$r_{earned} = r_{task} \cdot \gamma^t \tag{1}$$

where $\gamma \in (0, 1)$ is a tunable discount factor.

We utilize the Consensus-Based Bundle Algorithm (CBBA) [1], a decentralized task allocation method. CBBA involves agents independently bidding on tasks based on estimated rewards and resolving conflicts through iterative consensus. Agents in CBBA alternate between bundle construction and conflict resolution phases, progressively allocating themselves paths. However, classical CBBA assumes continuous task execution without interruptions, neglecting

[1]Rui de Gouvea Pinto is with the School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA rjdp3@gatech.edu

[2]Jonathan Rogers is with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA jonathan.rogers@ae.gatech.edu

real-world constraints such as finite battery life and payload limitations.

In this work, we explicitly integrate resource consumption predictions into CBBA's bidding process, allowing agents to strategically schedule replenishment events within their task sequences. This ensures feasibility under realistic resource constraints and improves overall operational efficiency.

III. METHODS

A. Resource-Aware CBBA

We propose Resource-Aware CBBA (RA-CBBA), extending classical CBBA to explicitly handle resource constraints through predictive replenishment planning. Our method integrates resource modeling directly into the bidding process, ensuring task sequences remain feasible under battery and payload limitations.

1) Resource Consumption Model: Agents utilize predictive models estimating battery depletion and payload use based on task execution and travel durations. This enables agents to forecast when and where resource depletion will occur along task sequences, guiding strategic replenishment event insertion.

2) Heuristic-Based Bid Estimation: In classical CBBA, agents compute their bid on a given task based on the marginal increase in reward that adding the task to their current path provides. In RA-CBBA, adding a task may necessitate the insertion of replenishment events, so the bid must account for these added costs. To avoid expensive full-sequence optimizations at each step, agents initially use a heuristic to estimate the bid value for each task.

For each candidate task, the agent simulates inserting it at all possible positions within its current task path. Each candidate insertion results in a different proposed task path, which is simulated to evaluate feasibility and reward.

For each candidate path, replenishment events are inserted immediately before any task that would otherwise violate resource constraints. This conservative placement strategy ensures resource feasibility of the path.

Each resulting task path is evaluated by summing the time-discounted reward of each task, calculated by Equation 1, accounting for travel durations, task execution times, and replenishment delays. The maximum-reward feasible path among the candidate insertions is used as the heuristic bid for that task.

3) Optimal Replenishment Planning: For the task with the highest heuristic bid, the agent refines its estimate by optimizing the placement of replenishment events within the task sequence. This process identifies both the timing of replenishments and the choice of base stations that together maximize the cumulative discounted reward while ensuring the path remains resource-feasible. The resulting bid accurately reflects the true cost of completing the task sequence under resource constraints.

4) Integration with CBBA: Resource-aware bids incorporate replenishment overhead implicitly through adjusted task timings and discounted rewards. Thus, CBBA's consensus-based conflict resolution remains unchanged, preserving the algorithm's decentralized structure and convergence properties.

IV. RESULTS

Resource-Aware CBBA was extensively evaluated in simulation to demonstrate its effectiveness under realistic operational constraints. Simulations modeled agents with a cruise velocity of 15 m/s, agents were given ample payload for all tasks, thus they were constrained exclusively by battery limitations in simulations. Agent batteries are initially randomized between 50–100% of maximum capacity, defined as remaining operational time. During each time step, battery levels decrease accordingly. At the start of each simulation, agents allocate tasks among themselves using RA-CBBA and then simulate task execution paths, with total reward computed as the cumulative discounted rewards.

A naive baseline was implemented for comparison, employing classical CBBA without explicit resource awareness. Naive agents handle resource constraints reactively, during task execution, checking their battery levels after each completed task and visited the nearest base station when insufficient battery remained to complete the next task.

A. Random Trials

Randomized trials involved 20 tasks per scenario, where each task comprised between 15 and 20 nearby points distributed along lines or curves. Tasks were randomly placed in a region approximately 6 km x 6 km. Each task was assigned a reward randomly sampled from the range 5,000-10,000. Two base stations were also randomly placed within this region, each imposing a replenishment delay of five minutes. Agents started at randomized initial locations within the region. Agent fleet size ranged from four to seven, and agent battery capacities ranged from 20 to 60 minutes. Each parameter permutation was tested on 10 randomized scenarios, and the results averaged.

Figure 1 summarizes the performance difference between RA-CBBA and the naive baseline. Under the tightest constraint (20 mins battery life), RA-CBBA outperformed the naive approach. At intermediate battery levels of 30 and 40 minutes the naive approach performs better, likely due to the RA-CBBA heuristic bid computation leading to sub-optimal task selection. However, as the resources become less constrained and battery is abundant, performance of both methods converges to about equal.

Although the performance of RA-CBBA in these randomized trials does not seem to provide a significant benefit over the naive baseline, there are several practical scenarios which could be particularly relevant in construction robotics, where RA-CBBA provides a large advantage over a naive strategy.

B. Scenarios Highlighting RA-CBBA Advantages

To highlight the benefit of RA-CBBA in these practical scenarios, two scenario archetypes were generated. The battery capacity of the agents was varied from 20 to 40 minutes, the replenishment delay incurred from visiting a base station was varied from one minute to five minutes and the number

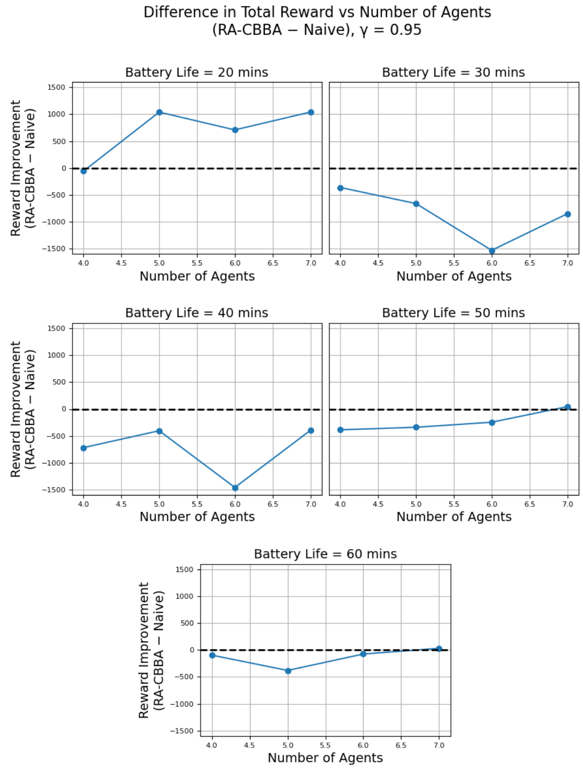


Fig. 1. Average reward difference (Resource-Aware − Naive) across varying battery capacities and fleet size. Positive values indicate RA-CBBA outperforms naive and vice versa.

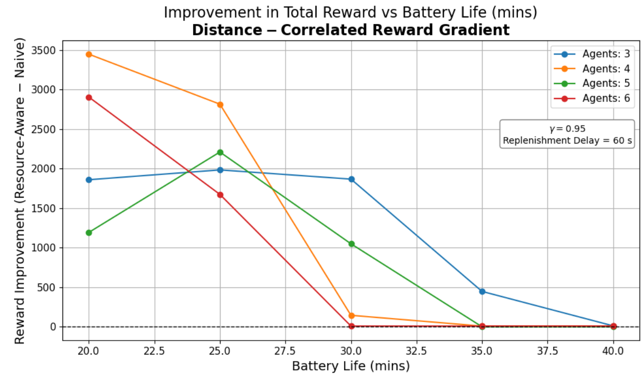


Fig. 2. Average reward improvement (RA-CBBA − Naive) across varying battery capacities for the distance-correlated reward gradient scenario type.

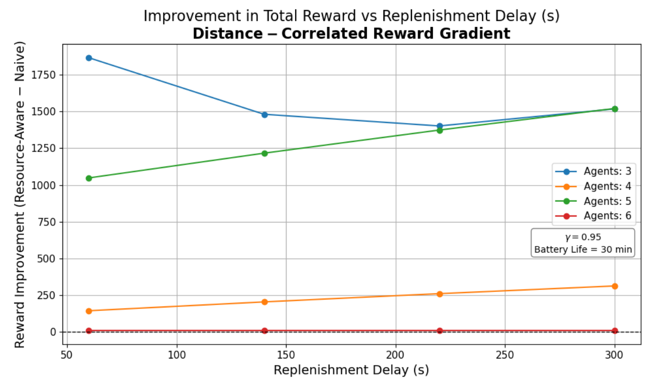


Fig. 3. Average reward improvement (RA-CBBA − Naive) across varying replenishment delays for the distance-correlated reward gradient scenario type.

of agents was varied from three to six. For each scenario type, each permutation of these configurations were tested on 10 different randomized versions of each scenario type.

1) Distance-Correlated Reward Gradient: In the first scenario type, all agents initially start near a single base station located at the origin. The tasks are spatially distributed along a straight line that extends outward from the base. The rewards of tasks are equal to twice their distance in meters from the base station. Specifically, 20 tasks are uniformly sampled along this line at distances between 1 and 5 km from the base station. The primary purpose of this scenario is to illustrate the advantage of proactively planning replenishment events when higher-value tasks are more distant and thus more resource-intensive to complete.

Figures 2 and 3 show the average performance improvement that RA-CBBA had over the naive baseline. As expected, the improvement is greatest when there are tighter resource constraints. Performance advantages of RA-CBBA are significant when battery life is low and decreases as the battery life extends. Similarly, as the replenishment delay increases the RA-CBBA performance advantage increases as well. Smaller agent fleets particularly benefited from integrated replenishment planning, as proactive strategies allowed efficient targeting of distant, high-reward tasks.

The primary advantage RA-CBBA has in these scenarios is the awareness that agents will need to return to the base station to replenish their resources at some point, allowing it to account for this cost in planning. Doing so allows agents to make decisions about whether to go straight for the further, higher reward tasks before they replenish, or even to recharge at the start of the simulation to maximize time at the further tasks, completing more than one at a time before recharging again. The naive agents, on the other hand, cannot account for their resources and thus their plan often involves completing lower-reward tasks along the way to the higher-reward tasks, often requiring them to recharge before reaching the high-reward tasks and thus losing significant portions of reward to time-discounting.

2) Near vs. Fast Base Stations: In the second scenario type, tasks are densely clustered within a radius of 600 m on one side of the operational region, approximately 1 km from the agents' initial location, which is near the midpoint between two base stations separated by approximately 500 meters. The base station closest to the task cluster imposes the nominal replenishment delay (between one and five minutes), whereas the farther base station imposes only half this delay.

Figures 4 and 5 show the average reward improvement that RA-CBBA has over the naive baseline for these scenarios.

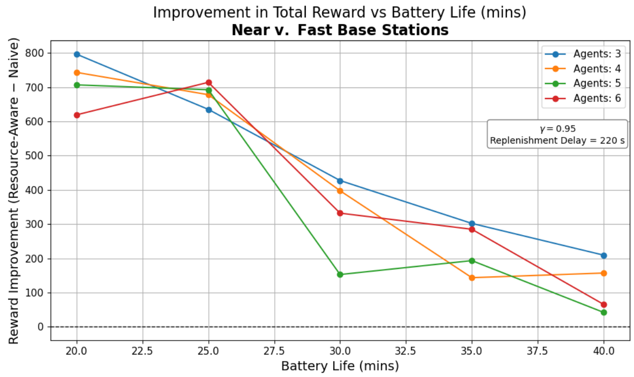


Fig. 4. Average reward improvement (RA-CBBA − Naive) across varying battery capacities for the near vs. fast base station scenario type.

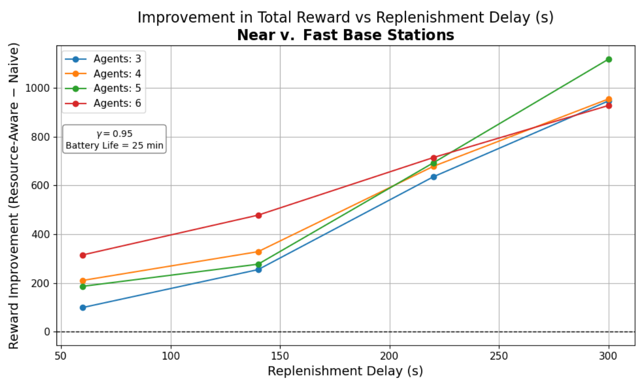


Fig. 5. Average reward improvement (RA-CBBA − Naive) across varying replenishment delays in the near vs. fast base station scenario type.

In low-battery scenarios, where agents must return to base stations often, RA-CBBA significantly outperforms the naive approach. However, this effect becomes less pronounced as agent battery capacity increases. Performance increases as the replenishment delay increases, as the time-savings earned by visiting the farther, faster base station increase.

This scenario is set up to show that RA-CBBA's ability to account for resources in planning gives agents the ability to choose to travel longer distances in order to save time in the long-run or even recharge earlier than needed to ensure the range to reach the farther, faster charging base station. This type of trade-off could occur in practice if an agent needing to charge must choose between the nearest base station to its current location, or the nearest base station to its next task. If the nearest base station to its current location is not on the way to the next task, it makes sense to travel farther to save time in the long-run. This scenario demonstrates RA-CBBA's ability to encode that trade-off during planning. Such trade-offs are representative of real-world settings where replenishment infrastructure may be unevenly distributed, and choosing between proximity and efficiency is essential for sustained operations.

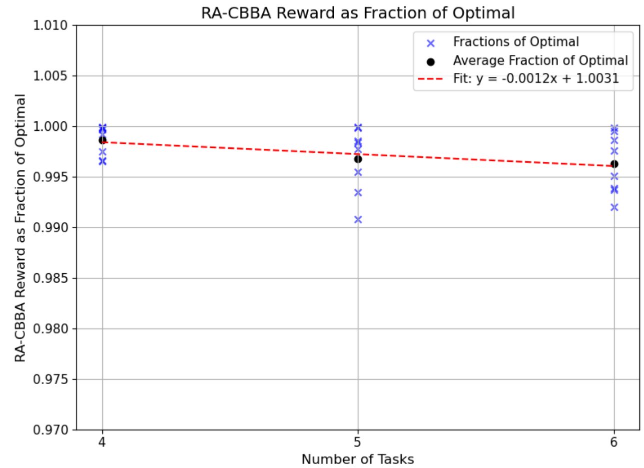


Fig. 6. Fraction of optimal RA-CBBA achieves given a three agent assignment to various number of tasks. On the plot, a value of 1 indicates global optimum performance.

C. Optimality Gap

Finally, we evaluated RA-CBBA's proximity to optimal solutions by conducting exhaustive searches on smaller problem instances (three agents, two base stations, and 4–6 tasks). Figure 6 shows RA-CBBA, on average, achieving solutions greater than 99.5% of optimality across all tested configurations, with the lowest recorded case still above 99%. Extrapolation suggests strong performance scaling potential, though larger instances remain computationally prohibitive for exhaustive verification.

V. CONCLUSIONS

In this work, we have presented Resource-Aware CBBA, an extension to the classical Consensus-Based Bundle Algorithm designed to handle realistic resource constraints encountered by robotic teams. By explicitly integrating predictive resource consumption models and replenishment planning into the bidding process, RA-CBBA ensures that task sequences remain feasible under energy and payload limitations. Our approach maintains the decentralized nature, scalability, and convergence guarantees of traditional CBBA.

Through extensive simulation experiments, we demonstrated that RA-CBBA outperforms naive approaches in scenarios with tight resource constraints and substantial replenishment delays. In particular, in scenarios designed to reflect practical field conditions, such as tasks with distance-correlated rewards and trade-offs between replenishment location proximity and delay, our method yielded considerable improvements in total discounted reward by strategically integrating replenishment events into task planning.

Moreover, we evaluated the optimality of RA-CBBA in small-scale scenarios, demonstrating its capacity to consistently achieve solutions within 99% of optimality. These results highlight RA-CBBA's potential for practical deployment in autonomous field robotics and resource-constrained multi-agent systems.

References

[1] Han-Lim Choi, L. Brunet, and J. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, Aug. 2009.

[2] X. Yu, J. Guo, and H. Zheng, "Extended-cbba-based task allocation algorithm for on-orbit assembly spacecraft," in *2019 IEEE International Conference on Unmanned Systems (ICUS)*, 2019, pp. 883–888.

[3] Y.-z. Zhang, J.-l. Xu, Z.-r. Wu, and Y.-h. Ma, "Complex task assignment of heterogeneous uavs under timing constraints," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, 2020, pp. 853–858.

[4] J. Li and C. Lin, "Coupling task allocation for heterogeneous auvs with communication constraints," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2023, pp. 1196–1201.

[5] J. Xu, J. Wang, and W. Chen, "An efficient recharging task planning method for multi-robot autonomous recharging problem," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019, pp. 1839–1844.

[6] B. Kannan, V. Marmol, J. Bourne, and M. B. Dias, "The autonomous recharging problem: Formulation and a market-based solution," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3503–3510.

Exploring Planning of Redistribution Trajectories for Profile Grading in Amorphous Materials

Louis Hanut[⋆,1,4], Yurui Du[2], Renaud Detry[1,2], Herman Bruyninckx[1,3,4]

***Abstract*— This paper addresses the problem of planning actions to reshape an amorphous material surface to match a desired distribution. This task is complicated by the uncertain and dynamic properties of such materials. Inspired from state-of-the-art planning for the manipulation of amorphous materials, we explore a next-best trajectory planning algorithm for sand redistribution. The method generates multiple candidate sweeps informed by domain-specific priors, including Optimal Transport, a Max-to-Min heuristic, and Adjacent Sweep strategies. Each candidate's outcome is predicted using a simplified sand-tool interaction model, and the trajectory yielding the lowest predicted error is executed. Simulation-based experiments demonstrate that combining multiple candidate generation strategies improves redistribution accuracy and robustness across diverse scenarios.**

I. INTRODUCTION

Automating construction tasks is crucial to meet growing demands while improving the working conditions of construction workers. However, the robotic manipulation of amorphous materials –such as sand, gravel, or plaster –remains challenging due to their uncertain, time-varying, and often heterogeneous properties.

Humans typically address these challenges through iterative and adaptive behavior, adjusting their actions based on observed outcomes. Inspired by this strategy, we study the planning of redistribution actions to shape an amorphous material surface.

Recent works have shown that including feedback, tool-material interaction models, and domain-specific heuristics can significantly enhance performance in amorphous material manipulation. Building on these insights, we propose a next-best-sweep planning framework, inspired by [6], that integrates material modeling and rule-based trajectory generation to produce candidate actions.

Through simulation experiments, we show that combining multiple candidate generation strategies leads to improved surface grading performance.

* Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or HADEA. Neither the European Union nor the granting authority can be held responsible for them.

Funded by the European Union

⋆ Corresponding author, louis.hanut@kuleuven.be

[1] Department of Mechanical Engineering, KU Leuven, Belgium.

[2] Department of Electrical Engineering, KU Leuven, Belgium.

[3] Department of Mechanical Engineering, TU/e Eindhoven, the Netherlands.

[4] Flanders Make @ KU Leuven, Belgium.

II. RELATED WORK

Different methods have been employed in the literature for the manipulation of amorphous materials. First, predictive models were developed to estimate the outcome of actions and build plans accordingly [1], [2].

These models vary in complexity, from simple binary-grid representations [3] to detailed models based on solid/fluid mechanics or discrete element methods. However, for real-time robotic applications, simplified models are generally preferred due to their lower computational demands [1], [4]. Including feedback to update the material state allowed to compensate model simplifications and inaccuracies [5], [6].

Alternatively, model-free approaches were proposed, using domain-specific heuristics to guide planning (e.g., vision-based contour shaping heuristics) [7].

Such rules have also been integrated as priors within model-based frameworks, e.g., for tasks like plastering [5] or cleaning [3]. Notably, [6] used Optimal Transport as a prior to propose candidate actions, which were then evaluated using a model of the material.

Finally, as prediction models depend on several parameters that are difficult to obtain (material, tool, environment conditions ...), learning-based methods have been used to model the material behavior [4], [8], [9]. Learning has also been applied to derive action-selection policies for tasks such as bulldozing [10], [11] and everyday material manipulation [12].

In this work, we present a planning algorithm that combines material state feedback, a simple predictive model and domain knowledge as informative prior to guide action selection.

III. PLANNING MATERIAL REDISTRIBUTION

We address the problem of planning redistribution actions to shape an initial material distribution H_{init} into a desired one H_{des}. Drawing from the literature on amorphous material manipulation, we identify three key elements to enhance planning: (1) intermediate feedback, (2) tool–material interaction modeling, and (3) domain-informed planning priors.

To integrate these components, we propose a "next-best sweep" planning algorithm similar to the one presented in [6]. In between each executed action, the algorithm generates a set of candidate sweep trajectories and evaluates them based on the predicted material redistribution outcome. The candidate generation and selection are essential in the planning and include domain knowledge for efficient redistribution. The planning algorithm is outlined in Algorithm 1, with its main components detailed in the following sections.

A. Simplifying Assumptions and Action Space Restriction

To reduce the complexity of the planning problem, the action space is limited to straight-line sweeps, each defined by the start and end coordinates in the workspace:

$$a = (x_{\text{start}}, y_{\text{start}}, x_{\text{end}}, y_{\text{end}})$$

We use discrete sweeps rather than continuous interactions to simplify execution. Continuous tool motion poses greater challenges for robotic kinematics and results in the constant accumulation and transport of material in front of the tool. In contrast, discrete sweeps allow localized accumulation, enabling material to be deposited temporarily and handled in subsequent actions. Furthermore, the total material quantity is assumed equal to the material quantity of the desired distribution such that no additive or subtractive operations are necessary.

