Conflict Resolution Case Study for Fleet Management System (FMS)

Scenario Overview

In a warehouse environment, a Fleet Management System (FMS) oversees a group of mobile robots tasked with navigating to specific locations, picking items, and delivering them to designated areas. Conflicts arise when:

- Two robots are scheduled to access the same aisle or node at the same time.
- One robot delays task completion, disrupting the schedule for other robots.

Objective:

Design a conflict resolution strategy to determine which robot gains access to a common path (aisle) while considering factors such as task urgency, proximity, and overall operational efficiency.

Proposed Conflict Resolution Strategy

Factors to Consider:

1. Task Urgency (Priority):
   - Tasks with higher urgency (e.g., time-sensitive deliveries) are given higher priority.
2. Proximity to the Conflict Zone:
   - The robot closer to the shared aisle is given priority to minimize delays.
3. Battery Levels:
   - Robots with lower battery levels are prioritized to prevent delays caused by battery depletion.
4. Operational Efficiency (System-Level View):
   - The system evaluates the overall impact of a delay on all tasks, aiming to minimize global disruptions.

Conflict Resolution Algorithm

1. Input Data:

   - Robot tasks, priorities, proximity to the conflict zone, and battery levels.

2. Decision Criteria (Weighted Factors):

   - Assign weights to factors based on their importance.

3. Priority Scoring:

   - Calculate a score for each robot based on the weighted factors.

4. Decision:

   - The robot with the higher score gains access to the aisle.

5. Adaptation for Scalability:

   - For larger fleets, group robots into clusters and resolve conflicts within each cluster to reduce computational overhead.

Implementation Example

Scenario:

- Robot A: Task urgency is High, Proximity is 2 meters, Battery is 50%.
- Robot B: Task urgency is Medium, Proximity is 4 meters, Battery is 80%.

Decision:

- Robot A gains access based on calculated priority score.

Trade-Offs

1. Proximity vs. Urgency:

   - Prioritizing proximity reduces immediate delays but may compromise time-sensitive tasks.

2. Scalability:

   - As the number of robots increases, the algorithm must balance local and global priorities.

3. Battery Consideration:

   - Over-prioritizing low-battery robots may lead to inefficiencies.

Scalability Solutions

1. Clustering:

   - Divide robots into smaller groups based on zones, resolving conflicts locally.

2. Distributed Decision-Making:

   - Equip robots with decentralized conflict resolution protocols for real-time decisions.

3. Dynamic Weight Adjustments:

   - Adjust factor weights based on the warehouse's operational state (e.g., peak hours).

Conclusion

The proposed conflict resolution strategy ensures:

- Minimized Delays by prioritizing task urgency and proximity.

- Operational Efficiency through consideration of global impacts.

- Scalability via clustering and dynamic adjustments.