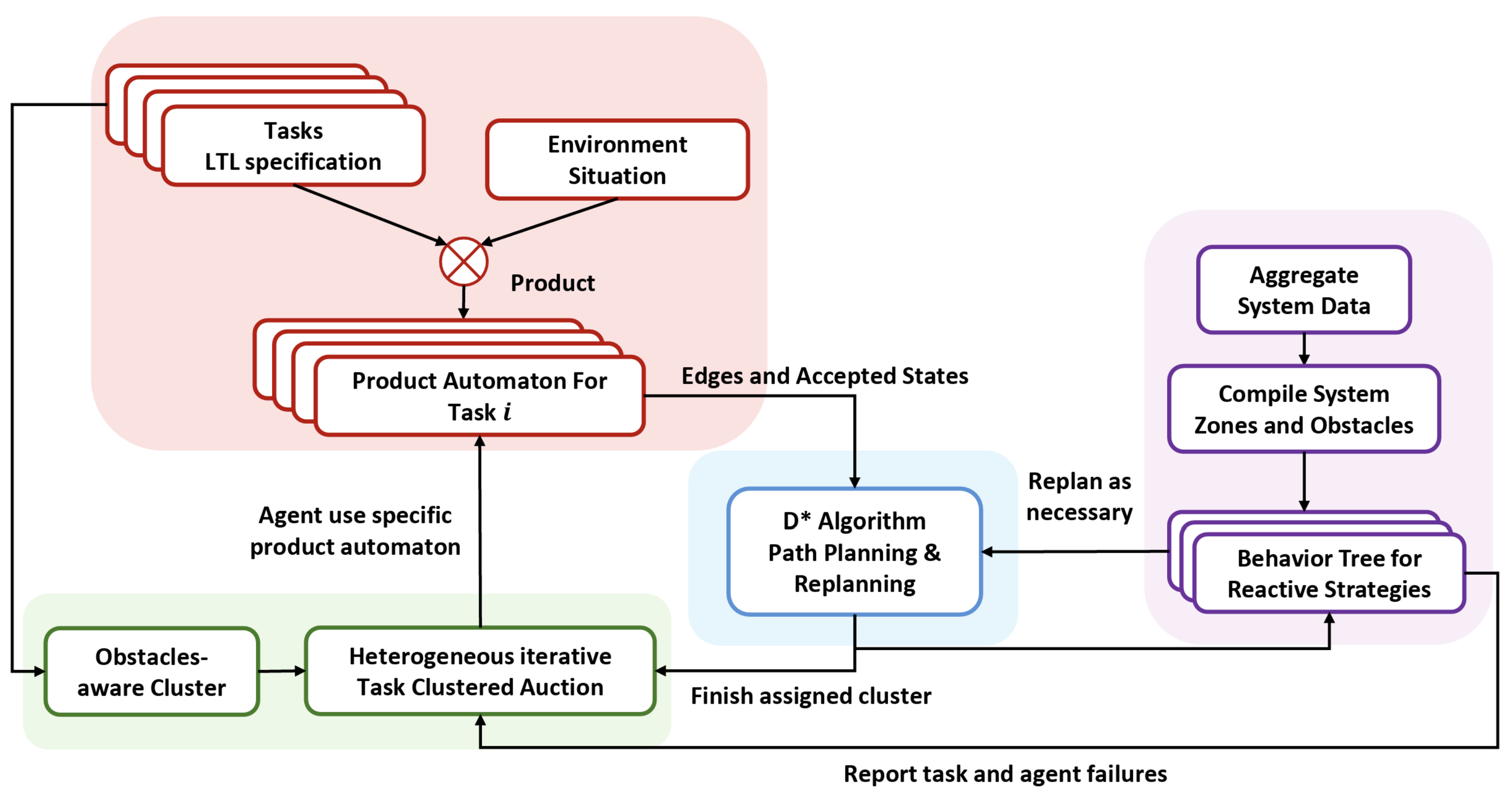
Resilient Task Allocation and Planning Framework for Heterogeneous Robot Teams