B. Sweep Candidates Generation

We propose three domain-informed methods to generate candidate sweep trajectories:

1) Optimal Transport (OT): Following [6], we leverage Optimal Transport to compute a material redistribution plan. The OT algorithm computes a transport matrix T indicating the optimal flow of material between grid cells. Due to the high computational cost on large grids, we apply OT to a downsampled version of the heightmap (see Fig. 1), yielding a coarse redistribution plan between aggregated zones.
OT assumes direct, non-interfering transport from source to destination cells (as for scoop-and-dump operations). However, our context involves sweeping actions that push material across the surface, which do not strictly adhere to the OT plan. Nevertheless, we assume that the OT plan still provides a strong prior for rearranging the material distribution. Deviations from the OT plan are leveraged by recomputing the transport plan in between each executed sweep, using updated heightmaps from sensor feedback.

2) Max-to-Min Heuristic: This approach defines trajectories from the maximum to minimum of the relative heightmap $H_{\text{rel}} = (H_{\text{mes}} - H_{\text{des}})$. The resulting trajectory connects the maximum material surplus to the maximum deficit points.

3) Adjacent Sweeps: During sweep execution, material accumulates in front of the tool and spills laterally, leaving residual traces beside the tool path. We generate candidate trajectories, a_{adj}, offset by $\pm\delta$ from the previous sweep, where δ is half the tool width. These parallel paths overlap with prior sweeps and allow to remove side-spilled material (see Fig. 1).

Candidate Set Construction: The final candidate set $\mathcal{A} = \{a_{\text{OT}} + \epsilon, a_{\text{max\_min}} + \epsilon, a_{\text{adj}}\}$ comprises:

- a_{OT}: 5 actions generated from the maximum element in the OT plan.
- $a_{\text{max\_min}}$: 5 actions generated from the Max-to-Min heuristic.
- a_{adj}: 2 paths adjacent to the previous sweep.

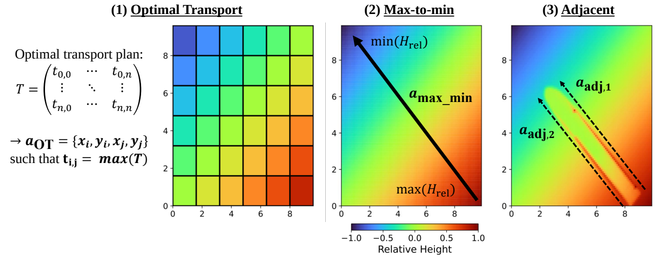


Fig. 1: Illustration of the candidate generation methods. (1) The Optimal transport action uses the maximum element of the transport plan T. (2) The max-to-min sweep is defined between $max(H_{\text{rel}})$ and $min(H_{\text{rel}})$. (3) Adjacent method produces two candidates along the side spills left by the previous sweep.

For a_{OT} and $a_{\text{max\_min}}$, a stochastic perturbation ϵ is introduced for the exploration of nearby trajectories. The candidate generation process is performed in line 1 of Algorithm 1.

C. Sweep prediction - Sand-tool interaction modeling

The next step in Algorithm 1 (line 3) predicts the resulting heightmap for each candidate $a \in \mathcal{A}$, using a model similar to [1] (see Fig. 2). The interaction between the tool and the material is modeled in two consecutive steps, applied pixel-wise along the sweep path:

1) **Step 1 - Tool-induced Motion**: All material volume within the tool's swept area is displaced to neighboring pixels.
2) **Step 2 - Natural Material Flow**: The material returns to its equilibrium state, defined by its angle of repose [13]. Flow only occurs in directions not obstructed by the tool (indicated in purple in Fig. 2).

Although more complex models for various amorphous materials exist in the literature, this simplified model is adequate for our work: the repose angle is its only parameter and could be estimated online through sensor data. While the model introduces approximations, it can still provide a good prior for action selection. Model inaccuracies are mitigated by sensor-based updates between planning iterations.

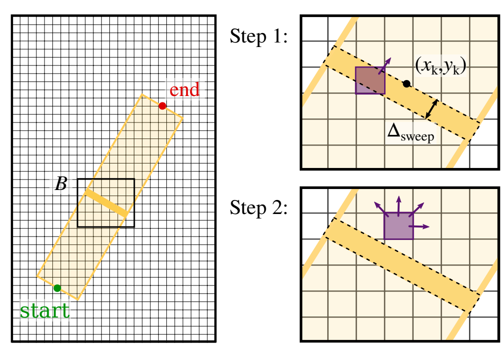


Fig. 2: Sand-tool interaction modeling [1]. For each pixel along the tool trajectory, two prediction steps are computed. Step 1: Tool-induced Motion. Step 2: Natural Material Flow.

D. Sweep Candidates Evaluation

Each predicted heightmap resulting from a candidate action $a \in \mathcal{A}$ is evaluated using a cost function that reflects task performance:

$$C_a = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_{i,\text{des}} - z_{i,\text{mes}})^2} \tag{1}$$

Here, $z_{i,\text{des}}$ and $z_{i,\text{mes}}$ denote the desired and measured heights at pixel i, respectively. This root mean square error (RMSE) quantifies the deviation from the target heightmap.

Candidate evaluation occurs in line 4 of Algorithm 1. The cost C_a of each action is compared to the current best cost C_{best}; if lower, the candidate becomes the new best action a_{best}.

E. Best Sweep Execution

The final step of Algorithm 1 (line 10) executes the selected best candidate a_{best}. During execution, control variables are adapted in real time to address unforeseen deviations and ensure compliance with task constraints, as described in [14]. After the execution, the obtained material surface is evaluated to decide if another iteration is necessary. In that case, Algorithm 1 is repeated.

Algorithm 1 Grading sweep planning algorithm based on a "next-best-sweep" approach [6]. From a set of sweep candidates, the best action is determined using the prediction model in Fig. 2.

```
1:  A ← Generate_sweep_candidates(H_des, H_mes)
2:  for a in A do
3:      H_obtained ← Sweep_prediction(a)
4:      C_a ← Evaluate_prediction(a, H_obtained)
5:      if C_a < C_best then
6:          a_best ← a
7:          C_best ← C_a
8:      end if
9:  end for
10: Execute_sweep(a_best)
```

IV. PRELIMINARY RESULTS AND DISCUSSION

To evaluate the performance of Algorithm 1, we conducted a series of simulation experiments using the model illustrated in Fig. 2. All experiments were performed on 100x100 pixels heightmaps. Fig. 3 contains the task progression for two redistribution scenarios. In Scenario (a), the initial and desired heightmaps correspond to vertical and horizontal linear gradients, respectively. In Scenario (b) both initial and desired distributions are bimodal, featuring two distinct summits.

For each scenario, we compare our method –which uses a candidate set combining all three candidate generation methods –with each method individually. Performance is evaluated based on the RMSE between the final and desired heightmaps after 50 sweep executions, the total distance traveled by the tool, and the total planning time (see Table I). From these results, we observe the following:

- **Improved Redistribution Accuracy**: Our combined approach yields the lowest RMSE in both scenarios, demonstrating the benefit of combining diverse candidate generation strategies.
- **Planning Time Considerations**: As expected, planning time increases with the number of simulated candidates and is higher for our method. Nevertheless, it remains acceptable compared to the real sweep execution time. In practice, planning time can be significantly reduced through parallelization of the prediction step.
- **Role of the Adjacent Method**: On its own, the Adjacent method performs poorly due to its dependence on the initial sweep direction (chosen arbitrarily along the diagonal). However, it complements other strategies well by enabling local refinement of the previous sweep.
- **OT vs. Max-to-Min Trade-offs**: The Max-to-Min method ignores spatial context and may result in inefficient long-range transport. In Scenario (a), this leads to higher tool travel distance compared to Optimal Transport (OT), despite similar RMSE values. Conversely, in Scenario (b), Max-to-Min outperforms OT in redistribution accuracy. This highlights the benefit of keeping both methods to improve robustness against different scenarios.

Overall, these preliminary findings underscore the advantage of combining multiple candidate generation heuristics to enhance robustness and redistribution quality across varying scenarios.

TABLE I: Results of the simulated redistribution after 50 sweeps for two scenarios: (a) Linear distributions and (b) Bimodal distributions (see Fig. 3). Performance is evaluated based on the RMSE between final and desired heightmaps, the total distance traveled by the tool, and the total planning time.

(a) Linear distributions	RMSE	Traveled distance	Planning time
OT	0.2056	368.67	164.3
Max-to-Min	0.2060	446.57	199.37
Adjacent	0.3033	256.4	68.15
Our method	**0.1420**	448.94	383.02

(b) Bimodal distributions	RMSE	Traveled distance	Planning time
OT	0.3154	263.44	129.94
Max-to-Min	0.2455	285.97	141.8
Adjacent	0.3950	206.33	66.07
Our method	**0.2074**	285.73	279.6

V. FUTURE WORK

A first direction for future work is to integrate the proposed planner with the real-time adaptive execution presented in [14]. This would allow adjustment of model parameters (e.g., material's repose angle) based on online measurements. Additionally, during adjacent sweep execution, controlling the tool orientation could help direct material flow more effectively, particularly by pushing material outward from the sweep path.

Next, the experimental scenarios considered in this paper are

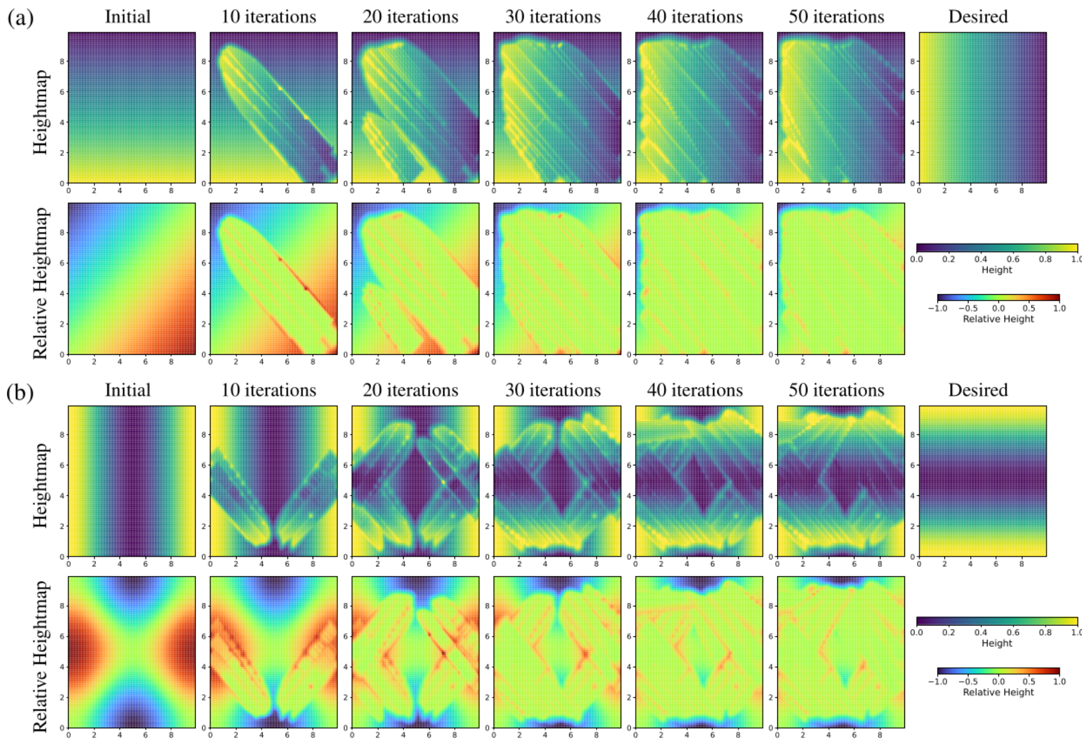


Fig. 3: Results of the simulation experiments, showing task progression when using our planning method (combining all three candidate generation heuristics) in the two redistribution scenarios: (a) Linear distributions and (b) Bimodal distributions.

relatively simple. To be applicable in real-world construction settings, the planner must be extended to handle more complex surface geometries and interactions with rigid obstacles. Finally, the current planning implementation selects the next action by minimizing the immediate cost. Planning over a longer horizon may enhance the overall redistribution performance [5].

REFERENCES

[1] W. Kim, C. Pavlov, and A. M. Johnson, "Developing a Simple Model for Sand-Tool Interaction and Autonomously Shaping Sand," arXiv, Aug. 2019. doi: 10.48550/arXiv.1908.02745.

[2] D. Leidner, G. Bartels, W. Bejjani, A. Albu-Schäffer, and M. Beetz, "Cognition-enabled robotic wiping: Representation, planning, execution, and interpretation," Robotics and Autonomous Systems, vol. 114, pp. 199–216, Apr. 2019, doi: 10.1016/j.robot.2018.11.018.

[3] S. Elliott and M. Cakmak, "Robotic Cleaning Through Dirt Rearrangement Planning with Learned Transition Models," in 2018 IEEE International Conference on Robotics and Automation (ICRA), May 2018, pp. 1623–1630. doi: 10.1109/ICRA.2018.8460915.

[4] K. Tsuruta, S. J. Griffioen, J. M. Ibáñez, and R. L. Johns, "Deep Sandscapes: Design Tool for Robotic Sand-Shaping with GAN-Based Heightmap Predictions," in 2022 Annual Modeling and Simulation Conference (ANNSIM), Jul. 2022, pp. 730–741. doi: 10.23919/ANNSIM55834.2022.9859472.

[5] M. A. Kuhlmann-Jørgensen, J. Pankert, L. L. Pietrasik, and M. Hutter, "PLASTR: Planning for Autonomous Sampling-Based Trowelling," IEEE Robotics and Automation Letters, vol. 8, no. 8, pp. 5069–5076, Aug. 2023, doi: 10.1109/LRA.2023.3291894.

[6] N. Alatur, O. Andersson, R. Siegwart, and L. Ott, 'Material-Agnostic Shaping of Granular Materials with Optimal Transport', in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2023, pp. 10652–10659. doi: 10.1109/IROS55552.2023.10342505.

[7] A. Cherubini, V. Ortenzi, A. Cosgun, R. Lee, and P. Corke, "Model-free vision-based shaping of deformable plastic materials," The International Journal of Robotics Research, vol. 39, no. 14, pp. 1739–1759, 2020, doi: 10.1177/0278364920907684.

[8] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning Robotic Manipulation of Granular Media," PMLR, Oct. 2017, pp. 239–248. Accessed: Mar. 13, 2023. [Online]. Available: https://proceedings.mlr.press/v78/schenck17a.html

[9] Z. Xian et al., "FluidLab: A Differentiable Environment for Benchmarking Complex Fluid Manipulation," in International Conference on Learning Representations, 2023.

[10] C. Ross, Y. Miron, Y. Goldfracht, and D. D. Castro, "AGPNet - Autonomous Grading Policy Network," ISARC Proceedings, pp. 40–43, May 2022.

[11] Y. Miron, Y. Goldfracht, C. Ross, D. D. Castro, and I. Klein, "Autonomous Dozer Sand Grading Under Localization Uncertainties," IEEE Robotics and Automation Letters, vol. 8, no. 1, pp. 65–72, Jan. 2023, doi: 10.1109/LRA.2022.3222990.

[12] Y. Zhang, W. Yu, C. K. Liu, C. Kemp and G. Turk, "Learning to manipulate amorphous materials", ACM Trans. Graph., vol. 39, no. 6, 2020.

[13] S. Nowak, A. Samadani, and A. Kudrolli, 'Maximum angle of stability of a wet granular pile', Nature Phys, vol. 1, no. 1, pp. 50–52, Oct. 2005, doi: 10.1038/nphys106.

[14] L. Hanut, Y. Du, A. Vande Moere, R. Detry, and H. Bruyninckx, "Robotic Framework for Iterative and Adaptive Profile Grading of Sand," in IEEE International Conference on Robotics and Automation, 2025.

Towards Collaborative Manipulation with Car-Like Robot Pushers

Shambhavi Singh[1] and Christoforos Mavrogiannis[1]

***Abstract*— We focus on collaborative manipulation with a team of car-like robot pushers. Pushing can be a practical manipulation strategy for rearranging large, heavy, or unstructured objects without needing grippers with high design complexity and cost. Prior work has focused on simplified problem instances, including prehensile manipulation using grippers, and pushing with holonomic robots. However, real-world applications of object rearrangement in construction, mining, or warehouses motivate the need to support manipulation of diverse objects and supply higher torques. Our key insight is to leverage nonprehensile manipulation to accommodate a wide range of object geometries, and, use car-like robot pushers to apply significantly higher torque than holonomic robots of comparable cost. The non-holonomic constraints imposed by car kinematics in conjunction with pushing-based constraints required for object controllability complicate planning, control, and coordination. To this end, we develop an architecture for planning the motion of multiple car-like robots to produce a desired object rearrangement. Given a goal pose for the object, we first extract a trajectory (sequence of twists) taking the object from its current pose to the goal. For each object twist, we solve an optimization instance to optimally distribute pushing forces and contact configurations among robots. We formulate the optimization as a quadratic programming problem and solve to minimize the magnitude of forces required for each object twist. Each robot executes the sequence of pushing forces that it was assigned in a decentralized fashion using a model predictive controller. Preliminary results validate our approach on four pushing scenarios each involving the rearrangement of a long rectangular object by two car-like robots. Ongoing work involves the evaluation of our architecture on hardware, using a team of 1/10th scale robot racecars.**

***Index Terms*—Pushing, Planar Manipulation, Multi-Robot Systems, Model-Based Optimization**

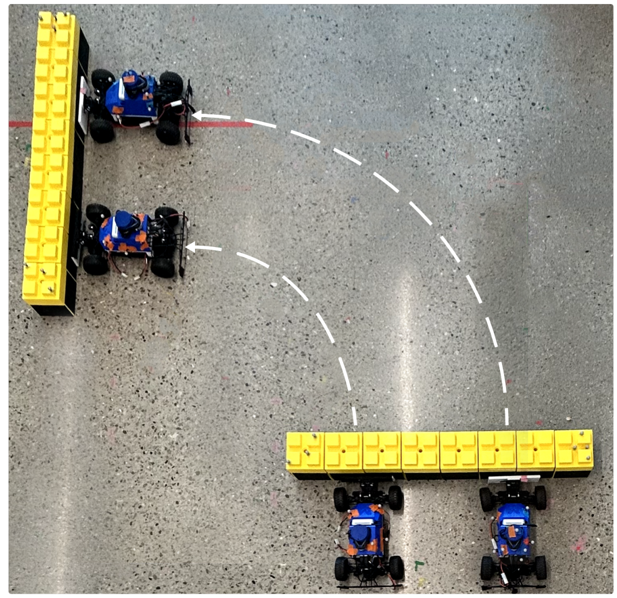


Fig. 1. Example setup of two car-like robots collaboratively pushing a long rectangular object along a curved path.

I. Introduction

Autonomous collaborative manipulation has the potential to transform how robots interact with large or heavy objects in environments like warehouses, construction sites, and factories. While prior approaches rely on holonomic robots or complex grippers, car-like robots offer a compelling alternative due to their higher torque and simpler mechanical design. Nonprehensile manipulation through planar pushing presents a robust strategy for object rearrangement, particularly for objects that are irregularly shaped, heavy, or too large for conventional grippers [1, 3, 6].

Mechanics of planar pushing: Mason (1982) and Peshkin and Sanderson (1988) develop the mechanics of pushing under quasistatic conditions where frictional forces on the object due to the surface μ_s quickly damp out any kinetic energy of the object. Goyal et al. (1991) define the convex boundary of frictional wrenches during contact as the limit surface. With this assumption, Lynch and Mason (1996) develop conditions for a mobile robot pushing an object with line-contact using stable pushes i.e. pushes without relative sliding [14]. Prior work has employed quasistatic stable pushing models to perform planar object rearrangement tasks [1, 19]. In our work, we devise the conditions for planar quasistatic pushing with multiple mobile robots using only stable pushes.

Model-based planning and control: Model-based optimization and data-driven approaches particularly reinforcement learning-based methods have been widely adopted for contact-rich pushing scenarios. Several works formulate the problem of finding pushing contacts and trajectories/forces as constraints of an optimization problem in single [2, 11, 13] or multi-robot scenarios [5, 12, 20] with centralized controllers. In contrast, prior works with decentralized multi-robot control define control laws based on pushing models where robots may take turns to execute pushing actions [16], follow a leader robot [21] or move as a formation in a swarm [4]. Although these works made great leaps in development of push manipulation abilities, their scope was limited to either small holonomic robots or manipulator arms. These works can

[1]Authors are with the Robotics Department, University of Michigan, Ann Arbor, MI, USA. {shasingh, cmavro}@umich.edu

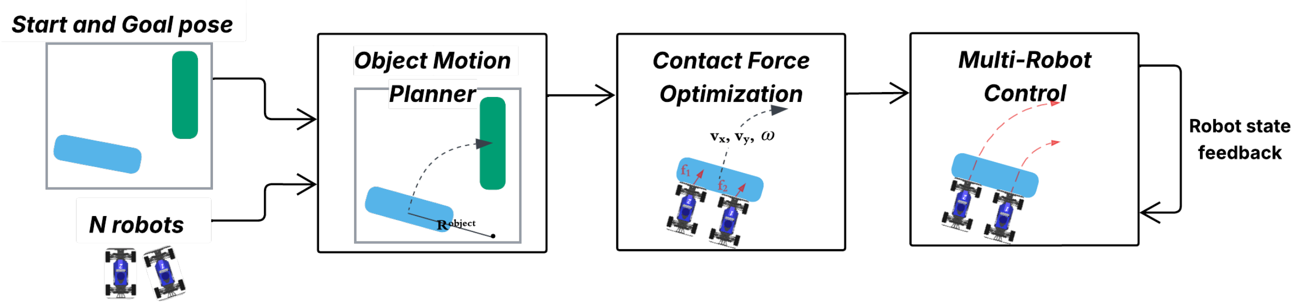


Fig. 2. System Architecture: Given an object start and goal pose, and N robots, we find a valid plan for pushing the object as a sequence of object twists. For each object twist, we distribute pushing forces among the robots. Finally, robots track a desired velocity profile using a closed loop multi-robot controller.

