**Proposal for Integrating Conflict-Based Search (CBS) into the Multi-Agent Pathfinding (MAPF) Module**

**1. Introduction**  
Our multi-agent logistics system requires robust path planning to move multiple agents through a warehouse without collisions. Current solutions such as lane-based, inash, or sampling-based planners can be effective in certain constrained or continuous settings. However, in a structured grid-like warehouse environment, we propose employing **Conflict-Based Search (CBS)** to achieve efficient, near-optimal multi-agent pathfinding.

**2. Overview of CBS**  
CBS addresses MAPF using a two-level approach:

1. **High-Level (Conflict Resolution):** Maintains a Constraint Tree (CT). When two agents collide, CBS branches into two child nodes, each enforcing a new constraint against one of the colliding agents. This systematically eliminates collisions.
2. **Low-Level (Single-Agent Pathfinding):** Within each CT node, agents plan individual paths subject to their constraints, typically using A\* or similar graph-based algorithms.

This hierarchical structure provides a strong balance of performance and solution quality. An enhanced version, **ECBS** (Enhanced CBS), allows bounded suboptimality (e.g., solutions within 1.1× the optimal cost) to further reduce computation time.

1. **Implementation Detail**
   1. **Collision avoidance:**

To account for the obstacle collision, we can add an extra boundary around the obstacles to prevent the algorithm from bumping into them. And to account for the agent collision, we can add constraints to maintain the distance between each agent.

* 1. Reassign Agent’s path:

When a agent reached a goal and is assigned with a new path, simulation will pause and wait for a new solution respecting the constraints.

**4. Proposed Integration Steps**

1. **Create a CBS Module as a Global Planner class:**
   * *Constraint Tree Manager:* Tracks constraints generated by collisions.
   * *Agent-Specific Planner:* Individual A\* (or D\* Lite) to compute paths under constraints.
   * *Collision Detector:* Identifies conflicts in the current set of paths.
2. **Iterative Search and Conflict Branching:** Continue expanding the Constraint Tree until no collisions remain or until a user-defined time/solution bound is reached.
3. **Execute the solution with Local Planner:** Use a simple local planner to move agent from point to point. Since we have ensure the passes from the global planner, we don’t need a complicated logic for the local planner.

**5. Benefits**

* **Better Scalability:** Focuses on actual collisions rather than exhaustively exploring continuous or lane pathways.
* **Flexible Constraint Handling:** Easily accommodates new rules (e.g., restricted zones, agent priorities).
* **Near-Optimal Solutions:** Yields provably valid paths, with adjustable trade-offs for speed vs. quality.

By integrating CBS into our current project structure, we can significantly improve the efficiency and robustness of our multi-agent pathfinding, ensuring collision-free navigation in a dynamic warehouse environment.

Project Timeline:

February code structure understanding and algorithm research

March algorithm proposal and start of implementation

April first demo and refining, start of project report

May submission