Nan Li, Haris Miller, Jiming Ren, Alagappan Swaminathan,
Samuel Coogan, Karen M. Feigh, and Ye Zhao

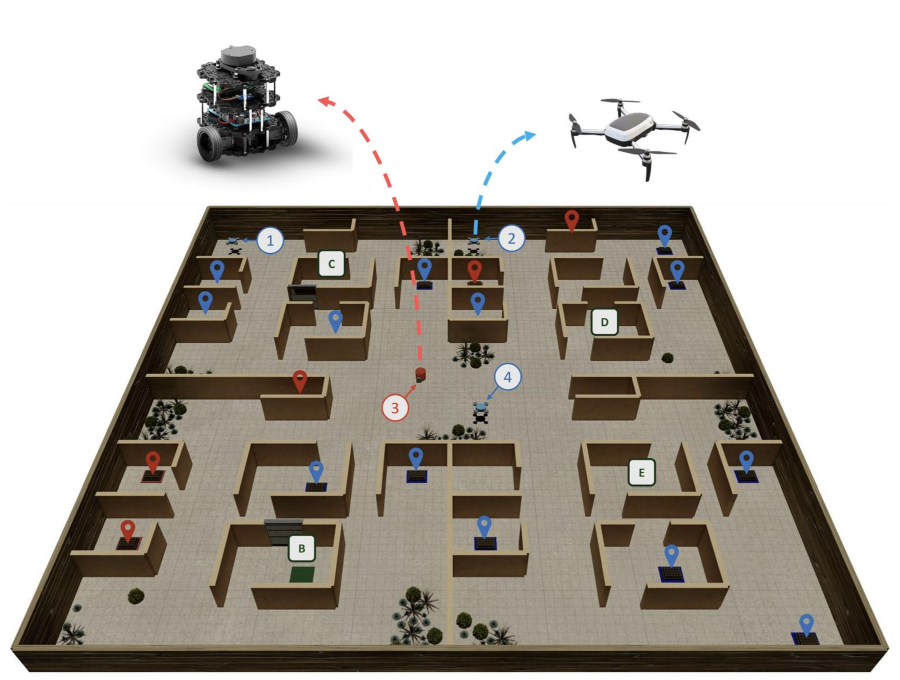
Objective

In real-world multi-robot systems, resilience and scalability are critical [1]. We propose a framework for **heterogeneous teams** that can **adapt to failures** and autonomously allocate tasks to aerial and ground robots, ensuring reliable execution under unexpected conditions.

Overall Framework



Simulation Environment



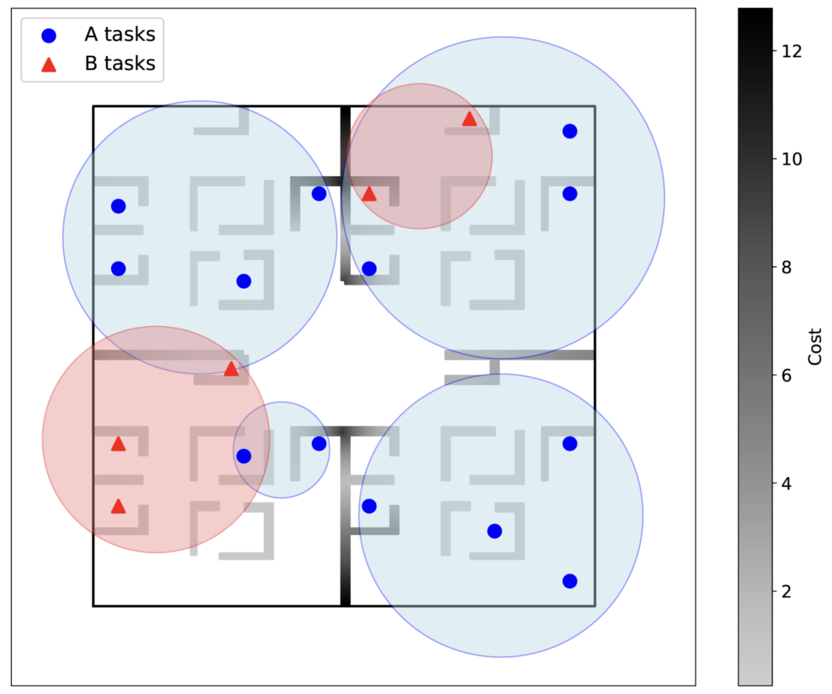
There are two types of agents: **drones and ground robots**. Similarly, there are two types of tasks, shown in the figure as blue and red. Ground robots can perform both types of tasks, whereas drones can only perform blue tasks.

