Multi-agent Motion Planning for Non-Holonomic Mobile Robots

Prasanna Natu
MS Robotics Engineering
pvnatu@wpi.edu

Shreya Bang
MS Robotics Engineering
srbang@wpi.edu

Shambhuraj Mane
MS Robotics Engineering
samane@wpi.edu

***Abstract*—Navigating the complexities of motion planning for multiple non-holonomic mobile robots presents challenges, especially with traditional. The aim of this project is to put forth a motion planning strategy capable of producing safe and dynamically feasible trajectories for multiple non-holonomic mobile robots. It offers a solution using a priority based approach with the Improved Conflict-Based Search (ICBS) for multi-robot planning which obstacle avoidance.**

I. Introduction

The modern landscape of warehousing and logistics has witnessed a paradigm shift with the integration of automation and robotics. The demand for efficient, precise, and agile operations within warehouses has escalated, driven by the growing e-commerce industry and consumer expectations for rapid order fulfillment. As a result, there is an increasing need for advanced planning algorithms that can demonstrate the movements of multiple agents within constrained spaces, ensuring not only operational efficiency but also safety and collision avoidance.

This project delves into the critical realm of multi-agent path planning of Non-holonomic Mobile robots. The objective is to design and implement ICBS (Improved Conflict-Based Search) that enables multiple agents to navigate environments, coordinating their movements seamlessly while avoiding collisions. The primary application focus is on pick and place operations within a warehouse setting, a domain where precision, speed, and safety are of paramount importance.

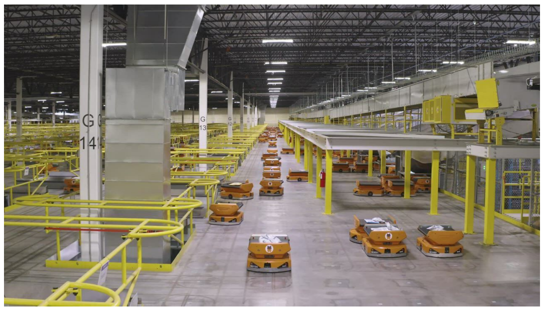


Fig. 1. Multi-Agent Robots

II. Literature Review

The field of Multi-agent Path Finding (MAPF) has attracted significant attention due to its varied applications, ranging from warehouse management and airport towing to shopping centers. Central to MAPF is the challenge of devising strategies that ensure multiple robots can navigate without collisions. A study by A. Bolu et al. [1] offers an intriguing solution for mobile robots navigating warehouse grids by proposing a variant of the A* algorithm, which incorporates turning costs.

On another note, G. Sharon et al. [2] present the Conflict Based Search (CBS), a two-tiered algorithm that notably avoids simplifying the multi-agent problem to single-agent models. Instead, it embarks on a unique path search journey via a conflict tree (CT), focusing on agent-specific conflicts. The low-level finds optimal paths for the individual agents. If the paths include conflicts, the high level, via a split action, imposes constraints on the conflicting agents to avoid these conflicts.

It is worth noting that at each level of the CT, standard CBS randomly chooses a conflict to split, which may result in poor choices that significantly impact algorithm performance. ICBS: Improved Conflict-Based Search Algorithm [3] was introduced to improve these choices by four extensions to standard CBS. All four enhancements to CBS (i.e., Meta-agent, Merge & Restart, Prioritizing Conflicts, and Bypassing Conflicts) are optional and can be added separately or in conjunction with the others, except for MR which is only relevant to MA-CBS.

