2nd Drone Routing Problem Competition : An Approach Based on Constrained Conflict-Based Search Algorithm

Atsushi Yanagisawa

Kyoto University

May 20, 2025

Table of Contents

Table of Contents

Introduction(Problem Setting)

Objective

For each step, create an algorithm that moves along **the shortest path possible without collisions**, based on information about the agent's current location and goal, as well as information from the map.

Table of Contents

Priority-based A* Routing for Multi-Agent Path Planning

- **1:Compute Shortest Paths:**
  - For each agent, run A* from current node to goal on the full graph.
  - Record path and total cost (sum of edge weights).
- **2:Assign Priority:**
  - Sort agents by ascending path cost (closer agents go first).
- 3: For each agent(in priority order):
  - **3-1:Reserve Resources:**
    - ★ In priority order, pick each agent ' s next move.
    - ★ Mark chosen nodes/edges as " occupied " to prevent conflicts.
  - **3-2:Collision-Aware Replanning:**
    - ★ Re-run A* to find an alternative collision-free path. (Avoid the occupied nodes/edges)
    - ★ If no path exists, the agent waits.

Table of Contents

Conflict-Based Search (CBS) Overview

- **High-Level Search:**manages conflicts between single-agent paths
  - Evaluate a set of paths from low-level for any conflicts.
  - If no conflict, return solution.
  - If conflict exists, add constraint to avoid it and request new low-level paths.
- **Low-Level Search:**computes individual paths under given constraints.
  - Run A* for one agent, obeying constraints imposed by high-level Search.
  - Output a candidate path for conflict checking.

The problem is that the computational cost is high(often failed to find path).

Constrained Conflict-Based Search(Proposed Algorithm)

- **Initial Constraints:**
  - Reserve nodes of agents already at goal.
  - Reserve edges currently occupied by other agents.
- Considering only the agents in nodes
- **1: Run CBS:**
  - If a conflict-free solution is found within time limit,
  - return each agent ' s first step.
- **2: Constrained CBS(if CBS failed to find path in the limited-time):**
  - Sort agents by distance to their goals, descending.
  - For each subset of agents (starting with the farthest):
    - ★ Create new constraints to " freeze " their current nodes.
    - ★ Rerun CBS with these constraints.
    - ★ If a solution is found, return each agent ' s first step.
  - If no solution is found after all subsets, fallback to prioritized A*.

Table of Contents

Results:

Using three different maps, a total of 30 different situations (number and placement of agents) each, the distance (cost) that an agent moved was calculated.

Algorithm	**Cost**
Random Algorithm	18,158
Simple A* Algorithm	16,187
Prioritized A* Algorithm	13,825
Constrained CBS Algorithm	**10,903**

Table: Performance Comparison of Different Routing Algorithms

Table of Contents

Constrained Conflict-Based Search: Discussion & Future Work

Discussion & Limitations

- Constrained CBS outperformed other algorithms in terms of cost.
- High Computational Cost
- Only about half of the agents move in each iteration; many must wait.

Future Work

- **Enhanced CBS variants**: Explore EECBS, ICBS, CBS with Deadlines for reduced cost.

Table of Contents

Conclusion

Conclusion

- Constrained Conflict-Based Search (CBS) effectively manages multi-agent path planning.
- It outperforms simpler algorithms like A* and prioritized A* in terms of cost.

Thank you for your attention!