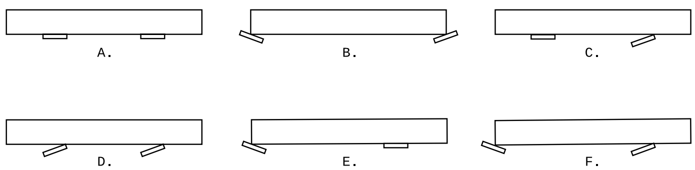


Fig. 3. Various configurations with two pusher robots (0.1m×0.02m) in multiple pushing contact modes with a large rectangular object (0.8m×0.1m).

approximate pushing contact as point-contact due to the small size of the end-effectors/robots compared to the object. In contrast, our work leverages car-like robots to deliver higher torque compared to holonomic robots of comparable cost. This, along with the front bumper of cars forming a line-contact, enables manipulation of objects that are too large, heavy or irregular in shape for point-contact pushing with holonomic robots.

Nonprehensile manipulation of large objects: Recent works highlight growing development of robots for manipulation of large objects. In [8], hierarchical reinforcement learning-based methods are used for obstacle-aware multi-robot manipulation of a large object by quadrupedal robots. Also, in [22] a differential-drive mobile robot navigates through a region cluttered with heavy movable obstacles using a pushing model from a physics engine to efficiently sample rollouts of their controller. Our work is a step towards collaborative non-prehensile manipulation where this reasoning is developed for multiple car-like robots. Our framework decomposes the collaborative pushing problem into three components: object motion planning, optimal force distribution among multiple robots, and low-level robot control. This approach enables manipulation of large objects with kinematically constrained car-like robots.

II. Approach

We formulate the problem for pushing an object in a planar workspace $W \subset SE(2)$ using two non-holonomic car-like robots or *pushers*. The 3D object has a mass M and moment of inertia I, a low center-of-gravity such that $\mathbf{o} = (x, y, \theta) \in W$ represents the object pose, and O represents the boundary of the object. We consider objects that are too heavy for a single robot but sufficiently large to allow multiple robots to push simultaneously. Although there are numerous configurations for pushing the object with car-like pusher robots as illustrated in Fig 3, for the scope of this work we focus on Fig 3-A. In configuration A, robots execute stable pushes maintaining *line-contact* with the object.

We represent the state of the robots as $\mathbf{p_1}, \mathbf{p_2} \in W$, the robot follows rear-axle simple-car kinematics:

$$\dot{\mathbf{p_i}} = [s_i \cos(\theta_i), s_i \sin(\theta_i), s_i \tan(\phi_i)/L] \quad (1)$$

where $\mathbf{u_i}$ is the control input, s_i is the speed, ϕ_i is the steering angle, and L is the wheelbase of the robots with $i \in \{1, 2\}$. Additionally, we assume uniform pressure distribution across the object and uniform frictional properties, where μ_s and μ_c give the friction between the object and the support ground, and the object and the robot respectively.

A. Object Motion Planner

The path of the object consists of a sequence of stable pushes, and the space of stable pushing directions imposes non-holonomic constraints on the motion of the object[14]. We use a hybrid-A^* planner to construct stable pushing paths for the object among obstacles. Our planner generates a Dubins Curve[7] as the shortest path between the start and goal location using L, R, and S primitives corresponding respectively to left, right, and straight motion. The left and right motion primitives are calculated using a minimum turning radius that ensures stable pushing under the quasistatic assumption similar to the planner in [1].

Consider robots $\mathbf{p_1}, \mathbf{p_2} \in W$ pushing an object $\mathbf{o} \in W$ with a constant velocity. Tang et al. (2024) prove that any object transformation with a constant velocity can be represented as an arc transformation. Let the turning radius of the arc be R^{object} as illustrated in Fig. 4, and the contact lines be represented by $\mathbf{r_1}, \mathbf{r_2} \in W$ as the coordinates of their mid-points. Since each car performs stable pushes, the radius of curvature of each car must be larger than R^{car}_{min}. Here, R^{car}_{min} is the minimum radius for sticking contact with stable pushing. Thus, the minimum radius of curvature for the object must be offset such that each robot can perform stable pushes, it is given by:

$$R^{object}_{min} = max(R^{car}_{min} + \mathbf{r_1}.\hat{\imath} - \mathbf{o}.\hat{\imath}, \quad R^{car}_{min} + \mathbf{r_2}.\hat{\imath} - \mathbf{o}.\hat{\imath}) \quad (2)$$

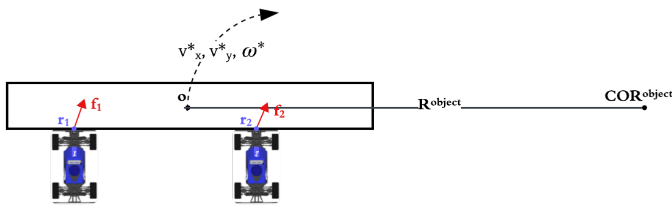


Fig. 4. The object at $o = (x, y, \theta)$ with the desired object twist (v_x^*, v_y^*, ω^*) traces an arc of radius R^{object} while robots apply forces f_1 and f_2 through contacts at r_1 and r_2 respectively.

where $\hat{\imath}$ is the unit vector along the X-axis. For the scope of this work, we use the farthest contact point $max(\mathbf{r_i}.\hat{\imath} - \mathbf{o}.\hat{\imath})$ to find the R^{object}_{min} that can allow robots anywhere along the edge of the object to complete that object transformation.

B. Force Distribution Optimization

For each desired object twist $\dot{\mathbf{o}}^* = (v_x, v_y, \omega)$ in the planned trajectory, we determine how to optimally distribute pushing forces among the robots. We formulate this as a Mixed-Integer Quadratically Constrained Programming (MIQCP) problem. Robots are allowed to push with their flat bumpers anywhere on the perimeter of the object, given that the line-contact does not intersect with a corner point of the object (to maintain configuration A from Fig. 3), or collide with another robot.

1) Friction Constraints: For two robots with sticking line-contacts, the configuration is defined as: $\xi \triangleq \mathbf{r}_1\mathbf{r}_2$ Further, for the i-th robot where $i \in \{1, 2\}$, the force $\mathbf{f}_i$ applied by the robot, can be decomposed into normal and tangential components: $\mathbf{f}_i \triangleq \mathbf{f}_i^{\mathrm{n}} + \mathbf{f}_i^{\mathrm{t}} \triangleq f_i^{\mathrm{n}}\mathbf{n}_i + f_i^{\mathrm{t}}\boldsymbol{t}_i$ where $\mathbf{n_i}$ and $\mathbf{t_i}$ represent the unit vectors and f_i^{n} and f_i^{t} represent the frictional force magnitudes in the normal and tangential directions respectively. With the following constraints due to Coulomb's law of friction for sticking contact:

$$0 \le f_i^{\mathrm{n}} \le f_{i,\max}; \quad 0 \le |f_i^{\mathrm{t}}| \le \mu_s f_i^{\mathrm{n}} \tag{3}$$

2) Generalized force for two robots: The total force applied by the two robots can be represented by $\mathbf{F}_\xi$ as:

$$\mathbf{F}_\xi \triangleq (\mathbf{F}_\xi^{\mathrm{n}}, \mathbf{F}_\xi^{\mathrm{t}}) \triangleq (f_1^{\mathrm{n}}, f_2^{\mathrm{n}}, f_1^{\mathrm{t}}, f_2^{\mathrm{t}}) \in \mathbb{R}^4$$

3) Object Dynamics: The combined generalized force applied on the object is:

$$\mathbf{q}_\xi \triangleq (f_x, f_y, m)$$

where:

$$\begin{aligned}(f_x, f_y) &= \mathbf{f}_1 + \mathbf{f}_2\\ m &= (\mathbf{r}_1 - \mathbf{o}) \times \mathbf{f}_1 + (\mathbf{r}_2 - \mathbf{o}) \times \mathbf{f}_2\end{aligned}$$

In matrix form $\mathbf{q}_\xi = \mathbf{J}\mathbf{F}_\xi$ where $\mathbf{J} = \nabla_{\mathbf{F}_\xi}\mathbf{q}_\xi$ is the Jacobian.

4) Limit Surface Model: Under the quasistatic assumption, the total generalized force $\mathbf{q}\xi$ is constrained on a limit surface approximated by an ellipsoid:

$$(f_x/f_{\max})^2 + (f_y/f_{\max})^2 + (m/m_{\max})^2 = 1 \tag{4}$$

Following the approach from the limit surface theory, the gradient of the limit surface is proportional to the desired object velocity:

$$\nabla\mathcal{L}(\mathbf{q}\xi) = \lambda\dot{\mathbf{o}}^* \tag{5}$$

where $\lambda > 0$ is a scaling factor and:

$$\nabla\mathcal{L}(\mathbf{q}\xi) = \left(\frac{2f_x}{f_{\max}^2}, \frac{2f_y}{f_{\max}^2}, \frac{2m}{m_{\max}^2}\right)$$

We also constrain the motion of the robots to their respective contact points, ensuring the motion of the contact point aligns with the kinematically constrained motion of the robot.

$$\begin{bmatrix} v_x\\ v_y\\ \omega\end{bmatrix} + \begin{bmatrix} -\omega(r_{i,y} - o_y)\\ \omega(r_{i,x} - o_x)\\ 0\end{bmatrix} = \begin{bmatrix}\dot{p}_{i,x}\\ \dot{p}_{i,y}\\ \dot{p}_{i,\theta}\end{bmatrix}, \quad \forall i \in \{1, 2\} \tag{6}$$

5) Stable Pushing Conditions: A stable pushing trajectory exists if the required generalized force $\mathbf{q}\xi$ lies within the achievable force set Q_ξ given the constraints on the robots' individual forces. In the objective function, we minimize a weighted sum of the L_1-norm along with the $L\infty$-norm of the individual force magnitudes $\mathbf{F}_\xi$. We thus minimize the total magnitude of forces applied on the object, and ensure that the pushing forces are distributed between the robots.

C. Multi-Robot Controller

Given pushing forces and pushing poses of robots for a desired object twist, we extract a velocity-profile for the motion of each robot. Under quasistatic conditions, the robots exert these pushing forces by tracking desired velocities for the extracted trajectory. We use a model predictive controller to ensure that robots maintain their desired velocities, thereby performing the object twist. For the scope of this work, the controller accounts for kinematic constraints and the robots maintain their velocities relative to each other, the controller does not account for motion of the object in the controller within its rollouts.

We assume the robots start in stable-pushing contact with the object at the start pose. When the robots successfully complete pushing the object along a desired twist, they reposition to pushing positions for the next object twist. We use car-like conflict based search (CL-CBS) [23] to plan the repositioning paths of the two robots. Our controller thus, switches between two modes of contact: stable pushing and repositioning. Each robot uses a Model Predictive Path Integral (MPPI) [24] controller that optimizes control inputs over a receding horizon to maintain its pushing configuration relative to the other robot. In future work, we aim to use an approximate pushing model to increase robustness and allow recovery from failures while pushing. We implement this path tracking model-predictive controller for each robot using the `pytorch_mppi` library.

III. Results

We demonstrate our framework on MuSHR [18], an open-source 1/10th-scale mobile robot racecar, augmented with a 3D-printed flat bumper for pushing in a Mujoco simulation environment shown in Fig. 5. Our test cases include the four

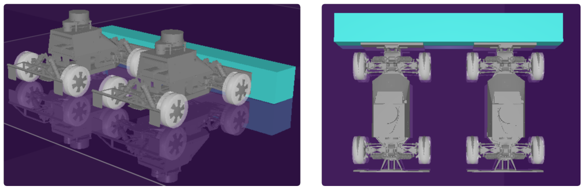


Fig. 5. Two MuSHR robot cars pushing a large object in a Mujoco simulation environment.

TABLE I
COMPARISON OF MEAN ERROR, STANDARD ERROR, AND PATH LENGTH ACROSS DIFFERENT TEST CASES OVER 100 TRIALS.

Test Case	Mean Error (m)	Std Error (m)	Path Length (m)
1	0.004	0.0016	4
2	-0.057	0.009	3.92
3	0.129	0.0045	9.84
4	-0.086	0.0039	7.84

pushing trajectories in Fig. 6 where robots track a desired object trajectory for an object of size 0.1m$\times$0.8m$\times$0.1m, we present the average metrics over 100 trials in Table I. The metrics show that error accumulation correlates with path length and complexity. The longer paths in cases 3 and 4 exhibit cumulative errors as robots execute turns while maintaining pushing contact. Case 3 has the largest path length, resulting in the largest error. Additionally, Case 2 has a large mean error given its small path length because of sliding motion observed while rotating the object. The object slides outwards during the transformation, resulting in the negative bias of the mean. Despite this, the low standard error across all test cases demonstrates the strengths of leveraging stable pushing even for extended trajectories.

IV. DISCUSSION

In the future, we aim to explore additional pushing configurations, including the ones illustrated in Fig. 3. These pushing configurations use a diverse range of contact-rich interactions with corner contacts, opposing forces and caging strategies [9] which, in turn, should help in finding more cost-effective object maneuvers. We also aim to demonstrate these pushing configurations with more than two robots.

Additionally, we plan to develop an approximate analytical pushing model to reduce error accumulation along longer pushing paths. In this work, we observe higher errors along the longer trajectories or larger number of segments. However, when using an approximate model to predict the state of the pushed object in the controller, we can enable robots to take corrective measures while pushing and improve pushing accuracy. Ongoing work involves the development of analytical and learning-based models for pushing with car-like robots.

Lastly, we plan to demonstrate our methods on hardware using 1/10th scale MuSHR robot racecars to identify and address practical challenges not captured in simulation.

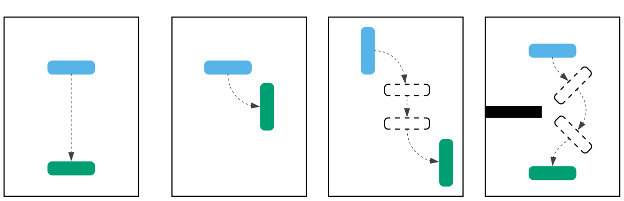


Fig. 6. Start (blue) and goal (green) positions of four test cases with intermediate transitions illustrated.

REFERENCES

[1] J. Ahn and C. Mavrogiannis. ReloPush: Multi-object rearrangement in confined spaces with a nonholonomic mobile robot pusher. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[2] M. Bauza, F. R. Hogan, and A. Rodriguez. A Data-Efficient Approach to Precise and Controlled Pushing. In *Proceedings of The 2nd Conference on Robot Learning*. PMLR, 2018.

[3] O. Ben-Shahar and E. Rivlin. Practical pushing planning for rearrangement tasks. *IEEE Transactions on Robotics and Automation*, 1998.

[4] F. Bertoncelli and L. Sabattini. Planar pushing manipulation with a group of mobile robots. In *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021.

[5] F. Bertoncelli, M. Selvaggio, F. Ruggiero, and L. Sabattini. Task-oriented contact optimization for pushing manipulation with mobile robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[6] M. Dogar and S. Srinivasa. A framework for push-grasping in clutter. *Robotics: Science and systems VII*, 1:65–72, 2011.

[7] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 1957.

[8] Y. Feng, C. Hong, Y. Niu, S. Liu, Y. Yang, W. Yu, T. Zhang, J. Tan, and D. Zhao. Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[9] J. Fink, M. A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[10] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction part 1. limit surface and moment function. *Wear*, 143:307–330, 1991.

[11] F. R. Hogan and A. Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. In *Algorithmic Foundations of*

Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, 2020.

[12] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer. Cooperative multi-robot object transportation system based on hierarchical quadratic programming. *IEEE Robotics and Automation Letters*, 2021.

[13] K.-C. Kuo and K.-Y. Chan. Stable pushing in narrow passage environment using a modified hybrid A* algorithm. *J Intell Manuf*, 2024.

[14] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *The international journal of robotics research*, 1996.

[15] M. Mason. Manipulator grasping and pushing operations. *Ph.D. Thesis*, 1982.

[16] M. J. Mataric, M. Nilsson, and K. T. Simsarin. Cooperative multi-robot box-pushing. In *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots*, 1995.

[17] M. Peshkin and A. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Journal on Robotics and Automation*, 4(6):569–598, 1988.

[18] S. S. Srinivasa, P. Lancaster, J. Michalove, M. Schmittle, C. Summers, M. Rockett, J. R. Smith, S. Choudhury, C. I. Mavrogiannis, and F. Sadeghi. Mushr: A low-cost, open-source robotic racecar for education and research. *CoRR*, abs/1908.08031, 2019.

[19] S. Talia, A. Thareja, C. Mavrogiannis, M. Schmittle, and S. S. Srinivasa. PuSHR: A Multirobot System for Nonprehensile Rearrangement. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[20] Z. Tang, Y. Feng, and M. Guo. Collaborative planar pushing of polytopic objects with multiple robots in complex scenes. In *Robotics: Science and Systems (RSS)*, 2024.

[21] Z. Wang and M. Schwager. Multi-robot manipulation without communication. In *Distributed Autonomous Robotic Systems*, 2016.

[22] J. J. Weeda, S. Bakker, G. Chen, and J. Alonso-Mora. Pushing through clutter with movability awareness of blocking obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[23] L. Wen, Y. Liu, and H. Li. Cl-mapf: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints. *Robotics and Autonomous Systems*, 2022.

[24] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Assembling Growing Structures with BuilderBots and Stigmergic Blocks

Yating Zheng,[1,2] Michael Allwright,[3] Weixu Zhu,[3] Majd Kassawat,[4] Heiko Hamann[5] and Marco Dorigo[3]

***Abstract*— In this paper, we discuss an algorithm to implement growing structures in a swarm construction system consisting of intelligent building materials, called *stigmergic blocks*, and of autonomous robots, called *BuilderBots*, capable of assembling the stigmergic blocks. The key idea is to distribute the intelligence necessary to coordinate the construction process between the stigmergic blocks—they can compute and communicate with other stigmergic blocks as well as with the robots—and the BuilderBots, whose behavioral rules can remain simple thanks to their interaction with the stigmergic blocks. We study two construction tasks: the construction of a 2D structure growing in one direction and the construction of a 3D hanging structure expanding in four directions. We show the feasibility of our approach with demonstrations of two BuilderBots and up to nine stigmergic blocks. The demonstration results show that our block-coordinated algorithm is able (i) to coordinate multiple robots to assemble growing structures; (ii) to guide the robots to find the available building sites; and (iii) to precisely control the construction progress of symmetrically growing structures.**

I. INTRODUCTION

State-of-the-art swarm construction systems typically consist of robots that assemble structures using passive building materials, such as amorphous materials [1], fibers [2], and modular blocks [3]–[5]. These robots operate based on behavioral rules, and a key challenge lies in ensuring that these rules are non-conflicting to prevent issues such as self-locking or overlapping during construction. This approach does not scale well: the complexity of the robots' behavioral rules severely increases when the building complexity increases.

To address this scalability issue, in our approach, we implement a stigmergy-based system, where stigmergic signals are deposited in the intelligent building material, that is, in the stigmergic blocks [6]. To be best of our knowledge, only few works have explored the idea of embedding intelligence into building materials for swarm robotics construction systems [7]–[9]. A part for these works, the scalability issue has also been considered in another track of research, where self-reconfigurable blocks are designed to be capable of traversing 3D structures using a variety of motion primitives, including jumping and controlled rolling [10]–[12]. In addition to research focused on designing and implementing swarm construction in real-world systems, there have also been theoretical studies exploring various collective construction algorithms in 2D or 3D environments [13]–[17].

In this paper, we develop a block-coordinated algorithm[1] for a physical swarm construction system composed of *BuilderBots* and *stigmergic blocks* whose task is to assemble growing structures in both 2D and 3D environments (see Fig. 1). In this swarm construction system, the construction control logic is embedded in the stigmergic blocks, whereas the BuilderBots follow relatively simple rules to search for free blocks, pick them up, and then deposit them at locations as indicated by the stigmergic blocks in the already partially built structure [21]. In contrast to stigmergy-based approaches, earlier non-stigmergy-based algorithms typically relied on the robot's onboard intelligence to manage the construction of target structures composed of passive, inert blocks. These earlier systems required each robot to posses a global understanding of the design and to coordinate its actions accordingly. Stigmergy-based systems, inspired by social insects such as termites, instead allow robots to interact indirectly via environmental cues. This enables complex structures to emerge from simple local interactions, without

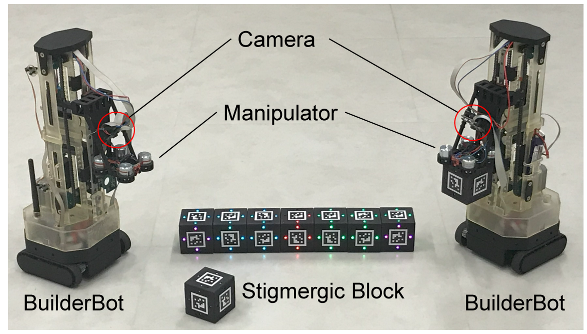


Fig. 1: Swarm robotics construction system composed of stigmergic blocks and robots. The stigmergic blocks can communicate with their neighbors through NFC tags installed on all the blocks' faces. The robots are called BuilderBot and cannot directly communicate with each other. They use their camera to locate blocks by detecting the tags on the block faces and to recognize the signalled LED colors. Using its manipulator, a BuilderBot can pick up a block and place it at a desired spot. More information about the hardware can be found in [18].