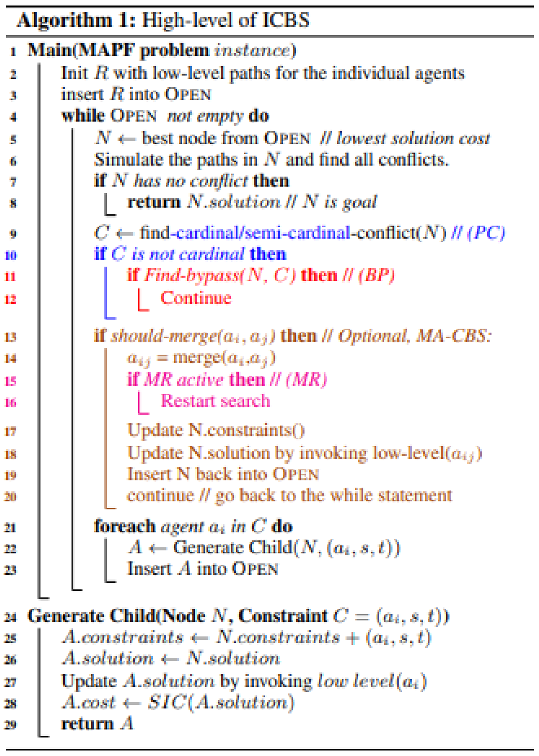
III. Proposed Solution

We will be implementing the priority based CBS version for the multi-agent path plannning project. The prioritization of conflicts in the CBS algorithm is an improvement called ”Prioritizing Conflicts” (PC). In the basic form of CBS, conflicts are arbitrarily chosen to be split, which can lead to poor performance and a large search tree. The PC improvement aims to address this drawback by prioritizing conflicts.

By prioritizing conflicts, CBS can immediately increase the solution cost in the subtree below the current node, reducing the size of the search tree. This improvement helps to improve the performance of the CBS algorithm.

ICBS arbitrarily chooses conflicts to split on. However, poor choices can substantially increase the size of its CT and thus its runtime. Improved CBS (ICBS) [Boyarski et al., 2015] addresses this issue by prioritizing conflicts.

We will be employing a 2D simulation environment using ARGoS, which will allow us to simulate the motion planning strategy for multiple robots within this warehouse scenario.



```
Algorithm 1: High-level of ICBS
Main(MAPF problem $instance$)
    Init $R$ with low-level paths for the individual agents
    insert $R$ into OPEN
    while OPEN not empty do
        $N \leftarrow$ best node from OPEN // lowest solution cost
        Simulate the paths in $N$ and find all conflicts.
        if $N$ has no conflict then
            return $N.solution$ // $N$ is goal
        $C \leftarrow$ find-cardinal/semi-cardinal-conflict($N$) // (PC)
        if $C$ is not cardinal then
            if Find-bypass($N$, $C$) then // (BP)
                Continue
        if should-merge($a_i, a_j$) then // Optional, MA-CBS:
            $a_{ij}$ = merge($a_i$,$a_j$)
            if MR active then // (MR)
                Restart search
            Update N.constraints()
            Update N.solution by invoking low-level($a_{ij}$)
            Insert N back into OPEN
            continue // go back to the while statement
        foreach agent $a_i$ in $C$ do
            $A \leftarrow$ Generate Child($N$, $(a_i, s, t)$)
            Insert $A$ into OPEN
Generate Child(Node $N$, Constraint $C = (a_i, s, t)$)
    $A.constraints \leftarrow N.constraints + (a_i, s, t)$
    $A.solution \leftarrow N.solution$
    Update $A.solution$ by invoking $low\ level(a_i)$
    $A.cost \leftarrow SIC(A.solution)$
    return $A$
```

Fig. 2. High-level of ICBS

IV. Methodology

A. Environment Setup

We have achieved a successful setup for a 2D and 3D obstacle field utilizing Matplotlib and ARGoS respectively. Environment setup is done in two stages :

- Stage 1: For first stage, holonomic environment is setup using matplotlib to work on MAPF for grid based holonomic agents. The same environment is then used for nonholonomic initial study and analysis. The environment is shown in Fig.03
- Stage 2: For second stage, 30X30 grid is setup using matplotlib to work on MAPF for non-holonomic agents and for 3D simulation third environment is an ARGoS 2D dynamic physic engine setup for representing the non-holonomic agents with kino-dynamic constraints using in-built ARGoS bots.

B. Implementation

For our Multi-Agent Path finding Problem, we have chosen to dive into the ICBS algorithm, which stands for Improved Conflict-Based Search. This approach builds upon the classic