Each task involves **picking up an item and delivering** it to a designated room labeled B, C, D, or E in the figure.

Multi-agent Task Allocation

1. Obstacle-aware Cluster

Obstacle-aware clustering uses a cost map with high wall costs to guide Agglomerative Clustering away from obstacle regions.



This figure illustrates the clustering of different task types. Blue tasks are grouped together into one cluster, and red tasks are grouped into another.

2. Consensus-based Auction Algorithm (CBAA)

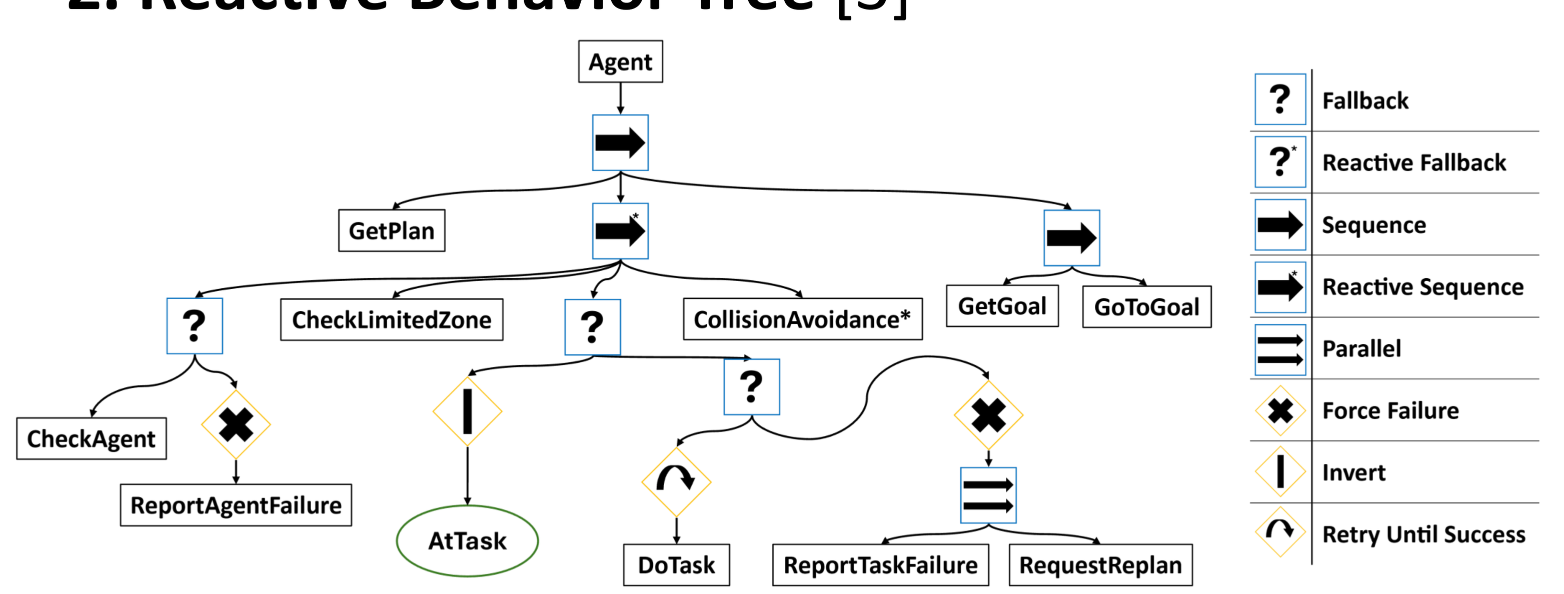
We use an iterative CBAA [2], a decentralized task allocation method in which agents bid on clusters and reach agreement through local communication.