[1] Institute for Theoretical Biology, Department of Biology, Humboldt-Universität zu Berlin, 10115 Berlin, Germany yating.zheng@hu-berlin.de

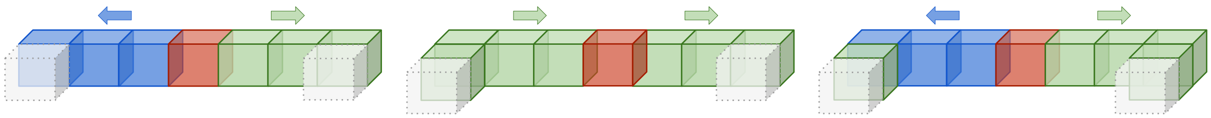
[2] Excellence Cluster 'Science of Intelligence', Technische Universität Berlin, 10587 Berlin, Germany

[3] IRIDIA, Université Libre de Bruxelles, Brussels, Belgium

[4] Universidad Jaume I, Castellon, Spain

[5] Department of Computer and Information Science, University of Konstanz, Konstanz, Germany

[1]We have previously studied our block-coordinated algorithm in simulation [19] using the ARGoS simulator [20].



(a) Step 1: guiding robots to left and right ends (b) Step 2: guiding robots to the right side (c) Step 3: guiding robots to left and right ends

Fig. 2: Steps of assembling symmetric growing structures using our block-coordinated algorithm. The root block is initialized in red to distinguish it from the other blocks. A block displaying blue indicates a left turn, while green indicates a right turn when the BuilderBot approaches the structure from the front. A transparent block with a dotted outline indicates the expected building site.

the need for centralized control.

II. Hardware Demonstrations

To demonstrate our swarm construction system using hardware (as opposed to simulated) robots and stigmergic blocks, we use it to build two target structures. In these demonstrations, we show that the block-coordinated algorithm effectively and accurately controls the building progress during the full construction process. A supplementary video showing the hardware results is provided [22].

A. Construction of a 2D structure

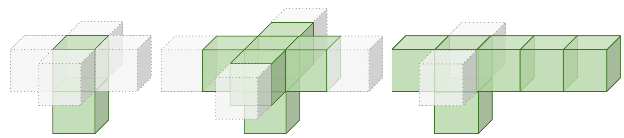
In swarm construction systems, robots often spend significant time searching for construction sites and for building material, while avoiding collisions with each other. In our system, the intelligence has been relocated from the robots to the blocks: the blocks use the colors displayed on their faces to guide the robots during construction, increasing overall building efficiency.

We show a target structure and a representative example of the construction process in Fig. 2. In this example, the task is to let the structure grow in a balanced way; that is, building material should be added to the left and right ends of the structure in an orchestrated manner, coordinated by the stigmergic blocks in the structure itself.

In our experiment, two robots coordinate using information provided by stigmergic blocks embedded in the structure [23]. We position two BuilderBots, each holding a block in its manipulator and facing the structure, and allow them to begin searching for a suitable building site. This setup eliminates the need for the BuilderBots to wander randomly in search of unused blocks. Each episode ends once both robots have placed their respective blocks. All stigmergic blocks remain active and coordinate autonomously throughout the process.

B. Construction of a 3D structure

In this experiment, we explore how dynamic construction paths can prevent a suspended structure from collapsing during the building process. We consider a 3D structure that requires balanced construction due to physical constraints. When building an overhanging structure, gravitational forces impose constraints that must be considered to prevent collapse. Here, we consider a simple overhanging cross structure: a supporting block carrying a cross made of five blocks,



(a) (b) (c)

Fig. 3: Construction of an overhanging cross. The initial (a) and final (b) states in the block-coordinated algorithm. The initial (a) and one possible final (c) state of using earlier non-stigmergy-based algorithms: the structure shown in (c) would collapse due to the unbalanced gravitational forces on the left and right arms.

as shown in Fig. 3. Earlier non-stigmergy-based algorithms typically rely solely on each robot's local perception of the structure composed of passive blocks [19], which can lead to unstable construction outcomes, as illustrated in Fig. 3(c). In contrast, our block-coordinated algorithm successfully enables the construction of this symmetric, overhanging 3D structure. We successfully constructed the 3D target structure using our swarm construction system with two BuilderBots [24], coordinated through stigmergic blocks. Once the initial four adjacent blocks are in place (green blocks in Fig. 3b, which are originally shown as transparent blocks in Fig. 3a), the root block broadcasts update messages to all blocks in the structure to initiate the next construction round—adding a second set of four adjacent blocks. Upon receiving these messages, the initial four blocks update their designated faces to purple, indicating the locations where the next blocks should be attached.

III. Conclusions

Our results demonstrate that embedding the coordination algorithm within the stigmergic blocks, together with simple behavior rules for the BuilderBots, can significantly simplify swarm construction tasks. In future work, we aim to enhance both the hardware and software components of our construction system to support the building of more complex structures.

IV. Acknowledgement

Yating acknowledges the funding by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC

2002/1 'Science of Intelligence', Project number 390523135. Weixu and Yating thank the China Scholarship Council (No. 201706270186, No. 201806040106) respectively. Marco Dorigo acknowledges support from the Belgian F.R.S.-FNRS, where he holds the position of Research Director.

REFERENCES

[1] N. Napp, O. R. Rappoli, J. M. Wu, and R. Nagpal, "Materials and mechanisms for amorphous robotic construction," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 4879–4885.

[2] C. Eschke, M. K. Heinrich, M. Wahby, and H. Hamann, "Self-organized adaptive paths in multi-robot manufacturing: reconfigurable and pattern-independent fibre deployment," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4086–4091.

[3] K. H. Petersen, R. Nagpal, and J. K. Werfel, "Termes: An autonomous robotic system for three-dimensional collective construction," *Robotics: science and systems VII*, 2011.

[4] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, 2014.

[5] M. Allwright, N. Bhalla, and M. Dorigo, "Structure and markings as stimuli for autonomous construction," in *Proceedings of the Eighteenth International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 296–302.

[6] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 20–28, 2006.

[7] J. Werfel, Y. Bar-Yam, D. Rus, and R. Nagpal, "Distributed construction by mobile robots with enhanced building blocks ," in *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2006, pp. 2787–2794.

[8] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2485–2492.

[9] S. Leder, H. Kim, O. S. Oguz, N. Kubail Kalousdian, V. N. Hartmann, A. Menges, M. Toussaint, and M. Sitti, "Leveraging building material as part of the in-plane robotic kinematic system for collective construction," *Advanced Science*, vol. 9, no. 24, p. 2201524, 2022.

[10] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, "3d m-blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1925–1932.

[11] J. W. Romanishin, J. Mamish, and D. Rus, "Decentralized control for 3d m-blocks for path following, line formation, and light gradient aggregation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4862–4868.

[12] J. Romanishin, J. M. Bern, and D. Rus, "Self-reconfiguring robotic gantries powered by modular magnetic lead screws," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4225–4231.

[13] J. Werfel and R. Nagpal, "Three-dimensional construction with mobile robots and modular blocks," *The International Journal of Robotics Research*, vol. 27, no. 3-4, pp. 463–479, 2008.

[14] A. Grushin and J. A. Reggia, "Automated design of distributed control rules for the self-assembly of prespecified artificial structures," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 334–359, 2008.

[15] N. Napp and E. Klavins, "Robust by composition: Programs for multi-robot systems," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 2459–2466.

[16] S.-k. Yun, M. Schwager, and D. Rus, "Coordinating construction of truss structures using distributed equal-mass partitioning," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 607–623.

[17] K. Sugawara and Y. Doi, "Collective construction of dynamic structure initiated by semi-active blocks," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 428–433.

[18] M. Allwright, W. Zhu, and M. Dorigo, "An open-source multi-robot construction system," *HardwareX*, vol. 5, p. e00050, 2019.

[19] Y. Zheng, M. Allwright, W. Zhu, M. Kassawat, Z. Han, and M. Dorigo, "Swarm construction coordinated through the building material," in *Artificial Intelligence and Machine Learning: 32nd Benelux Conference, BNAIC/Benelearn 2020, Leiden, The Netherlands, November 19–20, 2020, Revised Selected Papers*. Springer, 2021, pp. 188–202.

[20] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[21] Y. Zheng, M. Allwright, W. Zhu, M. Kassawat, Z. Han, and M. Dorigo, "Supplementary figure for the control logic for robot behaviors," https://osf.io/egdvj, 2025.

[22] Y. Zheng, M. Allwright, W. Zhu, M.Kassawat, H. Hamann, and M. Dorigo, "Supplementary video of hardware experiments," https://osf.io/tsc9m, 2025.

[23] Y. Zheng, M. Allwright, W. Zhu, M. Kassawat, Z. Han, and M. Dorigo, "Supplementary video of guided construction with two robots," https://osf.io/j3tg4, 2025.

[24] Y. Zheng, M. Allwright, W. Zhu, M.Kassawat, Z. Han, and M. Dorigo, "Supplementary video of cross construction with two robots," https://osf.io/vgzn2, 2025.

Hybrid Voting-Based Task Assignment in Modular Construction Scenarios

Daniel Weiner
Computer Science Department
Graduate Center, City University of New York
New York, United States
dweiner@gradcenter.cuny.edu

Raj Korpan
Computer Science Department
Hunter College and Graduate Center, City University of New York
New York, United States
raj.korpan@hunter.cuny.edu

***Abstract*—Modular construction, involving off-site prefabrication and on-site assembly, offers significant advantages but presents complex coordination challenges for robotic automation. Effective task allocation is critical for leveraging multi-agent systems (MAS) in these structured environments. This paper introduces the Hybrid Voting-Based Task Assignment (HVBTA) framework, a novel approach to optimizing collaboration between heterogeneous multi-agent construction teams. Inspired by human reasoning in task delegation, HVBTA uniquely integrates multiple voting mechanisms with the capabilities of a Large Language Model (LLM) for nuanced suitability assessment between agent capabilities and task requirements. The framework operates by assigning Capability Profiles to agents and detailed requirement lists called Task Descriptions to construction tasks, subsequently generating a quantitative Suitability Matrix. Six distinct voting methods, augmented by a pre-trained LLM, analyze this matrix to robustly identify the optimal agent for each task. Conflict-Based Search (CBS) is integrated for decentralized, collision-free path planning, ensuring efficient and safe spatio-temporal coordination of the robotic team during assembly operations. HVBTA enables efficient, conflict-free assignment and coordination, facilitating potentially faster and more accurate modular assembly. Current work is evaluating HVBTA's performance across various simulated construction scenarios involving diverse robotic platforms and task complexities. While designed as a generalizable framework for any domain with clearly definable tasks and capabilities, HVBTA will be particularly effective for addressing the demanding coordination requirements of multi-agent collaborative robotics in modular construction due to the predetermined construction planning involved.**

***Index Terms*—Multi-Agent Systems, Task Assignment, Voting, Large Language Models, Modular Construction**

I. Introduction

The construction industry is increasingly exploring the potential of robotic automation to improve efficiency, safety, and scalability [1]. Modular construction, characterized by the off-site fabrication of components and their subsequent on-site assembly, is particularly well-suited to automated implementation [2]. Modular construction provides a scalable, robot-friendly method for building. Multi-agent systems (MAS) offer significant opportunities for construction tasks ranging from site preparation, such as push manipulation of loose materials for mapping, leveling, and shaping, to the intricate processes of prefabricated assembly [3].

However, effectively deploying and coordinating these MAS, especially when they are heterogeneous (composed of agents with different capabilities), presents substantial challenges. Coordinating a team of heterogeneous agents efficiently requires sophisticated planning and scheduling algorithms. Key tasks include organizing and ordering agent actions and resources, and ensuring seamless agent coordination. Commonly used planning and scheduling techniques include sequencing, bioinspired methods, and optimization. Task assignment, determining which agent should perform which task, is a critical precursor to successful coordination and execution [4]. In dynamic construction environments, agents must not only be assigned tasks appropriately based on their capabilities but also navigate the shared workspace efficiently and without conflict. Traditional task allocation methods may struggle to capture the nuances of agent capabilities and task requirements, deal with unexpected task requirements, or adapt to changing conditions on a construction site [5].

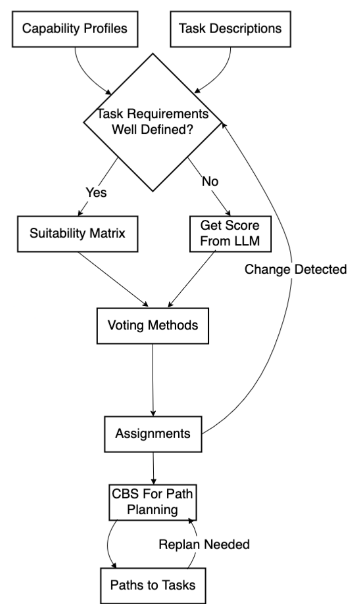


Fig. 1. Diagram of the HVBTA system. The Suitability Matrix is created from the Task Descriptions and Capability Profiles when all task requirements are well defined, otherwise, LLM integration is utilized to score suitability, then all scores are passed to voting methods for task allocation. Finally CBS plans paths to the tasks for each agent.

This paper introduces the Hybrid Voting-Based Task Assignment (HVBTA) framework, as shown in Figure 1, as a solution to the complex task allocation and coordination challenges in multi-agent modular construction. HVBTA is a generalizable framework that draws inspiration from human reasoning in task delegation. It combines voting with Large Language Model (LLM)-based reasoning to determine which robots are best suited for which tasks. HBVTA is designed to handle well-defined multi-agent settings, making it particularly effective for the demanding coordination requirements of collaborative modular construction due to the predetermined construction planning involved. The next section describes related work, and subsequent sections describe the HVBTA framework in detail and consider how HVBTA may be applied to MAS in the construction sector.

II. Related Work

Coordinating heterogeneous multi-agent teams for structured tasks, such as modular construction, has inspired a range of allocation and planning techniques. Prior work can be grouped into four main paradigms: optimization-based, market/auction methods, LLM augmentation, and decentralized path planning. In the following, we briefly survey each and highlight how HVBTA builds on and differs from these foundations.

A. Optimization-Based Approaches

Optimization-based approaches model task allocation as a constrained optimization problem, often using mixed-integer programming (MIP), constraint satisfaction, or heuristic-based models to maximize efficiency metrics such as task coverage, makespan, or resource utilization [6]. **Exact methods** (e.g., greedy algorithms, local search, Simulated Annealing) trade solution quality for speed but provide no optimality bounds [6]. **Metaheuristics** (e.g., Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization) explore large search spaces via population-based or bio-inspired strategies, yielding good solutions on mid-sized problems yet often requiring significant parameter tuning [6]. **Hybrid schemes** combine clustering or graph partitioning with exact or metaheuristic solvers to decompose large instances into manageable subproblems [6].

While these approaches excel when full cost functions and agent–task compatibilities are known a priori, they often falter under partial, noisy, or semantically rich requirements. In contrast, HVBTA's voting framework naturally accommodates incomplete suitability data and uses LLM tie-breaking to handle unseen or context-dependent demands.

B. Market and Auction-Based Methods

Market-based and auction algorithms have long dominated multi-agent task allocation, balancing agent bids against task priorities [7], [8]. While these methods are scalable, they often cannot capture the complex, nuanced relationships between tasks and agents that occur in real situations, and may oscillate under dynamic environments, making them less suited to construction environments.

C. LLMs for Semantic Matching

Our recent work proposed integrating LLMs to interpret high-level or unstructured task descriptions and mapping them onto agent capabilities when numeric scores are insufficient in the context of role-playing games [9]. Some works claim LLMs act as in-context semantic ”reasoners” [10]. HVBTA advances this by embedding LLM reasoning directly into its multi-rule voting pipeline, automatically generating concise prompts to adjudicate between top candidates and feeding the resulting “preference score” back into the assignment.

D. Multi-Agent Path Finding

Multi-Agent Pathfinding (MAPF) focuses on finding collision-free paths for multiple agents moving simultaneously in a shared environment. Classical MAPF methods fall into three broad categories. **Centralized search** formulates the joint configuration space of all agents and applies global planners (e.g., A*, IDA*) to find an optimal solution. While providing completeness and optimality guarantees, centralized search scales exponentially with the number of agents and is impractical beyond small teams. **Decoupled or prioritized planning** (e.g., Priority-Based Search) assigns each agent a priority ordering and plans paths sequentially, treating higher-priority agents as moving obstacles. These methods run in polynomial time but may fail to find a solution even when one exists, due to priority conflicts. **Conflict-Based Search** and its bounded-suboptimal variants (e.g., ECBS) combine the strengths of centralized and decoupled approaches. CBS performs low-level single-agent planning independently, then detects pairwise collisions and incrementally adds binary constraints to a high-level search tree, resolving conflicts until all paths are conflict-free. This yields optimal or bounded-suboptimal solutions with dramatically better scalability than naïve centralized search. HVBTA relies on MAPF, in particular CBS, to translate its high-quality task assignments into collision-free execution plans. By front-loading capability-aware, voting-driven allocation, HVBTA reduces the number of conflicts that MAPF must resolve, speeding overall coordination.

III. The HVBTA Framework

The HVBTA framework represents a novel hybrid approach that integrates structured *Task Descriptions* and *Capability Profiles* with semantic reasoning from LLMs and dynamic path planning from Conflict-Based Search (CBS). While applicable to various domains, its structure is well-suited for the task allocation and coordination needs of multi-agent construction teams. Utilizing well-defined *Task Descriptions* and agent *Capability Profiles*, HVBTA can calculate the suitability of agents to tasks by generating a *Suitability Matrix*, it then votes on which agent should complete which task, and then guides that agent to the task using CBS. HVBTA also accounts for unseen tasks by taking advantage of the reasoning capabilities of a pretrained LLM to provide a suitability score through a previously assembled template prompt, thereby increasing

TABLE I
SIMPLE EXAMPLE OF *Task Descriptions* AND AGENT *Capability Profiles*

Task	**Payload Size**	**Terrain Type**	**Reach Needed**
Place Wall Panel	400 kg	Flat	2.0 m
Transport Module	300 kg	Uneven	2.5 m
Agent	**Payload Capacity**	**Terrain Capability**	**Reach**
A	500 kg	Flat	2.0 m
B	100 kg	Fixed	1.2 m
C	450 kg	Uneven	2.8 m

adaptability. The core components of the HVBTA framework are as follows.

A. Task Descriptions and Agent Capability Profiles

The framework begins by defining tasks and agents. Each construction effort, whether mapping a site, leveling ground using push manipulation, or placing a prefabricated module, is defined by a *Task Description*, a detailed set of requirements outlining what is necessary for successful execution. Concurrently, each agent is described through its *Capability Profile*, detailing its specific abilities, tools (e.g., general-purpose pushing tools), strengths (e.g., precision, lifting capacity, speed), and limitations. This dual representation allows for a rigorous comparison of agent competencies against task demands. Table I shows an example of two *Task Descriptions* and three *Capability Profiles*.

B. Suitability Matrix Generation

HVBTA constructs a quantitative *Suitability Matrix* to evaluate the compatibility between agents and tasks. For every possible agent-task pair, the matrix assigns a score that reflects how well the agent's capabilities align with the task's requirements. Scores are calculated using a rule-based approach that compares the agent's *Capability Profile* with the *Task Description*, resulting in higher scores for agents better suited to a task. For example, in Table I, while the suitability for both *Agent A* and *Agent C* would be very high for *Place Wall Panel*, only *Agent C* has the capability to *Transport Module*.

C. LLM Integration for Semantic Reasoning

In cases where rule-based scoring cannot definitively determine the suitability of an agent-task pairing, particularly when the task is unclear or novel or a robot is introduced with new capabilities, HVBTA integrates a pre-trained LLM to prompt for a score to evaluate suitability more holistically. The LLM utilizes its semantic understanding to interpret subtle nuances within both the *Task Descriptions* and agent *Capability Profiles*, refining the suitability assessment. An automatically generated prompt focuses the LLM on the specific component(s) whose compatibility is in question, allowing for more flexible assessment of suitability.

D. Voting and Allocation Mechanism

HVBTA employs a robust voting and allocation mechanism to resolve potential conflicts and balance assignments. This system leverages six distinct voting methods, such as Borda, approval, and majority voting, to aggregate and interpret the scores from the *Suitability Matrix*. This ensures that assignments are aligned with both the agents' capabilities and the tasks' requirements, while also considering the overall distribution of tasks across the heterogeneous team. Recognizing that a single, highly capable agent might be the best candidate (highest-scoring) for multiple tasks, HVBTA is capable of delegating tasks efficiently and not relying on the most capable agents. For example, in Table I, even though *Agent C* scores well on its suitability with both tasks, HVBTA would assign it to *Transport Module* because no other agent scored well on it. *Agent A* would be assigned to *Place Wall Panel* while *Agent B* would remain unassigned.

E. CBS for Path Planning

After task assignments are finalized, HVBTA integrates CBS for path planning. It plans optimal paths for the agents from their start positions to their designated tasks while avoiding obstacles and collisions with other agents in the physical environment. Path planning finds an optimal sequence of states, locations on a map, to move a robot from one location to another. This integration ensures efficient, collision-free spatial coordination, which is critical for the safe and timely execution of tasks. CBS can dynamically update agents' paths as the environment or an agent's state changes. Although HVBTA does not address task execution, task planning algorithms could also be integrated.

By automating decisions regarding task assignment and coordination that might previously require external intervention, HVBTA addresses key limitations of earlier approaches and enables the creation of efficient and dynamically managed automated workflows on a construction site.

IV. APPLYING HVBTA TO MULTI-AGENT CONSTRUCTION

The challenges of coordinating heterogeneous MAS for modular construction and site preparation tasks are significant. These tasks require a wide range of capabilities and frequently involve unforeseen requirements that must be addressed through contextual reasoning. HVBTA is particularly well-suited to address these challenges.