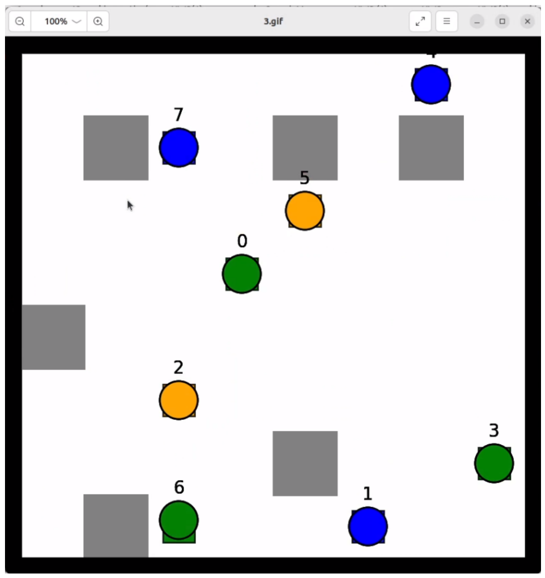


Fig. 3. Stage1: 2D environment

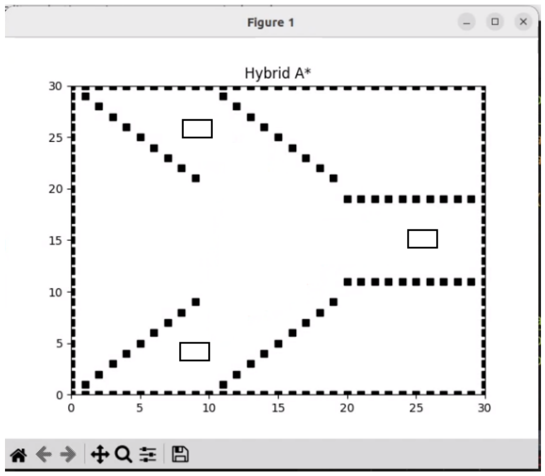


Fig. 4. Stage 2: 2D environment

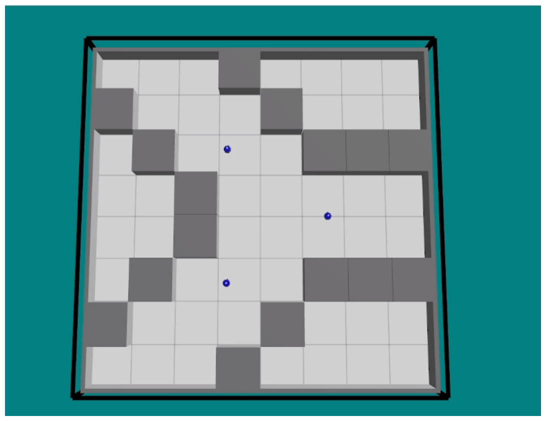


Fig. 5. Stage 3: ARGoS environment

CBS method by incorporating four key strategies: Prioritizing Conflicts, Bypassing Conflicts, Meta-agent handling, and Merge and Restart.

We have implemented the algorithm which employs prioritized conflict resolution and disjoint splitting to enhance the path planning process. The algorithm utilizes prioritized conflict resolution to handle situations where conflicts arise among multiple robots. The prioritization is based on a heuristic that considers the Euclidean distance, guiding the algorithm to resolve conflicts efficiently. A key feature of the algorithm is disjoint splitting, especially beneficial when two agents conflict at a specific point. By excluding that point from their individual path planning, robots avoid unnecessary collisions and optimize their trajectories. Three stages of implementation address various constraints for point robots with holonomic constraints, point robots with non-holonomic constraints, and differential drive robots with non-holonomic constraints.

In ICBS, the low-level algorithm is responsible for finding collision-free paths for individual agents while taking into account the current configuration of the environment. We have employed the A* algorithm which will be acting as the low-level algorithm for ICBS in order to address conflicts in the path of the agents. However for non holonomic constraints, we have shifted to hybrid A* implementation.

Stage 1: Point Robot with Holonomic Constraints: For the initial stage, the algorithm is applied to point robots with holonomic constraints. In this setting, the robots can move freely in any direction without any restrictions. The algorithm considers static obstacles in the environment and dynamically moving robots as potential obstacles. The prioritized heuristic, based on Euclidean distance, guides the path planning process.

