WS 2025-2026, SMT-ROB, DISAL-SP203 Project Summary

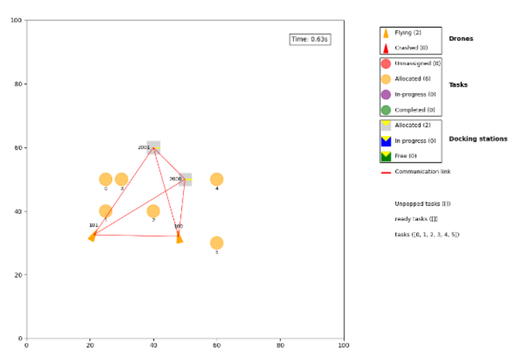
Market-Based Multi-Robot Task Allocation Algorithms for Micro Aerial Vehicles Performing Inspections

Francis Pannatier

Professor : Alcherio Martinoli
Assistant(s) : Kagan Erunsal

Robotics has gained increasing importance in recent years, with many applications requiring the coordination of multiple robots rather than a single agent. A key challenge in such systems is the assignment of tasks to robots, known as Multi-Robot Task Allocation (MRTA). This problem is NP-hard and computationally expensive to solve optimally. As a result, suboptimal but efficient approaches are often preferred. Market-based algorithms provide a scalable solution by modeling robots as self-interested agents that bid on tasks based on execution costs such as distance or energy consumption, enabling decentralized and efficient task allocation.

In this project, two market-based algorithms were implemented to address a MRTA problem with resource management constraints. The drones operate with limited battery capacity and can recharge at docking stations. Additional constraints were considered, including limited task visibility range, restricted communication distance between neighboring drones, and noise affecting the perceived positions of tasks.

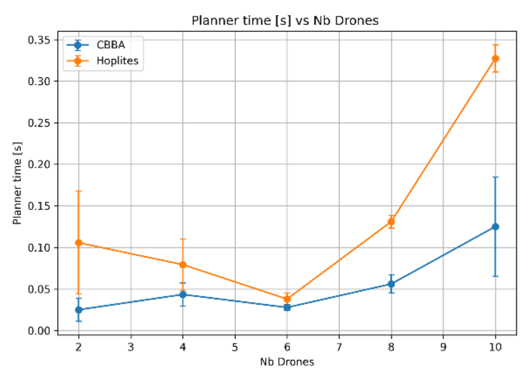


Simulation of a simple scenario

To address this problem, the Consensus-Based Bundle Algorithm (CBBA) and Hoplites were selected. The algorithms were evaluated in a two-dimensional Python-based simulation across multiple scenarios featuring different task distributions, task arrival rates, communication radii, and noise levels affecting both task locations and drone positions.

During implementation, CBBA showed reliable performance in simple scenarios but suffered from convergence issues in more complex settings with many drones and strict battery constraints. Despite extensive debugging, these issues could not be fully resolved, limiting direct comparisons with Hoplites to a reduced set of scenarios.

To manage the constraint of resources, docking stations were enforced at the end of each robot plan. For CBBA, each drone was addressed a default docking station who would be consistent across the all simulation where Hoplites gave the possibility to the drone to bid on them.



Comparison of planner computation time versus number of drones for CBBA and Hoplites

The results show that CBBA is mainly sensitive to the number of tasks, with computation time increasing significantly as task count grows, whereas Hoplites' computation time is more sensitive to the number of agents. Although Hoplites aims to locally optimize allocations through docking station assignment, it generally produces less optimal solutions in terms of task completion time and average travel distance, and leads to a higher number of drone crashes, possibly due to the docking station allocation strategy.