A. Handling Heterogeneity

HVBTA's emphasis on detailed agent *Capability Profiles* allows the framework to explicitly model the unique strengths, tools, and limitations of each heterogeneous agent in the team. *Task Descriptions* capture the specific requirements of different construction tasks. The *Suitability Matrix* quantitatively compares these profiles and descriptions, enabling the system to identify which agent is best equipped for a given task, whether it requires heavy lifting, fine manipulation, or robust pushing power. For example, a construction site with specialized robots for material transportation, dexterous manipulation, and site monitoring could use HVBTA to coordinate their behavior so that they are assigned to the tasks that each is best suited for, particularly when the robots have diverse skillsets and strengths.

B. Efficient Task Assignment

The combination of suitability scoring and multiple voting methods allows HVBTA to efficiently assign tasks across the entire team. This process considers not just the single best agent for a task, but how assignments can be balanced to utilize the entire team effectively, preventing bottlenecks where one agent is assigned multiple tasks while others are idle. The LLM integration provides the flexibility to resolve ambiguous suitability scores, leveraging contextual understanding to make more informed decisions based on subtle factors in agent capabilities or task needs that might not be captured by quantitative scores alone. For example, on a construction site with powerful winds, recently placed modules may need to be anchored to prevent them from becoming unmoored. Although the task lacks explicit physical requirements like in Table I, the LLM can infer that an agent with strong anchoring capability, weight, and size is going to be best suited for the task by leveraging contextual understanding of construction dynamics.

C. Seamless Coordination and Path Planning

The integration of CBS directly addresses the critical need for efficient and collision-free coordination on a busy construction site. Once tasks are assigned, CBS calculates optimal paths for agents to move to their work locations, considering the positions and planned movements of other agents and obstacles. This decentralized path planning component ensures that agents can operate in shared workspaces safely and efficiently, minimizing delays caused by congestion or collisions. CBS's spatio-temporal path planning abilities are vital for automated construction on a busy work site with multiple active agents and obstacles.

D. Adaptability

Although construction planning often involves predetermined steps, the dynamic nature of a construction site (e.g., unexpected obstacles, changes in material distribution) requires adaptable agent behavior. HVBTA's integrated approach, particularly with the LLM for nuanced suitability scoring and CBS for dynamic path re-planning, allows the system to delegate tasks efficiently while adapting to changing site conditions, task requirements, or the team of agents.

We are confident that HVBTA will allow multi-agent construction teams to achieve efficient, conflict-free assignment and coordination. This has the potential to facilitate potentially faster and more accurate modular assembly.

V. Conclusion

The increasing complexity of MAS and the structured demands of modular construction highlight the need for sophisticated task allocation and coordination frameworks. The Hybrid Voting-Based Task Assignment (HVBTA) framework draws inspiration from human reasoning to efficiently manage heterogeneous automated MAS. HVBTA manages this by rigorously defining agent *Capability Profiles* and *Task Descriptions*, leveraging a *Suitability Matrix*, employing robust voting mechanisms, using an LLM to resolve ambiguities, and incorporating CBS for efficient, collision-free spatio-temporal coordination. This framework is particularly well-suited for the challenges of modular construction and site preparation, enabling efficient task assignment and coordination of agents engaged in tasks like push manipulation and prefabricated assembly. Current work is focused on evaluating HVBTA's performance across various simulated scenarios involving diverse robotic platforms and task complexities. Ultimately, HVBTA holds significant promise for enhancing the efficiency, safety, and scalability of multi-agent construction.

References

[1] J. M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, and H. Owolabi, "Robotics and automated systems in construction: Understanding industry-specific challenges for adoption," *Journal of Building Engineering*, vol. 26, p. 100868, 2019.

[2] Y.-W. Lim, P. C. Ling, C. S. Tan, H.-Y. Chong, and A. Thurairajah, "Planning and coordination of modular construction," *Automation in Construction*, vol. 141, p. 104455, 2022.

[3] B. Xiao, C. Chen, and X. Yin, "Recent advancements of robotics in construction," *Automation in Construction*, vol. 144, p. 104591, 2022.

[4] Z. Chen, J. Alonso-Mora, X. Bai, D. Harabor, and P. Stuckey, "Integrated task assignment and path planning for capacitated multi-agent pickup and delivery," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 04 2021.

[5] A. Ali, D. M. Tilbury, and L. P. R. Jr, "Considerations for task allocation in human-robot teams," 2022.

[6] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, "Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art," *Robotics and Autonomous Systems*, vol. 168, p. 104492, 2023.

[7] L. Parker, "Alliance: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[8] B. Kalecı, O. Parlaktuna, M. Özkan, and G. Kirlik, "Market-based task allocation by using assignment problem," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp. 135–141, 2010.

[9] D. Weiner and R. Korpan, "Hybrid voting-based task assignment in role-playing games," in *HRI 2025 Dungeons, Neurons, and Dialogues: Social Interaction Dynamics in Contextual Games Workshop*, 2025.

[10] X. Tang, Z. Zheng, J. Li, F. Meng, S.-C. Zhu, Y. Liang, and M. Zhang, "Large language models are in-context semantic reasoners rather than symbolic reasoners," 2023.

Multi-object Rearrangement in Confined Spaces using a Car-like Robot Pusher

Jeeho Ahn[1] and Christoforos Mavrogiannis[1]

***Abstract*— We focus on push-based multi-object rearrangement planning using nonholonomically constrained mobile robots. The simultaneous geometric, kinematic, and physics constraints make this problem especially challenging. Prior work often relaxes some of these constraints by assuming dexterous hardware, prehensile manipulation, or sparsely occupied workspaces. Our key insight is that by capturing these constraints into a unified representation, we could empower a constrained robot to tackle difficult problem instances by modifying the environment in its favor. To this end, we introduce a Push-Traversability graph, whose vertices represent poses that the robot can push objects from, and edges represent optimal, kinematically feasible, and stable transitions between them. Based on this graph, we develop ReloPush, a planning framework that takes as input a complex multi-object rearrangement task and breaks it down into a sequence of single-object pushing tasks. We evaluate ReloPush across challenging scenarios, involving the rearrangement of up to nine objects, using a 1/10-scale robot racecar. Compared to two baselines lacking our proposed graph structure, ReloPush exhibits orders of magnitude faster runtimes and significantly more robust execution in the real world, evidenced in lower execution times and fewer losses of object contact.**

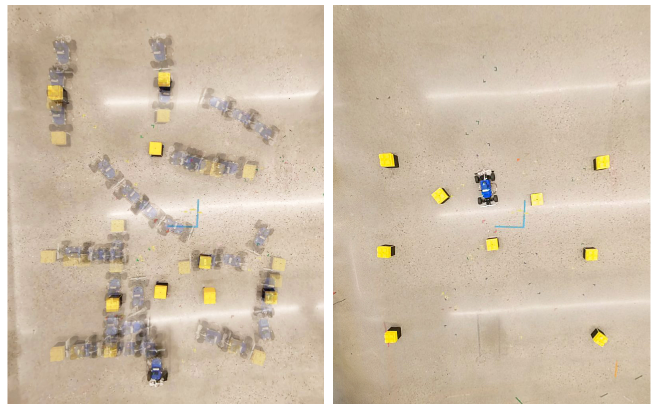
I. Introduction

Autonomous mobile robots have revolutionized fulfillment by offering a robust and scalable solution for large-scale rearrangement tasks. Fulfillment centers leverage extensive structure: robots often move across rectilinear rail grids, and make use of specialized docking mechanisms. This structure is missing from many other critical domains like construction, waste management, and small-to-medium warehouses. These domains give rise to rearrangement tasks involving objects of various shapes, and navigation among dense clutter while respecting boundary and kinematics constraints.

A practical approach to extending the range of rearrangeable objects is pushing, a class of *nonprehensile manipulation* that handles objects without requiring secure grasping [8]. This technique is appealing, as it enables manipulation of large, heavy, or irregularly shaped objects using relatively simple mechanisms. However, pushing introduces motion constraints: maintaining object stability requires avoiding abrupt turns and excessive accelerations.

To enable navigation among dense clutter in constrained workspaces, prior research addressed such challenges using the paradigm of *planning among movable objects* [13, 14], strategically modifying the environment to facilitate planning. Yet, these approaches typically involve dexterous manipulators capable of unconstrained grasping, neglect orientation constraints on goal object poses, and assume generous workspace boundaries.

[1]Department of Robotics, University of Michigan, Ann Arbor, USA. Email: {jeeho, cmavro}@umich.edu



(a) Executing a rearrangement plan. (b) Resulting rearrangement.

Fig. 1: In this work, we describe ReloPush [2], a planning framework for tackling multi-object rearrangement tasks with a nonholonomic mobile robot pusher.

Here, we focus on multi-object rearrangement via pushing within confined workspaces using a nonholonomic mobile robot pusher. Our key insight is that integrating geometric, kinematic, and physics constraints into a unified representation enables strategic environmental modification, thus facilitating complex rearrangement tasks. To this end, we introduce a *push-traversability* graph, where edges represent kinematically feasible and stable object displacements. Planning on this graph yields effective rearrangement plans for densely cluttered environments (Fig. 1). Extensive hardware experiments, including the creation of room-scale pixel art [1], underscore the robustness of our system. An extended version of this work appears at ICRA 2025 [2].

II. Problem Statement

We consider a mobile robot *pusher* and a set of m polygonal *blocks* in a workspace $\mathcal{W} \subset SE(2)$. We denote the state of the pusher as $p \in \mathcal{W}$ and the states of the blocks as $o_j \in \mathcal{W}$, $j \in \mathcal{M} = \{1, \ldots, m\}$. The pusher follows rear-axle, simple-car kinematics $\dot{p} = f(p, u)$, where u represents a control action (speed and steering angle), and may push objects using a flat bumper attached at its front. The goal of the pusher is to rearrange the blocks from their starting configuration, $O^s = (o^s_1, \ldots, o^s_m)$, to a goal configuration, $O^g = (o^g_1, \ldots, o^g_m)$. We seek to develop a planning framework to enable the pusher to efficiently rearrange all objects into their goal poses. We assume that the pusher has accurate knowledge of its ego pose at all times, and of the starting configuration of all objects, O^s.

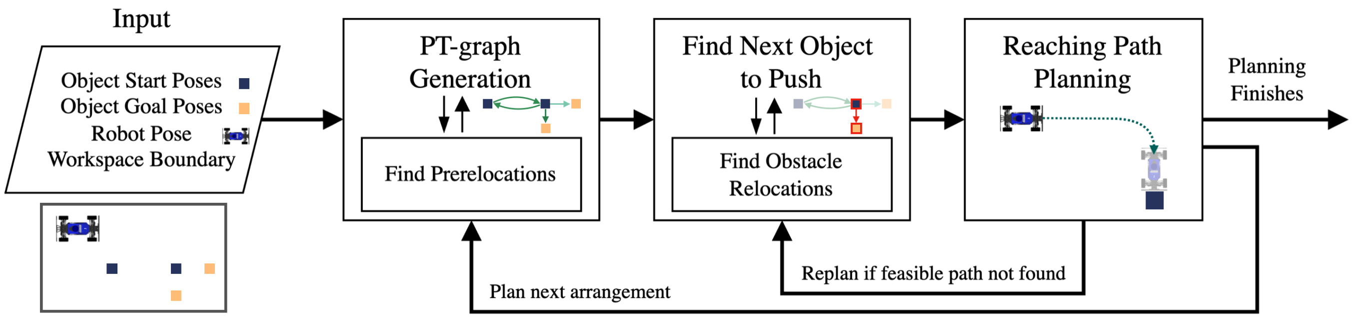
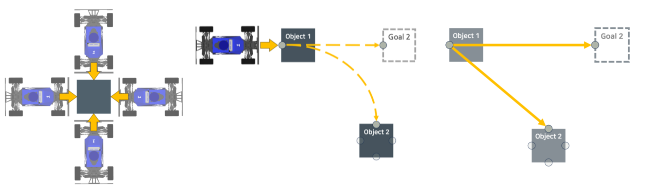
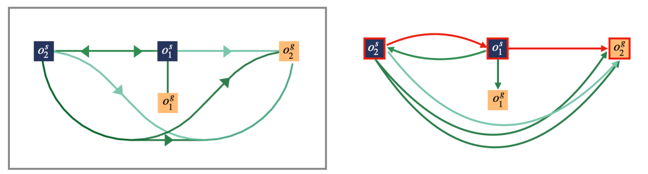


Fig. 2: The ReloPush architecture. Given the initial pose of the pusher and a rearrangement task in the form of start/goal object poses, ReloPush plans an efficient sequence of rearrangement subtasks to be executed by the robot via pushing.



(a) Pushing poses (b) Path Plan (c) PT-graph

Fig. 3: PT-graph generation. (a) First, every object is assigned K pushing poses (e.g., a cubic object has 4 pushing poses). (b) For any pair of pushing poses, we check if a collision-free path that respects the steering limit for quasistatic pushing can be drawn. (c) For each valid path, we construct a directed edge between its start/goal vertices.



(a) Object traversability. (b) PT-graph for the task in (a).

Fig. 4: (a) Two objects (navy squares) need to be rearranged to goal poses (yellow squares). (b) The PT-graph: nodes are pushing poses and edges are Dubins paths connecting them. By searching the graph, we can determine if any blocking objects need to be removed. For instance, the initial pose of object 1 is found to be blocking the shortest rearrangement of object 2 (red path).

III. RELOPUSH: NONPREHENSILE MULTI-OBJECT REARRANGEMENT

A. System Overview

Given a workspace $\mathcal{W}$, an initial robot pose p_s, and a set of objects that need to be reconfigured from their starting poses O^s to their goal poses O^g, ReloPush finds a sequence of rearrangements in a greedy fashion. It first constructs a rearrangement graph (PT-graph) that accounts for robot kinematics, push stability, and workspace boundary constraints. Using graph search, ReloPush searches for the collision-free object rearrangement path of lowest cost. If such a path is found, the graph is updated to mark the rearranged object as an obstacle, and the planner is invoked again to find the next rearrangement of lowest cost. If the path found passes through a blocking object, ReloPush displaces that object out of the way first. If the path violates the workspace boundary, ReloPush displaces the object to be pushed until the path to its goal meets the boundary constraint. If the path is infeasible (i.e., fails to find a motion to approach the object to push), it replans with next rearrangement candidate that has the next lowest cost. This process is repeated until a full rearrangement sequence for all objects is found. An overview of our architecture is shown in Fig. 2.

B. Push-Traversability Graph

A *traversability* graph (T-graph) is a representation of how movable objects can be reconfigured in a cluttered scene [9]. In its original form, vertices represent (starting and goal) positions of objects and edges represent collision-free transitions between them. By searching the graph, a collision-free rearrangement plan can be found.

Here, we build on the T-graph representation to introduce the *push-traversability* graph (PT-graph) $G(V, E)$, which not only captures the spatial relationships among movable objects but also integrates the kinematic constraints of the pusher and push-stability constraints of objects within the edges. Because in push-based manipulation of polygonal blocks, the block orientation is important, each vertex in our graph $v_i \in V$ represents a valid robot *pushing pose* p_i, i.e., a pose from which the pusher can start pushing a block (see Fig. 3).

For each vertex pair (v_s, v_g) representing start and goal *pushing poses*, we construct a directed edge if the optimal Dubins path [3] connecting them is collision-free and within workspace bounds. The optimal path uses left (L), right (R), and straight (S) motion primitives with a minimum turning radius ρ to ensure quasistatic pushing stability [4, 5, 7, 16]. Each valid edge is assigned a weight equal to the path length, with direction dictated by stability constraints (e.g., forward-only pushing).

C. Prerelocation: Change of Starting Pushing Pose

Often, an edge between two vertices cannot be formed because the connecting Dubins curve violates the workspace boundary, typically due to the limited turning radius ρ required for push stability (see Fig. 5). Our insight is that a slight adjustment of the initial pushing pose can yield an optimal, collision-free rearrangement path within workspace bounds. ReloPush leverages Dubins path classification [6, 10] to examine the case of the initial Dubins curve for rearrangement, namely *long-path* case or *short-path* case. When the start and goal poses are too closely located that it require large turning, a *short-path* case, ReloPush attempts to find

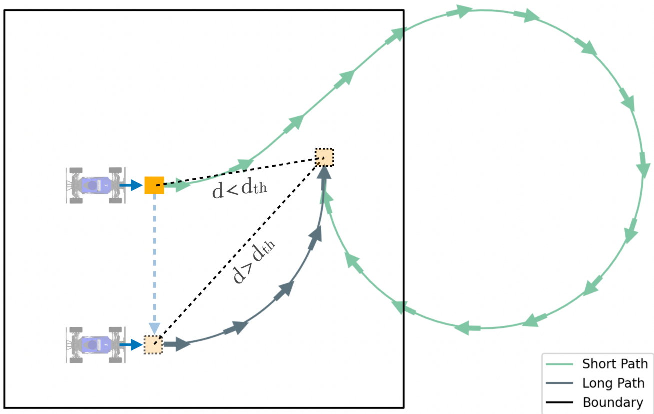


Fig. 5: Two Dubins curves with the same goal pose (top right) and maximum turning radius. When the start pose is too close to the goal ($d \leq d_{th}$), the resulting Dubins curve (green color) is a *Short Path* involving large turns violating the workspace boundary. Using Dubins path classification [6, 10], we can determine a *prerelocation* of the object's starting pose to allow reaching the goal via a *Long Path* ($d > d_{th}$) which will involve smaller turning arcs (gray color).

another pose to start from that makes it a *long-path* case. We refer to this change of starting pose as a *Prerelocation*.

D. Removing Blocking Objects

Extracting a rearrangement path plan can be done by searching the PT-graph using any graph search algorithm. The extracted path may include a vertex that is different from the start and goal vertex. If that is the case, then that vertex corresponds to an object that is physically blocking the rearrangement path. This object needs to be displaced before the plan can be executed. To do so, we follow a similar technique to how we plan *Prerelocations*: we find the closest relocation along the object's pushing directions (see Fig. 3a) that unblocks the path execution. This method of finding what object to remove is shown to be complete [9].

E. Analysis of the Algorithm

Theorem 3.1: Assuming a bounded number of pushing poses per object, K_{max}, the graph construction runs in polynomial time.

Proof: The number of vertices per object is bounded by K_{max}, thus, a fully connected graph in our case has $K_{max} \cdot m$ vertices. For each edge, a Dubins path is found in $O(1)$ and its collision checking is done in $O(1)$ assuming a bounded number of configurations checked due to our confined workspace. Searching a graph with Dijkstra's algorithm runs in $O(|V|^2)$ in a directed complete graph (the number of edges dominates). Thus, the runtime for a the rearrangement of m objects reduces to $O(m^3)$. To plan motion to approach an object, we invoke Hybrid A*, whose runtime also reduces to a polynomial expression on the number of objects assuming fixed workspace discretization, resolution of driving directions, and replanning attempts. ■

IV. Evaluation

A. Implementation

Experimental Setup. We implement our framework on MuSHR [12], an open-source 1/10th-scale mobile robot racecar, augmented with a 3D-printed flat bumper for pushing deployed in a workspace of area $4 \times 5.2m^2$. Across simulations and hardware experiments, we assume access to accurate robot localization (in real experiments, we make use of an overhead motion-capture system). We use objects of cubic shape with a side of $0.15m$ and a mass of 0.44 kg. The friction coefficient on the bumper-object surface was measured to be ~ 0.73.

Software. We implement our framework using the Open Motion Planning Library for Dubins path planning [15], and the code of Wen et al. [17] for Hybrid A* planning. Across simulated and real-world experiments, we use a Receding Horizon Controller (RHC) based on the implementation of the MuSHR [12] ecosystem. We run graph construction and search using Boost Graph Library [11]. We share our software implementation online at **https://github.com/fluentrobotics/ReloPush**.

Metrics. We evaluate our system with respect to the following metrics:

- S: Success rate – a trial is successful if a planner successfully finds a feasible rearrangement sequence.
- T_p: The time it takes to compute a complete rearrangement plan.
- L_t: The total length of the path that the robot travelled, including the reaching and pushing segments.
- N_{loss}: The total number of objects the robot lost contact with.
- T_e: The total time takes to execute a complete rearrangement plan.

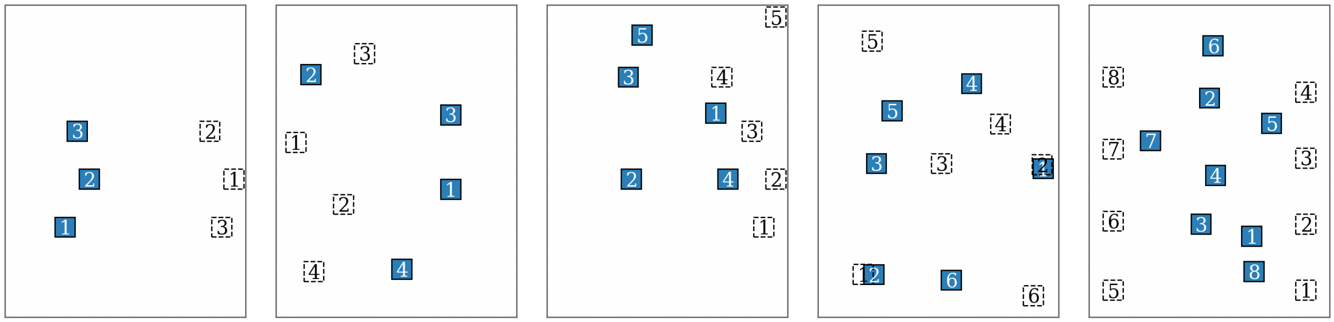
We also extract insights on the planning behavior of all algorithms using the following indices:

- N_{pre}: The total number of objects prerelocated to a feasible starting pushing pose (see Fig. 5).
- N_{obs}: The total number of removed blocking objects (see Fig. 4).
- L_p: The total length of path segments involving pushing.