Coordination Manager

1. System Compiler

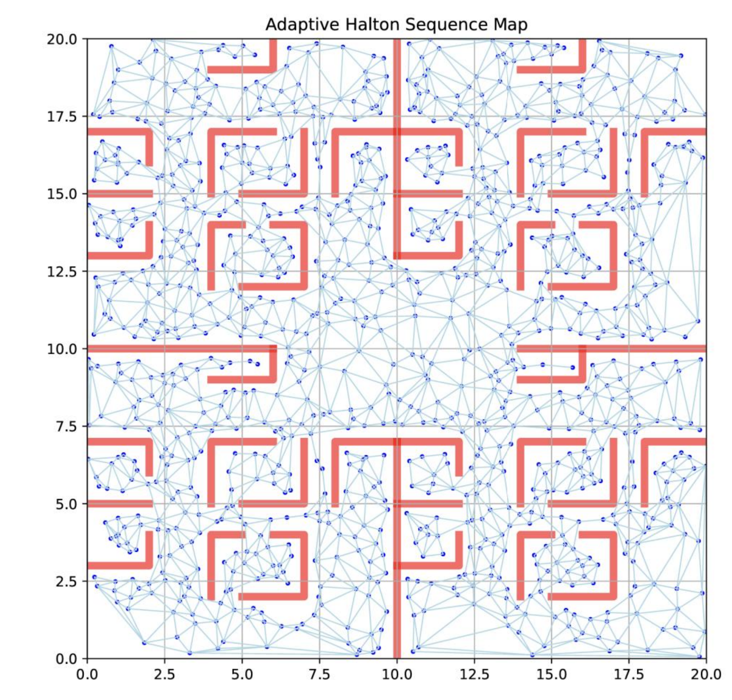
The system compiler aggregates all system information at execution such as agent states, environment states, system zones, etc. and shares it with other parts of the framework, as necessary.

2. Reactive Behavior Tree [3]



Dynamic Path Planning

1. **D* Lite Path Planning Algorithm**
   Heuristic search algorithm that efficiently replans paths in dynamic environments [4], [5].
2. **Adaptive Halton Sequence Map**



The Adaptive Halton Sequence adjusts point density based on the **distribution of obstacles**. It assigns lower point density in open areas and higher point density in regions with dense obstacles.

This approach keeps the total number of points low while capturing the complexity of the environment.

Preliminary Results

Metrics	10 Tasks	15 Tasks	20 Tasks	25 Tasks
Total Cost (steps)	199	497	687	868
Total Time (Seconds)	36.26	36.62	52.70	79.02

Table 1. The metrics above show that our planner scales effectively as the number of tasks increases. Note that the reported times exclude the construction of individual product automata, which is done before execution. For example, building all product automata for 18 tasks takes approximately 113 seconds.

The system can recover from unexpected scenarios, including agent failures and task failures.

Reference:

[1] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, “Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art,” Robot. Auton. Syst., vol. 168, p. 104492, Oct. 2023, doi: 10.1016/j.robot.2023.104492.

[2] Han-Lim Choi, L. Brunet, and J. P. How, “Consensus-Based Decentralized Auctions for Robust Task Allocation,” *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, Aug. 2009, doi: 10.1109/TRO.2009.2022423.

[3] M. Colledanchise, A. Marzinotto, D. V. Dimarogonas, and P. Ögren, “The Advantages of Using Behavior Trees in Mult-Robot Systems,” 2016.

[4] J. Ren, H. Miller, K. M. Feigh, S. Coogan, and Y. Zhao, “LTL-D*: Incrementally Optimal Replanning for Feasible and Infeasible Tasks in Linear Temporal Logic Specifications”.

[5] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” IEEE Trans. Robot., vol. 21, no. 3, pp. 354–363, Jun. 2005, doi: 10.1109/TRO.2004.838026.

nan.li@gatech.edu