Baselines. We compare the performance of ReloPush against two baselines:

- NO-PRERELO (NPR): a variant of RELOPUSH that also uses the PT-graph to handle nonmonotone cases but does not plan *prerelocations*. Instead, it invokes Hybrid A* if a collision-free Dubins path is found, to add edges.
- MP: a variant of our system that does not make use of the PT-graph at all but rather invokes Hybrid A* to plan a sequence of rearrangement tasks in a greedy fashion, and thus can only handle monotone cases.

Experimental Procedure. We consider a series of rearrangement scenarios of varying complexity (see Fig. 6) instantiated in simulation and the real world. To extract statistics on planning performance, we instantiated 100 trials of each scenario by locally randomizing the start and goal positions of objects within a range of $\pm 0.05m$ around the nominal instances of Fig. 6. To evaluate the robustness of our complete architecture, we executed the same scenarios in a physical workspace on a real MuSHR [12] robot. To ensure fairness in real-robot experiments, we chose instances



(a) $m = 3$ (b) $m = 4$ (c) $m = 5$ (d) $m = 6$ (e) $m = 8$

Fig. 6: Evaluation scenarios. Solid squares represent starting object poses and dashed squares represent goal poses.

TABLE I: Planning performance. Each cell lists the mean and the standard deviation over 100 trials per scenario.

Scenario	$m = 3$			$m = 4$			$m = 5$			$m = 6$			$m = 8$		
Algorithm	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP
S (%)	**100**	55	52	**100**	**100**	73	**80**	64	56	**89**	25	3	**86**	12	6
T_p (ms)	**40** (4.3)	1864 (141.1)	465 (67.4)	**86** (3.5)	5376 (131.5)	956 (36.0)	**146** (6.8)	9034 (581.5)	1175 (117.6)	**318** (20.8)	14489 (682.3)	1487 (57.4)	**529** (23.5)	25654 (2205.6)	2073 (149.5)
L_t (m)	**32.3** (3.1)	42.1 (5.2)	38.7 (7.3)	**40.3** (0.8)	45.5 (2.5)	45.1 (2.5)	**48.3** (3.5)	60.8 (6.5)	61.0 (9.1)	77.6 (5.5)	74.5 (15.2)	**62.9** **(1.0)**	**90.3** (6.0)	105.8 (4.2)	101.6 (3.5)

TABLE II: Planning behavior. Each cell lists the mean and the standard deviation over 100 simulated trials per scenario.

Scenario	$m = 3$			$m = 4$			$m = 5$			$m = 6$			$m = 8$		
Algorithm	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP
N_{pre}	0.9 (0.29)	-	-	1.0 (0.00)	-	-	1.6 (0.49)	-	-	3.0 (0.45)	-	-	4.4 (0.59)	-	-
N_{obs}	0.0 (0.00)	0.0 (0.00)	-	0.0 (0.00)	0.0 (0.00)	-	0.5 (0.52)	1.0 (0.25)	-	1.0 (0.00)	0.5 (0.50)	-	0.6 (0.49)	0.0 (0.00)	-
L_p (m)	**8.4** (0.1)	19.1 (4.4)	17.8 (4.3)	**8.5** (0.10)	18.1 (0.8)	17.3 (0.9)	**11.2** (0.5)	29.6 (10.1)	37.6 (7.2)	**13.9** (0.9)	37.2 (7.8)	35.6 (0.4)	**23.7** (0.76)	59.3 (3.1)	55.9 (4.0)

TABLE III: Results from real-world trials. Each cell lists the mean and the standard deviation over 5 trials per scenario.

Scenario	$m = 3$			$m = 4$			$m = 5$			$m = 6$			$m = 8$		
Algorithm	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP	RELOPUSH	NPR	MP
N_{loss}	**0.0** (0.0)	0.6 (0.49)	1.0 (0.0)	**0.0** (0.0)	0.6 (0.49)	1.0 (0.63)	**0.0** (0.0)	0.4 (0.49)	0.8 (0.75)	**0.2** (0.40)	1.4 (0.49)	1.2 (0.40)	**0.2** (0.4)	2.0 (0.63)	1.6 (0.49)
T_e (s)	**96.6** (0.07)	114.4 (0.49)	104.0 (4.23)	**121.7** (0.32)	137.4 (1.05)	133.7 (0.37)	**148.1** (0.66)	154.0 (1.46)	235.8 (1.71)	256.2 (3.08)	198.9 (2.74)	**182.97** (1.20)	**274.4** (0.33)	314.9 (2.28)	285.9 (1.40)

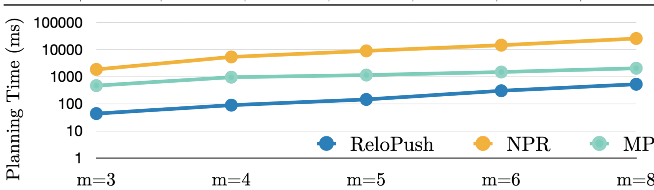
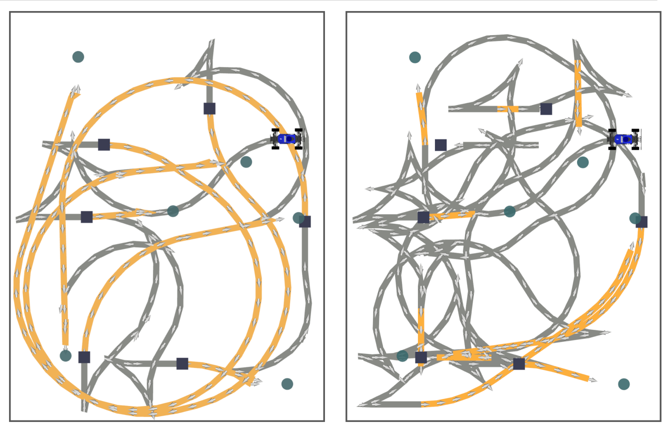


Fig. 7: Average planning time across scenarios, shown in logarithmic scale. ReloPush scales well with the number of objects compared to baselines.

where all algorithms were successful in planning. We ran each scenario 5 times per algorithm.

B. Results

Planning Performance. ReloPush dominates baselines in success rate and planning time (see Table I, Fig. 7). The gap becomes more pronounced as the clutter (number of objects) increases. This happens because increased clutter is more likely to lead to nonmonotone instances. For example, most $m = 6$ instances are nonmonotone because two objects overlap with goals of other objects. Since MP can only handle monotone rearrangements, it fails more frequently in these harder instances. It is also worth observing that kinematic constraints make some instances harder to solve *regardless* of the number of objects. For example, some of the $m = 3$ instances are challenging because o_2^s is situated so close to its goal o_2^g that the optimal path connecting them goes out of the boundary. In contrast, *ReloPush* handled this scenario effectively via *prerelocation*. Table II provides intuition on the planning decisions that ReloPush made enabled increased performance. As clutter increases, ReloPush makes increasingly more workspace modifications (prerelocations and blocking-object removals).



(a) MP (b) RELOPUSH

Fig. 8: Paths generated by MP (a) and RELOPUSH (b) for the $m = 6$ scenario. Squares and circles represent respectively start and goal object poses. Continuous lines represent planned paths with pushing segments shown in yellow. RELOPUSH plans substantially shorter pushing segments to minimize the risk of losing contact with an object during execution.

Real Robot Experiments. ReloPush never lost contact with any objects, in contrast to baselines (see Table III). One reason for that is that ReloPush maintains low pushing path length (for better or similar total path length) as shown in Table II. The shorter the pushing distance, the lower the risk of losing the object due to model errors and uncertainties (see Fig. 8). A video with footage from our experiments can be found at https://youtu.be/_EwHuF8XAjk.

References

[1] J. Ahn and C. Mavrogiannis. From A to Ω: Pixel art with a mobile robot. In *Special Session: Arts in Robotics, IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[2] J. Ahn and C. Mavrogiannis. ReloPush: Multi-object rearrangement in confined spaces with a nonholonomic mobile robot pusher. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[3] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516, 1957.

[4] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction part 1. limit surface and moment function. *Wear*, 143(2):307–330, 1991.

[5] J. King, M. Klingensmith, C. Dellin, M. Dogar, P. Velagapudi, N. Pollard, and S. Srinivasa. Pregrasp manipulation as trajectory optimization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2013.

[6] J. Lim, F. Achermann, R. Bähnemann, N. Lawrance, and R. Siegwart. Circling back: Dubins set classification revisited. In *Workshop on Energy Efficient Aerial Robotic Systems, International Conference on Robotics and Automation (ICRA)*, 2023.

[7] K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[8] K. M. Lynch and M. T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999.

[9] C. Nam, S. H. Cheong, J. Lee, D. H. Kim, and C. Kim. Fast and resilient manipulation planning for object retrieval in cluttered and confined environments. *IEEE Transactions on Robotics*, 37(5):1539–1552, 2021.

[10] A. M. Shkel and V. Lumelsky. Classification of the dubins set. *Robotics and Autonomous Systems*, 34(4): 179–202, 2001.

[11] J. Siek, L.-Q. Lee, and A. Lumsdaine. Boost graph library. http://www.boost.org/libs/graph/, June 2000.

[12] S. S. Srinivasa, P. Lancaster, J. Michalove, M. Schmittle, C. Summers, M. Rockett, R. Scalise, J. R. Smith, S. Choudhury, C. Mavrogiannis, and F. Sadeghi. MuSHR: A low-cost, open-source robotic racecar for education and research, 2019. *arXiv:1908.08031* [cs.RO].

[13] M. Stilman and J. Kuffner. Planning among movable obstacles with artificial constraints. *The International Journal of Robotics Research*, 27(11-12):1295–1307, 2008.

[14] M. Stilman, J.-U. Schamburek, J. Kuffner, and T. Asfour. Manipulation planning among movable obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3327–3332, 2007.

[15] I. A. Sucan, M. Moll, and L. E. Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[16] S. Talia, A. Thareja, C. Mavrogiannis, M. Schmittle, and S. S. Srinivasa. PuSHR: A multirobot system for nonprehensile rearrangement. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Sytems (IROS)*, pages 5380–5387, 2023.

[17] L. Wen, Y. Liu, and H. Li. Cl-mapf: Multi-agent path finding for car-like robots with kinematic and spatiotemporal constraints. *Robotics and Autonomous Systems*, 150:103997, 2022.

SPACE: 3D Spatial Co-operation and Exploration Framework for Robust Mapping and Coverage with Multi-Robot Systems

Sai Krishna Ghanta   Ramviyas Parasuraman

***Abstract*— Multi-robot systems hold promise for accelerating cooperative construction tasks such as site preparation and modular assembly. However, dynamic inter-robot occlusions in 3D point-cloud mapping introduce ghosting artifacts that compromise surface reconstruction accuracy and impede downstream planning for grading and leveling. Furthermore, traditional 2D grid-based frontier approaches fail to capture volumetric nuances in partially reconstructed areas, limiting exploration. We propose SPACE, a semi-distributed framework that (1) employs geometric mutual-awareness coupled with image-plane clustering to suppress dynamic robot artifacts, and (2) introduces a bi-variate frontier detection and assignment scheme that classifies and prioritizes both unexplored and weakly mapped regions. SPACE achieves up to 99% reduction in ghosting volume and 95% exploration coverage in ROS-Gazebo experiments and real-world experiments.**

I. Introduction

Robotic site preparation, including mapping, grading, and modular assembly, can make construction safer and faster, but only with precise 3-D surface models [1]. Moreover, accurate 3D surface models are essential for tasks like material distribution, topographic grading, and aligning prefabricated components [2]. However, when multiple dynamic robots map simultaneously with overlapping fields of view, there can be "ghosting trails" in the merged point cloud, obscuring true surface geometry and hindering downstream tasks as shown in Fig. 1 [3]. Despite its impact, there has been limited research on addressing the ghosting trail effect due to the dynamics of robots, particularly in the multi-robot exploration (MRE) context. Furthermore, frontier-based exploration methods built on 2D occupancy grids neglect volumetric gaps in the 3D global maps, leaving weakly mapped regions unaddressed [4], [5]. These limitations constrain reliability in complex construction sites. We propose SPACE, a semi-distributed pipeline for multi-robot mapping and exploration tailored to construction scenarios, which:

- employs a geometric mutual-awareness test and a dynamic robot filter to eliminate inter-robot occlusions in visual mapping,
- detects unexplored and weakly explored 3D frontiers and balances exploration with dense reconstruction via a frontier-importance framework.

Extensive simulations show that SPACE outperforms RTAB-MAP [6], Kimera-Multi [7], and exploration baselines such as RRT [8], DRL-Voronoi [9], and SEAL [10]. We release SPACE as an open-source ROS package[1] to support adoption and further research.

School of Computing, University of Georgia, Athens, GA 30602, USA.
Author emails: {sai.krishna;ramviyas}@uga.edu

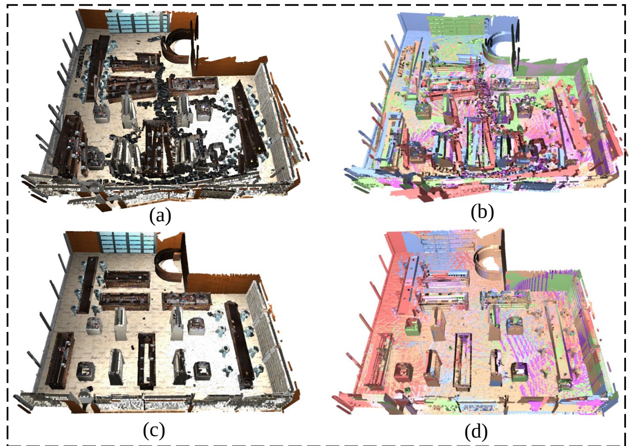


Fig. 1: **Ghosting Trail Problem:** in 3D maps due to inter-robot occlusions. (a,b) RRT [8]+Kimera-Multi [7]; (c,d) **SPACE**+RTAB-MAP [6].

II. Architectural Overview

SPACE is a semi-distributed framework in which each of the n robots $R = \{r_1, \ldots, r_n\}$ runs on-board Visual SLAM, mutual awareness, dynamic robot filtering, frontier validation and local path planning to build a local point-cloud map $\mathscr{P}_i$, while an edge/central unit continuously fuses these into a global map $\mathscr{P}^*$, refines global odometry using known initial poses, and performs 3D map merging [11], 3D frontier detection and frontier assignment (Fig. 2). SPACE integrates readily with existing visual SLAM packages (e.g. RTABMap [6], Kimera-Multi [7]) and exploration packages [10].

A. Mutual Awareness & Dynamic Robot Filter

We consider n robots denoted by the set $\mathtt{R} = \{r_1, r_2, \ldots, r_n\}$, each with the known initial position in a global frame, which is continually obtained/refined using a map merging process. Given observer r_o and target r_t with poses $\mathbf{p}_o = (X_o, \psi_o)$, $\mathbf{p}_t = (X_t, \psi_t)$ in the global frame, we first estimate proximity $\|\mathbf{p}_o - \mathbf{p}_t\| \leq R$, where R is the RGB-D sensor range. If the target is in proximity, we estimate whether the target (of angular size $\alpha = \arctan\left(\frac{\gamma}{\|\mathbf{p}_o - \mathbf{p}_t\|}\right)$ with

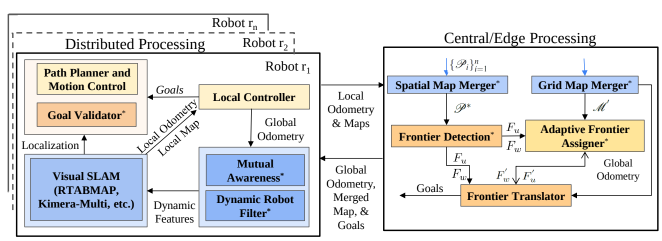


Fig. 2: Overview of the proposed methodology.

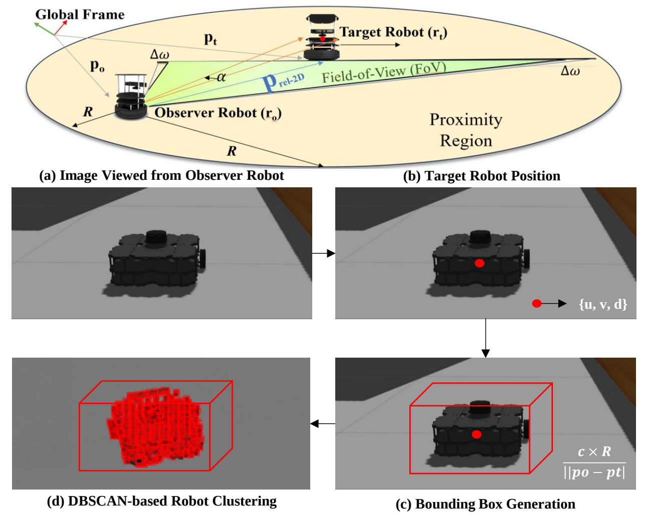


Fig. 3: (Top) Conceptual diagram of the mutual-awareness module, showing how an observer tracks a target within its FoV and proximity region; (Bottom) DRF pipeline: the target's position is estimated in camera frame; (c) perspective-scaled bounding boxes are generated; (d) DBSCAN filtering.

γ the robot radius) is in Field of View (FoV) of observer. We then convert 2D relative vector $\mathbf{p}_{\text{rel}} = X_t - X_o$ to 3D camera frame of observer and project target position via intrinsic $\mathbf{K}$ and extrinsic $[\mathbf{R}_{\text{ext}} \mid \mathbf{T}_{\text{ext}}]$ parameters of RGB-D sensor of observer. A perspective-scaled bounding box of size $\propto \frac{cR}{d}$ is centered at (u, v), and DBSCAN clusters are used to remove dynamic target robot features as shown in Fig. 3.

B. Bi-variate Frontier Detection & Assignment

With global point cloud $\mathscr{P}^*$, a bi-variate frontier detector computes for every down-sampled point p the neighborhood density $\rho(p)$ inside radius r_s and the variance $\sigma^2(p)$ inside χ as shown in Fig. 4. The unexplored frontiers F_u are the loci with $\rho < \delta_d$, whereas weakly mapped frontiers F_w satisfy $\sigma^2 > \delta_v$. After projection of these frontiers onto a 2-D grid $\mathscr{M}'$, a frontier translator yields $f \in \{F'_u, F'_w\}$. The robots are assigned to these frontiers based on frontier importance and non-probabilistic information gain. Moreover, exploration strategy is designed based on exponential formulation to prioritize the exploration of the unexplored regions initially, and over time, the importance shifts towards weakly mapped regions.

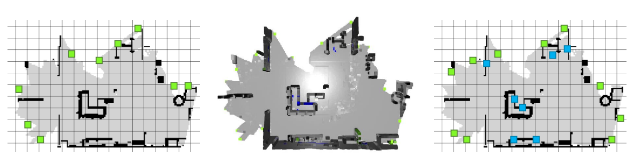


Fig. 4: (L) 2D Frontier Detection [12]; (C) SPACE Frontier Detection; (R) Translated Spatial Frontiers.

III. Experimental Results and Analysis

We implemented SPACE in ROS-*Gazebo* with Turtlebot Waffle robots, using RTAB-Map and Kimera-Multi for local SLAM, merging 2D grids via multi-robot-map-merge [2], and navigating with move-base[3] (A* for global planning, DWA for local avoidance). Experiments were conducted in two fully constructed indoor simulation environments—AWS House[4] (70m^2) and AWS Bookstore[5] (100m^2)—with three robots and six robots respectively. Each Turtlebot, equipped with a realsense camera ($\text{FoV}_{\text{cam}} = 84.1°$, range=5m).

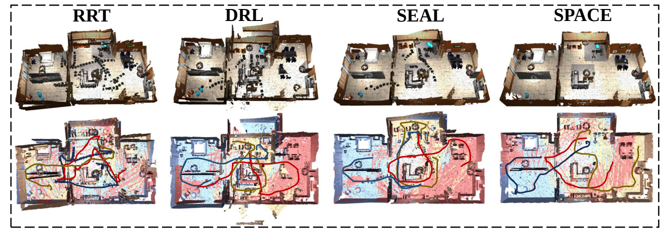


Fig. 5: The top row and bottom row presents the 3D Reconstruction Maps and colored local maps with trajectories in AWS House with 3 robots.

SPACE integrated mapping with RTAB-Map [6] and Kimera-Multi [7] achieved faster mapping times, shorter travel distances, and near-complete 2D/3D coverage, while reducing overlap and ghosting volumes by over 99.8% as shown in Fig. 5. The mapping RMSE remained low and stable, reflecting the effectiveness of the mutual awareness and dynamic robot filtering modules. SPACE exploration outperformed RRT, DRL, and SEAL in simulation, achieving up to 14.3% more coverage in less time and distance. Over multiple scenarios, SPACE explored 90% of environments up to 38% faster and with significantly lower overlap and RMSE compared to benchmarks. In real-world TurtleBot2e experiments, SPACE reduced ghosting volumes by 95–99% across 10 room and corridor trials as shown in Fig. 6. These real-world tests confirmed the framework's practicality, achieving reliable 3D reconstructions without ground-truth models.

In conclusion, SPACE offers a robust semi-distributed multi-robot exploration pipeline for indoor environments, optimizing both mapping accuracy and exploration efficiency. The pipeline's computational complexity scales near-linearly with the number of robots, enabling real-time performance even for larger teams. Future work will extend SPACE to outdoor UAV-based exploration, further enhancing adaptability and 3D mapping capabilities.

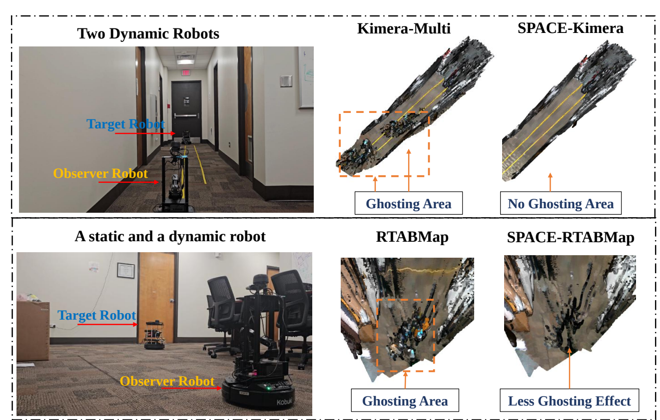


Fig. 6: Snapshots of the real-world robots performing multi-robot spatial mapping. The SPACE approach reduced the ghosting regions significantly compared to the state-of-the-art approaches. The corridor scenario (top) has a higher robot density, resulting in $\approx 99\%$ reduction in the ghosting effect.

REFERENCES

[1] T. Bock and T. Linner, *Robot-Oriented Design: Design and Management Tools for the Deployment of Automation and Robotics in Construction*. Cambridge University Press, 2015.

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[3] R. Sequeira, H. Ferreira, G. Pereira, and U. Nunes, "Multi-robot decentralized occupancy grid merging using dempster–shafer," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6110–6116.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 1997, pp. 146–151.

[5] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[6] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[7] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[8] L. Zhang, Z. Lin, J. Wang, and B. He, "Rapidly-exploring random trees multi-robot map exploration under optimization framework," *Robotics and Autonomous Systems*, vol. 131, p. 103565, 2020.

[9] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 413–14 423, 2020.

[10] E. Latif and R. Parasuraman, "Seal: Simultaneous exploration and localization for multi-robot systems," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5358–5365.

[11] S. Yang, H. Zhao, X. Jiang, and J. Wang, "Multi-ground-robot system 3d map merging method based on global localization and robust icp," in *2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*. IEEE, 2023, pp. 18–22.

[12] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.

NOTES

[1] https://github.com/herolab-uga/SPACE-MAP

[2] https://wiki.ros.org/multirobot_map_merge

[3] https://wiki.ros.org/move_base

[4] https://github.com/aws-robotics/aws-robomaker-small-house-world

[5] https://github.com/aws-robotics/aws-robomaker-bookstore-world

NASA ARMADAS Approach to Collaborative Multi-Robot Construction

Kenneth Cheung[1], James Gloyd[2], Irina Kostitsyna[2], Tom Peters[3], Greenfield Trinh[1]

Abstract

Collaborative multi-robot autonomous construction enables infrastructure development in extreme environments and enhances operational efficiency in high-performance applications. We present our algorithmic approach to multi-robot modular construction, based on the NASA Automated Reconfigurable Mission Adaptive Digital Assembly Systems (ARMADAS) framework, in which multiple types of small robots collaboratively build large lightweight lattice structures.

Or solution is comprised of two interdependent components: (1) a planner of the order of construction, and (2) a multi-robot path planner. The planner of the order of construction iteratively identifies each next location for attaching building blocks based on the changing state of the system. The multi-robot path planner computes a collision-avoiding schedule for the robots to bring a building block to the specified location and attach it.

The class of structure geometries that ARMADAS robots can build is much wider than the class of histogram shapes, to which other multi-robot construction systems are often limited. Many of the techniques we use can be applied to any robotic assembly system whose robots perform locomotion over the structure that they are building.

I. INTRODUCTION

In-space assembly is crucial for advancing human space presence and in-space capabilities, as it allows for the construction of large structures that would otherwise be impractical or impossible to deploy due to size and weight constraints. While human directed and performed assembly is possible with extra-vehicular activity by astronauts, a markedly more advantageous option is to use autonomous robotic agents. We need not look far to see benefits provided by robotic automation terrestrially, and many of the same benefits to productivity and efficiency can be achieved through automation in space as well. One key benefit provided by robotic automation for in-space assembly is scalability, specifically, being able to increase project size, scope, and efficiency by simply supplying additional robotic teams, which then work together in parallel. Alongside this scalability comes an ability to distribute and redistribute robotic agents and teams to multiple endeavors at a time, adapting to changing requirements, goals, and conditions.

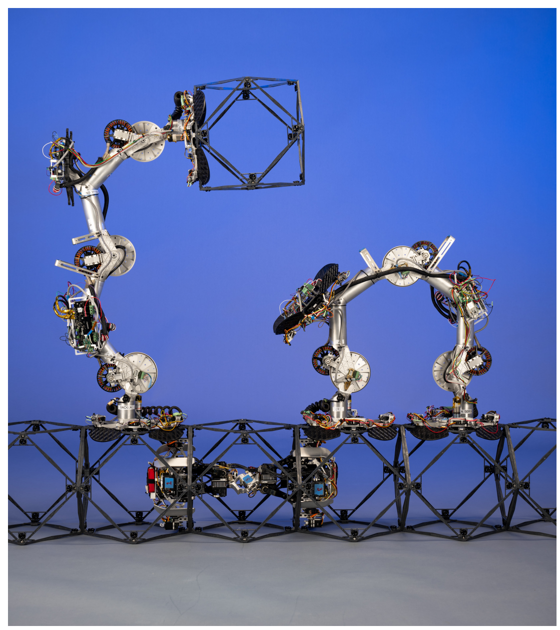


Fig. 1. Two types of ARMADAS robots on a row of voxels. A crane SOLL-E (left) is picking up a voxel from the backpack of a cargo SOLL-E (right). MMIC-I (inside the structure) is poised to crawl into the next voxel placed by the crane SOLL-E and fasten the voxel to the structure.

The NASA Automated Reconfigurable Mission Adaptive Digital Assembly Systems (ARMADAS) project [1] has developed a highly modular and versatile multi-robot assembly system. Structures produced by the ARMADAS system are comprised of a class of ultra-light weight and strong

*This work was supported by NASA Game Changing Development (GCD) Program, Space Technology Mission Directorate

[1]Coded Structures Lab, NASA Ames Research Center, Moffett Field, CA
[2]KBR, Coded Structures Lab, NASA Ames Research Center, Moffett Field, CA
[3]TU Eindhoven, Eindhoven, the Netherlands

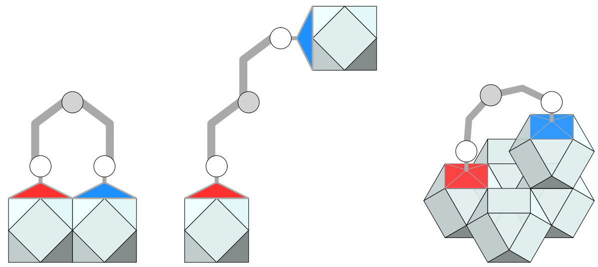


Fig. 2. Three examples of the 31 possible SOLL-E poses on the surface of a structure.

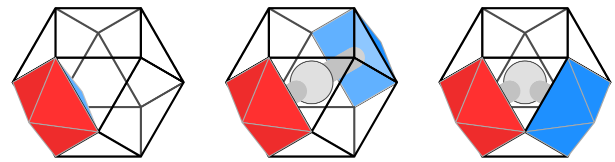


Fig. 3. The three poses of MMIC-I inside a structure, up to rotation.

materials, which consist of unit-size building blocks called *voxels*. These strut-based building blocks are assembled together into discrete grid patterns by a crew of two types of robots (see Fig. 1): the Scaling Omni-directional Lattice Locomoting Explorer (SOLL-E) [2] and the Mobile Metamaterial Internal Co-Integrator (MMIC-I) [3]. SOLL-E is a bipedal robot that can walk along the surface of the structure and can carry voxels in its backpack. It grips to the external faces of the voxels in the structure, and, unlike robots in other robotic construction systems, is not constrained by gravity: it can walk vertically on walls of the structure and even upside down on the ceiling. MMIC-I is an internal robot that can crawl through the structure and fasten voxels together with its bolting mechanism. Working together in teams, these robots can assemble voxels into various structures, depending on the instructions the robots are given.

While the benefits are numerous, the task of autonomous assembly of large and complex structures presents a non-trivial algorithmic challenge. For structures consisting of thousands of building blocks, it is impossible to optimize the assembly plan by hand, and the level of sustained economic and exploration activity envisioned for in-space (and on lunar surface) presence by the government and commercial partners would require structures on a much larger scale [4]. This shortfall in current capabilities is the directive motivation for this work, and in this note, we present an algorithmic framework for a multi-robot assembly system to address exactly that.

A. Algorithmic Approach to Assembly

Over the past decade, the topic of multi-robot construction has been gaining popularity in academic circles. The approach of having a large number of small and agile mobile construction robots operating collaboratively makes for an appealing alternative to traditional assembly lines with their massive and expensive robotic arms. An already classic example of such an assembly system is TERMES [5], developed to perform construction by a fleet of car-like robots in a decentralized fashion. TERMES robots place box-shaped building blocks to form a modular structure on a grid.

Subsequent works have studied the combinatorial optimization question of construction planning for similar systems [6]–[8]. These systems operate under the constraint of gravity: robots must be supported by the building blocks underneath them, and the building blocks themselves can be placed only on top of other building blocks. Thus the gravity constraint limits the feasible geometries of the structures to the class of *histogram* shapes. With constraints on vertical traversal capabilities in these systems' robots (i.e., a robot can climb or descend a limited number of blocks in a single horizontal step, usually one or two), construction of some geometries requires the robots to construct and deconstruct ramp-like supports to access certain build regions. In some instances, such ramps can be constructed by rearranging existing building blocks, but in general, building the required supporting structures necessitates additional building blocks beyond those used in the final geometry.

A number of other systems feature two-legged robots manipulating cubic building blocks [9]–[12], but these are often similarly restricted by gravity.

In contrast, ARMADAS robots are not subject to the same gravity-based restrictions. SOLL-E can walk on walls and ceilings, and MMIC-I can crawl through a structure in any direction. As a result, the geometry of the ARMADAS structures is not limited to histogram structures, since the voxels are directly bolted to one another. The only restriction we make is that the structure must be face-connected.

The ARMADAS system separates the functionalities of transporting a voxel and attaching it to the structure between robotic entities. This separation of robot roles was inspired by a desire to minimize the energy usage, in particular, by minimizing the mass associated with particular functionalities. For example, we avoid adding the bolting mass to robots that are required to repeatedly traverse long distances to and from the building block source [13]. Another potential benefit, although not considered in the current paper, is that a single crane robot can place building blocks while multiple other robots act as couriers to bring material from the source. While separating the robot roles increases build efficiency and versatility of the system, this also increases the challenge and complexity of path planning for the robots.

Existence and order of assembly: Closely related to the construction systems studied in this paper are the abstract models of modular reconfigurable robots explored in the algorithmic community [14]–[17]. These works explore questions of universal reconfiguration: is it possible to reconfigure from any modular shape to any other shape? Recent work [18] introduces a reconfiguration model with realistic movement constraints motivated by the ARMADAS system. The authors show that with the use of additional voxels as a scaffold, a structure of any shape can be assembled. In the case when no scaffold voxels are allowed, i.e., voxels are only ever added to the structure and never removed, there exist shapes that are impossible to assemble. Nevertheless, a large class of shapes, specifically shapes with the so called external-feature-size-2 property, can always be assembled in a monotone additive fashion.

Multi-robot path planning: One of the major tasks in multi-robot assembly planning is robot path planning. The robots have to travel from the depot, where they pick up new voxels, to the location of their placement, all while avoiding collisions and deadlocks with other robots. Multi-robot path planning is a vastly researched topic, with approaches ranging from exact solutions (e.g., A* planning in highly-dimensional state spaces, conflict-based search [19] and its variations) to heuristic solutions (e.g., decoupled approaches [20], AI-based approaches [21]).

Particularly relevant to our setting is the lifelong multi-robot path planning [22], where robots receive new tasks and targets upon finishing previous ones. This setting renders the exact approaches to planning impractical even for a small number of robots due to the exploding complexity of the problem state representation. The solution proposed in [22] uses a decoupled approach, iteratively constructing robots trajectories task by task and robot by robot. The authors introduce the notion of *endpoints*, serving as safe spaces for robots to park temporarily to avoid potential conflicts with new targets and future actions of other robots.

Another technique we utilize was presented in the context of the multi-labeled A* (MLA*) algorithm [23]. The state of a robot is incorporated into the search graph. This can be useful, for example, in the warehouse setting, where robots already carrying a package have different state than the robots moving to pick up a package. Since certain varieties of our robots transport building blocks, it is easy to see how this inclusion is appropriate.

B. Problem Statement

The robotic part of the ARMADAS system consists of multiple teams of robots, each team working independent of others. A team is composed of a *cargo* SOLL-E, a *crane* SOLL-E and one MMIC-I. The cargo SOLL-E is responsible for picking up a new voxel at the depot location and bringing it in its backpack to the assembly location, where the crane SOLL-E picks the voxel up from the cargo's backpack and places it on the structure. Then MMIC-I enters the new voxel and fastens it to the structure along every face adjacent to the existing structure.

As ARMADAS robots can only exist on (or in) a structure, we assume that the initial state of the system includes a seed structure with a specified depot position, and a supply of voxels ready to be served to cargo robots at the depot. The role of the seed structure is to support the robots' initial positions, as well as serving as a seed structure from which the target structure will be assembled by attaching voxels one by one. Thus, we can formulate the problem as: Given a seed structure S, a target structure T, and k teams of robots, our goal is to find an efficient construction plan for the robots to build T while avoiding collisions and deadlocks.

II. CONSTRUCTION PLANNING FRAMEWORK

Construction order planning: To enable precise alignment and to add more stability to the final structure, voxels have mechanical alignment features (see Fig. 4). Consequently, for fine positioning of a voxel being attached to the structure, it must be slid from a position some small distance away from the existing faces. That is, the placement of a voxel requires a certain amount of additional space around the unit cell of its target location. This leads, in particular, to it being impossible for a voxel to move through a unit-wide gap between two other voxels. A new voxel must always be attached at an end of a row and/or column. This constraint imposes a direct ordering on all connected voxels in a row (in any orientation) as soon as one of them has been placed. This significantly limits the possible options on the order of assembly, and makes the optimization problem challenging.

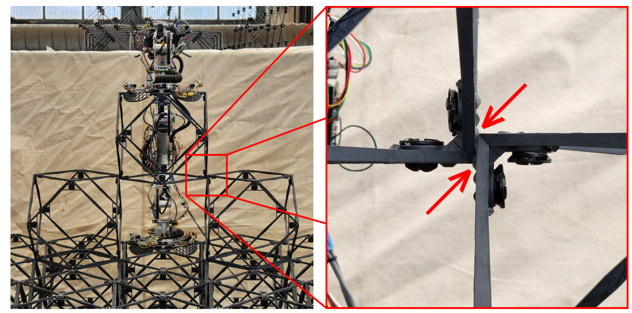


Fig. 4. Photograph of armadas robot attempting to place a voxel between two other voxels and failing due to collisions of mechanical alignment features (red arrows) [18].

Our approach to this problem is to partition the target structure geometry hierarchically in simple sub-components, and fix the order of assembly within and between them. These components will be dynamically assigned to robot teams, ensuring the teams are well-separated in space an in time, thus, maximizing parallelism of the construction process. For details on our approach refer to [24].

Path planning: Many grid-based multi-robot path planning approaches [6]–[8], [10] operate under the simplifying assumption that each robot fits into one cell of the grid. However, ARMADAS robots occupy multiple grid cells, which makes system state representation more complex, and both, collision detection and deadlock avoidance, more challenging. To address these challenges, we develop generalizations of several approaches to multi-robot path planning and combine them into the path-planning framework presented here. Despite our framework being developed specifically for the ARMADAS system, the underlying algorithms can be applied to other robotic assembly systems that feature a robot / building block codesign.

One approach to multi-robot motion planning is to consider the robots as one *meta*-agent, and represent their states in a high-dimensional state space. This reduces the task to shortest path planning in the graph encoding the states of the meta-agent and transitions between them, and results in an exact optimal solution to the problem. Unfortunately, even for one team of ARMADAS robots and a very small structure, the number of states in the meta-agent state space can be on the order of billions. We therefore must sacrifice optimality of the solution in favor of efficiency of computation. We decouple the state representation of the individual robots, and take the approach of planning each next step in the construction process for each robot individually. In order to avoid collisions, we develop a data structure that tracks robots' paths and reserves the occupied grid cells during the corresponding time intervals. To avoid deadlocks, we use an approach similar to the one proposed in [22], where we introduce a *parking* location reserved for each robot, which the robots can use to avoid blocking others.

For more details on our approach to path planning refer to [25].

References

[1] C. E. Gregg, D. Catanoso, O. I. B. Formoso, I. Kostitsyna, M. E. Ochalek, T. J. Olatunde, I. W. Park, F. M. Sebastianelli, E. M. Taylor, G. T. Trinh, and K. C. Cheung, "Ultralight, strong, and self-reprogrammable mechanical metamaterials," *Science Robotics*, vol. 9, no. 86, 2024. doi: 10.1126/scirobotics.adi2746

[2] I.-W. Park, D. Catanoso, O. Formoso, C. Gregg, M. Ochalek, T. Olatunde, F. Sebastianelli, P. Spino, E. Taylor, G. Trinh *et al.*, "SOLL-E: A module transport and placement robot for autonomous assembly of discrete lattice structures," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 736–10 741.

[3] O. Formoso, G. Trinh, D. Catanoso, I.-W. Park, C. Gregg, and K. Cheung, "MMIC-I: A robotic platform for assembly integration and internal locomotion through mechanical meta-material structures," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7303–7309.

[4] "Moon to Mars architecture." [Online]. Available: https://www.nasa.gov/moontomarsarchitecture/

[5] K. Petersen, R. Nagpal, J. Werfel *et al.*, "TERMES: An autonomous robotic system for three-dimensional collective construction." in *Robotics: science and systems*, vol. 7, 2011, pp. 257–264.

[6] E. Lam, P. J. Stuckey, S. Koenig, and T. S. Kumar, "Exact approaches to the multi-agent collective construction problem," in *Principles and Practice of Constraint Programming: 26th International Conference, CP 2020, Louvain-la-Neuve, Belgium, September 7–11, 2020, Proceedings 26*. Springer, 2020, pp. 743–758.

[7] A. K. Srinivasan, S. Singh, G. Gutow, H. Choset, and B. Vundurthy, "Multi-agent collective construction using 3D decomposition," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9963–9969.

[8] M. Rameš and P. Surynek, "Action duration generalization for exact multi-agent collective construction," in *16th International Conference on Agents and Artificial Intelligence (ICAART)*, 2024.

[9] Y. Terada and S. Murata, "Automatic assembly system for a large-scale modular structure - hardware design of module and assembler robot," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004. doi: 10.1109/IROS.2004.1389760 pp. 2349–2355.

[10] ——, "Automatic assembly system for modular structure," in *Proc. 22nd International Symposium on Automation and Robotics in Construction (ISARC)*, 2005. doi: 10.22260/ISARC2005/0028

[11] B. Jenett, A. Abdel-Rahman, K. Cheung, and N. Gershenfeld, "Material–Robot System for Assembly of Discrete Cellular Structures," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4019–4026, 2019. doi: 10.1109/LRA.2019.2930486

[12] B. Jenett and K. Cheung, "BILL-E: Robotic platform for locomotion and manipulation of lightweight space structures," in *25th AIAA/AHS Adaptive Structures Conference*, 2017, p. 1876.

[13] B. Bernus, G. Trinh, C. Gregg, O. Formoso, and K. Cheung, "Robotic specialization in autonomous robotic structural assembly," in *2020 IEEE Aerospace Conference*, 2020. doi: 10.1109/AERO47225.2020.9172620 pp. 1–10.

[14] Z. Abel, H. A. Akitaya, S. D. Kominers, M. Korman, and F. Stock, "A universal in-place reconfiguration algorithm for sliding cube-shaped robots in a quadratic number of moves," in *40th International Symposium on Computational Geometry (SoCG 2024)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 293, 2024. doi: 10.4230/LIPIcs.SoCG.2024.1. ISBN 978-3-95977-316-4. ISSN 1868-8969 pp. 1:1–1:14.

[15] I. Kostitsyna, T. Ophelders, I. Parada, T. Peters, W. Sonke, and B. Speckmann, "Optimal in-place compaction of sliding cubes," in *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 294, 2024. doi: 10.4230/LIPIcs.SWAT.2024.31. ISBN 978-3-95977-318-8. ISSN 1868-8969 pp. 31:1–31:14.

[16] C. Sung, J. Bern, J. Romanishin, and D. Rus, "Reconfiguration planning for pivoting cube modular robots," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1933–1940.

[17] D. Feshbach and C. Sung, "Reconfiguring non-convex holes in pivoting modular cube robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6701–6708, 2021. doi: 10.1109/LRA.2021.3095030

[18] J. Brunner, K. Cheung, E. Demaine, J. Diomidova, H. D. Christine Gregg, and I. Kostitsyna, "Reconfiguration algorithms for cubic modular robots with realistic movement constraints," in *Proc. 19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, 2024, pp. 34:1–34:18.

[19] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015. doi: https://doi.org/10.1016/j.artint.2014.11.006

[20] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[21] J.-M. Alkazzi and K. Okumura, "A comprehensive review on leveraging machine learning for multi-agent path finding," *IEEE Access*, vol. 12, pp. 57 390–57 409, 2024. doi: 10.1109/ACCESS.2024.3392305

[22] H. Ma, J. Li, T. S. Kumar, and S. Koenig, "Lifelong multi-agent path finding for online pickup and delivery tasks," in *Proc. 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2017. doi: 10.5555/3091125.3091243 pp. 837–845.

[23] F. Grenouilleau, W.-J. Van Hoeve, and J. N. Hooker, "A multi-label A* algorithm for multi-agent pathfinding," in *Proc. International Conference on Automated Planning and Scheduling (ICAPS)*, vol. 29, no. 1, 2019. doi: 10.1609/icaps.v29i1.3474 pp. 181–185.

[24] T. Peters, K. Cheung, and I. Kostitsyna, "Assembly order planning for modular structures by autonomous multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[25] I. Kostitsyna, J. Gloyd, and K. Cheung, "Multi-agent collective construction of general modular structures," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Stretch, But Don't Snap: Coordinating Multi-Agent Teams Under Competing Goals

Federico Oliva[1,2], Ido Sherf[1], and Amir Degani, *Member IEEE*[1,2]

***Abstract*— This work explores coordination strategies for heterogeneous multi-agent systems. The team comprises larger leader agents and smaller follower agents (children), each equipped with UWB antennas to measure relative distances within a limited range. We frame the coordination challenge as a trade-off between expansion and contraction: the former aims to cover the environment, while the latter maintains proximity for communication and task execution. These conflicting goals are modeled through a lightweight, decentralized optimization framework in which each agent balances local objectives with team-level robustness constraints.**

To analyze and guide this coordination, we define three key metrics: coverage (to assess spatial expansion), connectivity (based on edge cuts and inter-agent distances), and dispersion (using the Laplacian of the positioned graph). The trade-off between team and local goals is modeled through a convex combination of expansion and contraction tasks. Through both centralized and decentralized optimization scenarios, we show how tuning the dispersion constraint impacts the team's behavior, shaping feasible configurations that preserve connectivity while maximizing the current task.

I. Introduction

Recently, increasing interest has been drawn to collaborative robotics, specifically in heterogeneous multi-agent systems (MAS) [1]; these are characterized by agents differing in cognitive or physical properties, typically operating in unknown environments [2], [3]. In these scenarios, collaborative exploration becomes a fundamental task, mainly addressed through *Simultaneous Localization And Mapping* (SLAM) [4]–[7]. In our specific case study, we aim to use heterogeneous multi-agent teams to manipulate and shape aggregates for civil and environmental construction tasks. The team includes bulkier agents that act as leaders and smaller ones that follow as children.

In this work, we focus on a team of generic, localized agents operating in a shared environment. Each agent is assumed to be localized, and to have knowledge of its neighbours' position, enabling coordination through local interactions. The core challenge lies in managing competing tasks: on the one hand, agents are required to expand and explore the environment to maximize coverage; on the other, they must remain close enough to maintain connectivity for communication and collaborative task execution. This tension between expansion and contraction forms the basis for our coordination strategy [8], [9].

[1]Civil and Environmental Engineering, Technion - Israel Institute of Technology

[2]Technion Autonomous Systems Program, Technion - Israel Institute of Technology

This work was supported in part by the Jack Buncher Foundation.

II. Problem Statement

Consider a team of n agents on a 2D map; the team is divided into a leader A_L and the remaining children. Every agent has an *Ultra Wide Band* (UWB) range sensor. UWB is a low-cost range technology widely used in robotics and sensor networks for indoor and outdoor localization. In our case we define a sensing limit δ chosen according to the range of UWB antennas. Indeed, for a generic agent A_i provided with an UWB sensor, we can define its local ranging neighborhood $\mathcal{N}_{i,RD}$ as the subset of agents at most at a distance of δ from A_i, i.e.,

$$\mathcal{N}_{i,RD} \triangleq \{A_j \mid d_{ij} < \delta\}, \tag{1}$$

where RD stands for *Relative Distance*, and d_{ij} is the distance between A_i and A_j. Each agent is assumed to know its own position, which is shared within its neighborhood. Agents also have an operational area of radius δ_M, within which they can operate the environment. A graphical representation of the model is shown in Fig. 1, where the purple circle represents the team leader, the blue circles the children, the dashed lines the UWB measurements, and the grey circles the mapping sensors range (δ_M). In this work, we consider a noise-free scenario.

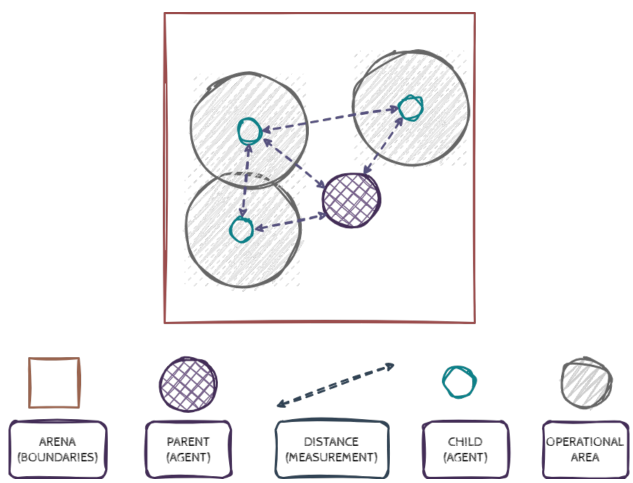


Fig. 1: Graphical representation of our case study.

To model this setup we start by defining a communication graph $\mathcal{G}$ within the network. If two agents (nodes) can measure their relative distance d_{ij}, then there exists an arc between them in the communication graph $\mathcal{G}$. The arc length is the Euclidean distance d_{ij} between two nodes (A_i, A_j). Thus, the communication graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}$ and by a set of edges $\mathcal{E}$. In fact, if $d_{ij} < \delta$, the edge $(i, j) \in \mathcal{E}$. For a team with agents' positions $\{\boldsymbol{p}_i\}$ and communication graph $\mathcal{G}$ defined by the ranging sensors, we define the related framework as the pair $(\mathcal{G}, \boldsymbol{p})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\boldsymbol{p}$ maps every node from $\mathcal{V}$ to a point $\boldsymbol{p}_i \in \mathbb{R}^2$ [10], [11].

A. Robustness

Robustness is essential for multi-agent teams operating in dynamic and uncertain environments, like construction sites, where communication can fail and tasks require adaptability. A robust team maintains connectivity, tolerates disturbances, and balances the need to spread out for task execution with the need to stay coordinated. In our case, agents must move outward to collect materials and return to deposit them, so robustness ensures they remain connected without collapsing into a single point or spreading too far apart to coordinate.

We define the robustness of a multi-agent team in terms of dispersion, a measure that captures how well agents are spatially distributed while maintaining network connectivity. This concept is formalized using the graph Laplacian, a matrix representation of the team's communication structure:

$$\mathcal{L} = \begin{bmatrix} \sum_{j\neq 1} w_{1j} & -w_{12} & \dots & -w_{1n} \\ -w_{21} & \sum_{j\neq 2} w_{2j} & \dots & -w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{n1} & \dots & -w_{nn-1} & \sum_{j\neq n} w_{nj} \end{bmatrix}, \quad (2)$$

where $w_{ij} = d_{ij}$ if $(i, j) \in \mathcal{E}$. It can be easily shown that

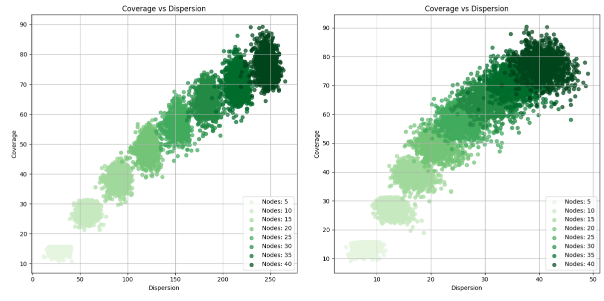
$$DISP = p^T \mathcal{L} p = \sum_{(i,j) \in \mathcal{E}} \| \mathrm{p}_i - \mathrm{p}_j \| . \quad (3)$$

B. Expansion

We introduce a coverage metric tailored to this task to quantify the team's ability to expand in the environment. Each agent operates within a circular area of fixed radius r, and we define coverage as the total area occupied by all agents, computed as the sum of their operational areas minus any overlapping regions. This metric directly reflects how well the agents are distributed in space rather than how connected they are. For example, agents may remain connected via UWB sensing without overlapping operational areas. Therefore, coverage is a meaningful indicator of spatial expansion, complementing connectivity-based measures:

$$COV = \frac{1}{A_{tot}} \left(\sum_{i=1}^{n} A_i - \sum_{(i,j) \in \mathcal{E}} A_i \cap A_j \right). \quad (4)$$

We analyze the relationship between coverage and dispersion by plotting the coverage metric against the dispersion value across 1,000 randomly generated teams, with team sizes ranging from 5 to 40 agents. We consider two scenarios by varying the UWB maximum sensing distance. Higher dispersion usually correlates with higher coverage, but when the UWB range is reduced, dispersion decreases while coverage stays similar, indicating that connectivity affects dispersion. In contrast, the coverage metric provides a more reliable measure of the team's expansion ability (Fig. 2).



(a) (b)

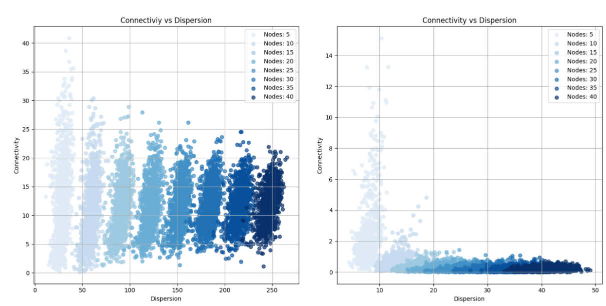
Fig. 2: (a) $\delta = 7$ (b) $\delta = 3$

C. Connectivity

We define our connectivity metric using the graph's edge connectivity, namely the minimum number of edges that need to be removed to disconnect the graph, scaled by a factor depending on the number of agents. This term is upper bounded by the graph's algebraic connectivity and reflects purely topological robustness, independent of agent positions. We multiply the metric by the total sum of distances between connected agents to make it continuous. While this introduces a minor dependence on spread, the influence is minimal, keeping the metric focused on connectivity [12]:

$$CON = 2 \cdot e(\mathcal{G}) \left[1 - \cos\left(\frac{\pi}{n}\right) \right] \cdot \sum_{(i,j) \in \mathcal{E}} d_{ij}. \quad (5)$$

We analyze the relationship between our connectivity metric and the dispersion by plotting the coverage metric against the dispersion value across 1,000 randomly generated teams, with team sizes ranging from 5 to 40 agents. We consider two scenarios by varying the UWB maximum sensing distance. Reducing the UWB range lowers connectivity and dispersion, but while connectivity saturates with more agents, dispersion continues to grow. This highlights their differing behaviors and the substantial impact of limited communication range (Fig. 3).

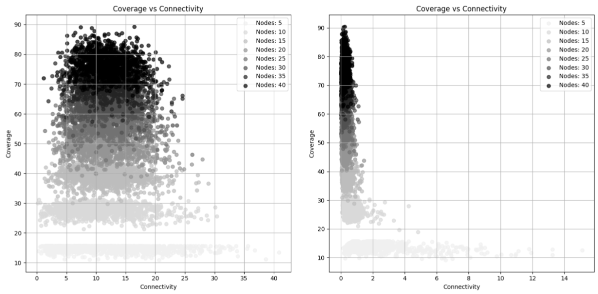


(a) (b)

Fig. 3: (a) $\delta = 7$ (b) $\delta = 3$

D. Interpretation

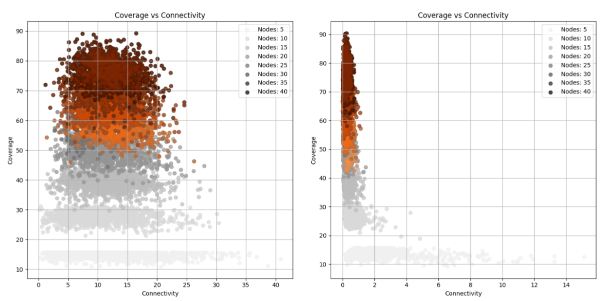
Plotting coverage against connectivity reveals an expected inverse relationship, highlighting the trade-off between expanding coverage and staying connected (Fig. 4).



(a) **(b)**

Fig. 4: (a) $\delta = 7$ (b) $\delta = 3$

Using dispersion as a robustness constraint narrows the feasible team set to those with sufficient coverage and limited connectivity. As shown in the same scenarios, this constraint filters out overly compact or over-expanded teams, aligning with previously observed trends between dispersion, coverage, and connectivity. For the sake of brevity, we only report the coverage against connectivity plots. In Fig. 5 we show the constraint $(DISP \geq 0.5DISP_{max})$.



(a) **(b)**

Fig. 5: (a) $\delta = 7$ (b) $\delta = 3$

III. Optimization

We now introduce our optimization problem, which balances expansion and contraction through a convex combination of both objectives. The goal is to maximize coverage while maintaining connectivity, with dispersion as a constraint to ensure robustness. To facilitate optimization, the coverage and connectivity metrics are normalized, ensuring that all terms in the cost function have the same order of magnitude and can be appropriately weighted during optimization:

$$\begin{aligned} \max_{p} \quad & \alpha \cdot CON(\mathcal{G}_p) + (1-\alpha) \cdot COV(\mathcal{G}_p) \\ \text{s.t.} \quad & DISP(\mathcal{G}_p) > DISP_{min} \\ & \mathcal{G}_p \text{ is connected} \end{aligned} \tag{6}$$

where α is a scalar used to determine if the current optimization task is more towards the expansion or contraction of the team.

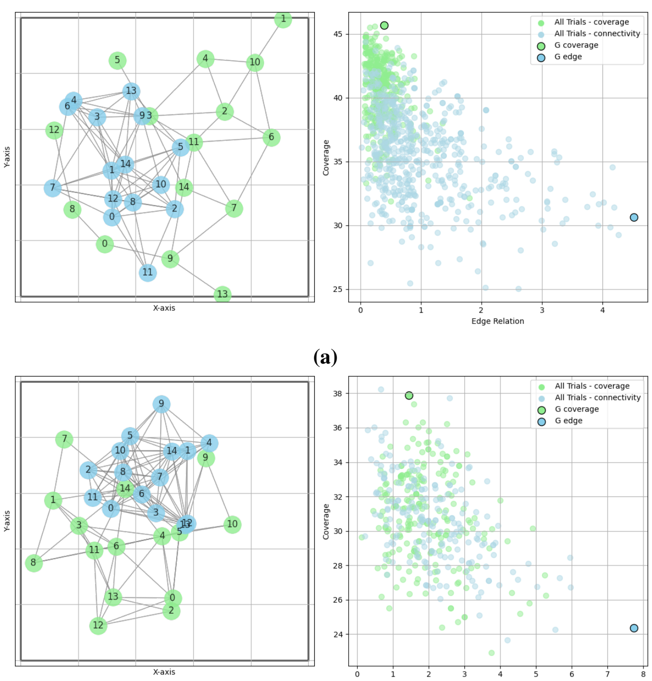
IV. Results

We present the results of two scenarios: a centralized optimization where all node positions are optimized together, and a decentralized scenario where each agent optimizes within its own neighborhood. All optimization problems will be solved numerically using the Optuna solver with the NSGAIISampler.

A. Centralized optimization

In the centralized scenario, we observe that varying the parameter alpha results in teams adapting their shape following the desired balance between expansion and contraction. As alpha is adjusted, the team transitions smoothly between prioritizing coverage and focusing on maintaining proximity. Furthermore, tightening the dispersion constraint leads to solutions that converge, as expected from the inverse relationship between coverage and connectivity. The results confirm that when the constraints are tightened, the teams become more similar, with reduced coverage and increased connectivity. This illustrates the expected behavior based on the coverage/connectivity trade-off.

In the simulation in Fig. 6, with 15 agents and a maximum distance $\delta = 3$, we performed 1000 iterations of the optimizer, where α values of 0 (green) and 1 (blue) represent the extremes of expansion and contraction, respectively.



(a)

(b)

Fig. 6: (a) $DISP \geq 10$ (b) $DISP \geq 15$

B. Decentralized optimization

In the decentralized optimization, agents locally adjust between contraction and expansion, and with a relaxed dispersion constraint ($DISP \geq 1$), the results align with the centralized case, emphasizing the coverage-connectivity trade-off. The optimization considers only the neighborhood of the agent, and hence the coverage and connectivity are rewritten as

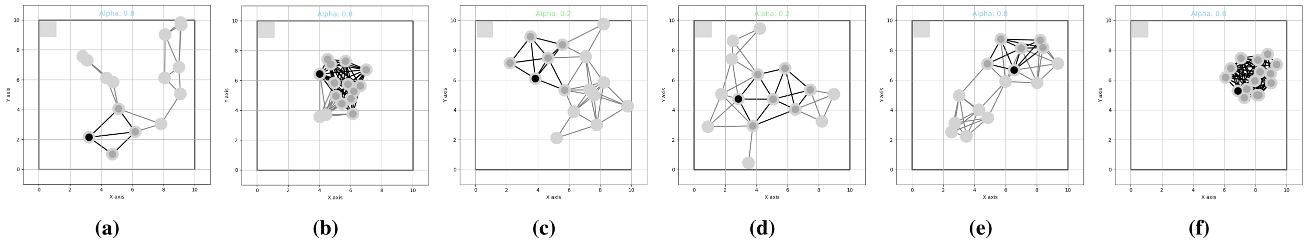


Fig. 7: (a) Cycle: 5 ($\alpha = 0.8$) (b) Cycle: 15 ($\alpha = 0.8$) (c) Cycle: 25 ($\alpha = 0.2$) (d) Cycle: 40 ($\alpha = 0.2$) (e) Cycle: 60 ($\alpha = 0.8$) (f) Cycle: 75 ($\alpha = 0.8$)

$$COV = \frac{1}{A_{tot}} \left(\sum_{i=1}^{|\mathcal{N}_i|} A_i - \sum_{(i,j) \in \mathcal{N}_i} A_i \cap A_j \right). \quad (7)$$

$$CON = 2 \cdot e(\mathcal{N}_i) \left[1 - \cos\left(\frac{\pi}{|\mathcal{N}_i|} \right) \right] \cdot \sum_{(i,j) \in \mathcal{N}_i} d_{ij}. \quad (8)$$

In this setup, the optimization runs for 10 iterations, with 80 cycles in total. During each cycle, all 15 agents are optimized together. The value of alpha changes periodically: for the first 20 cycles, alpha is set to 0.8 (favoring expansion), then for the next 40 cycles, alpha is set to 0.2 (favoring contraction), and finally, for the last 20 cycles, alpha returns to 0.8. The maximum communication distance ($\delta = 3$), and each agent operates within a 3x3 box, defining the area around itself where it searches for neighbors and adjusts its position. This setup allows the agents to dynamically switch between expansion and contraction tasks, testing how they adapt to changing objectives over time.

We present in Fig. 8 the coverage vs. connectivity plot, which clearly illustrates the inverse relationship between expansion and contraction, highlighting this dynamic even more effectively than in the centralized case.

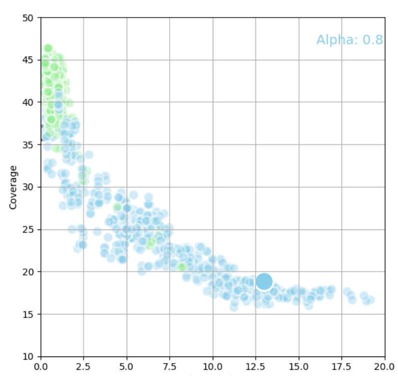


Fig. 8: Coverage vs Connectivity - points visited during the decentralized optimization.

We show in Fig. 7 a sequence of snapshots from the decentralized optimization across the entire cycles, illustrating the expansion and contraction of the team.

V. CONCLUSIONS

In conclusion, we have explored the trade-off between expansion and contraction in multi-agent systems, highlighting the role of coverage, connectivity, and dispersion in shaping team coordination. Through both centralized and decentralized optimization scenarios, we demonstrated how teams adapt their configurations based on task priorities and robustness constraints. The centralized approach provides a global perspective, while the decentralized formulation enables lightweight, distributed decision-making with consistent performance. Our results emphasize the effectiveness of dispersion as a robustness constraint and its impact on feasible team configurations.

Moving forward, we aim to leverage algebraic connectivity to enable analytical gradient-based optimization, further improving computational efficiency and scalability.

REFERENCES

[1] Yara Rizk, Mariette Awad, and Edward W. Tunstel. Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Computing Surveys*, 2019.

[2] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002.

[3] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, 2005.

[4] John B. Folkesson and Henrik I. Christensen. Robust SLAM. *IFAC Proceedings Volumes*, 2004.

[5] Ragesh K Ramachandran, Sean Wilson, and Spring Berman. A probabilistic topological approach to feature identification using a stochastic robotic swarm. In *Distributed Autonomous Robotic Systems: The 13th International Symposium*. Springer, 2018.

[6] Farid Bounini, Denis Gingras, Hervé Pollart, and Dominique Gruyer. From Simultaneous Localization and Mapping to Collaborative Localization for Intelligent Vehicles. *IEEE Intelligent Transportation Systems Magazine*, 2021.

[7] Sarah Brent, Chengzhi Yuan, and Paolo Stegagno. Swarm localization through cooperative landmark identification. In *Distributed Autonomous Robotic Systems: 15th International Symposium*. Springer, 2022.

[8] Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, 2013.

[9] Lijun Sun, Yijun Yang, Qiqi Duan, Yuhui Shi, Chao Lyu, Yu-Cheng Chang, Chin-Teng Lin, and Yang Shen. Multi-agent coordination across diverse applications: A survey, 2025.

[10] T. Eren, O.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In *IEEE INFOCOM 2004*, 2004.

[11] Daniel Zelazo, Antonio Franchi, Heinrich H. Bülthoff, and Paolo Robuffo Giordano. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *The International Journal of Robotics Research*, 2015.

[12] Russell Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 197-198:143–176, January 1994.