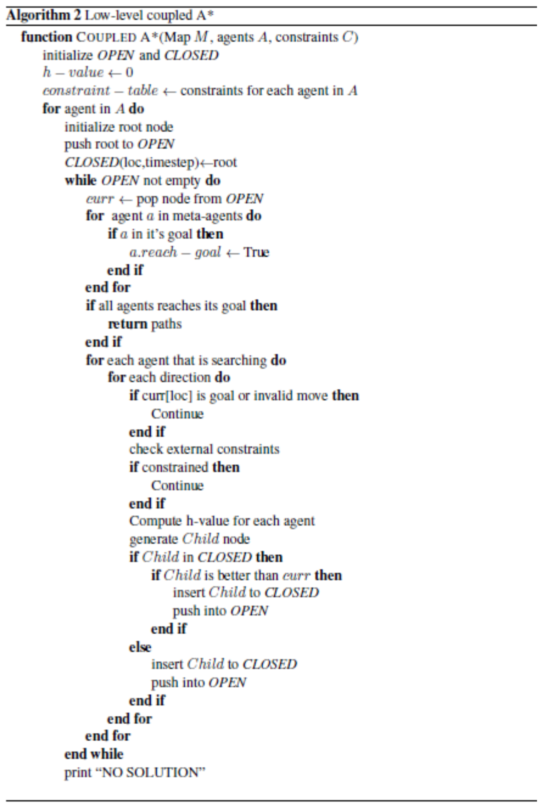
Stage 2: Point Robot with Non-Holonomic Constraints: Moving forward, the algorithm is extended to accommodate point robots with non-holonomic constraints. Non-holonomic constraints restrict the robot's motion, requiring careful consideration in path planning. Despite these constraints, the algorithm efficiently navigates the robots around static obstacles and dynamically moving agents.

Stage 3: Differential Drive Robot with Non-Holonomic Constraints: In the final stage, the algorithm is adapted to address robots with differential drive mechanisms and non-holonomic constraints. Differential drive robots have specific constraints on their motion, and the algorithm optimizes their paths while avoiding static obstacles and dynamically moving robots.

V. Challenges

Navigating Challenges in Holonomic Point Robots

1) **Algorithm Scalability:** As the number of robots increased, the scalability of the ICBS algorithm became a concern. The combinatorial nature of conflict resolution and coordination among multiple robots added computational overhead, affecting the real-time performance of the system. Scaling the algorithm to handle a larger number of robots while maintaining efficiency became a crucial aspect that required attention.



```
Algorithm 2 Low-level coupled A*
  function COUPLED A*(Map $M$, agents $A$, constraints $C$)
      initialize OPEN and CLOSED
      $h - value \leftarrow 0$
      $constraint - table \leftarrow$ constraints for each agent in $A$
      for agent in $A$ do
          initialize root node
          push root to OPEN
          CLOSED(loc,timestep)←root
          while OPEN not empty do
              $curr \leftarrow$ pop node from OPEN
              for  agent $a$ in meta-agents do
                  if $a$ in it's goal then
                      $a.reach - goal \leftarrow$ True
                  end if
              end for
              if all agents reaches its goal then
                  return paths
              end if
              for each agent that is searching do
                  for each direction do
                      if curr[loc] is goal or invalid move then
                          Continue
                      end if
                      check external constraints
                      if constrained then
                          Continue
                      end if
                      Compute h-value for each agent
                      generate $Child$ node
                      if $Child$ in CLOSED then
                          if $Child$ is better than $curr$ then
                              insert $Child$ to CLOSED
                              push into OPEN
                          end if
                      else
                          insert $Child$ to CLOSED
                          push into OPEN
                      end if
                  end for
              end for
          end while
          print "NO SOLUTION"
```

Fig. 6. Low-level Coupled A*

2) **Limited Collision Consideration in Diagonal Movements:** One of the primary challenges encountered during the implementation of the ICBS algorithm with an A* low-level planner for point robots was the limited collision consideration during diagonal movements. The choice of a step size of 0.1 for achieving smoother and more continuous movement in the environment inadvertently led to an oversight in collision detection when two robots were crossing each other diagonally. As a result, the robots did not adequately account for collisions and executed direct swaps in grid cells, compromising the overall safety of the system.

Continuous Movement for Non-Holonomic Point Robots

1) **Partial Robot Exploration near Obstacles:** The implementation of a non-holonomic point robot introduced a unique challenge where, during exploration, it was observed that the robot's representation as a point failed to adequately account for the physical constraints. In illustrations, it became evident that the half of the robot's virtual presence was extending into obstacles during close proximity exploration. This limitation highlighted the necessity for a more accurate representation of the robot's geometry to ensure collision-aware planning.

2) **Continuous Movement Oscillations and Convergence Issues:** While striving for continuous movement with a step size of 0.1, challenges arose with the tuning

of speed and omega hyperparameters. Some parameter combinations led to undesirable behavior, where the robot, especially when approaching the goal, exhibited oscillations and slow convergence. This behavior was practically incorrect and required careful tuning to strike a balance between continuous movement and stable convergence to the goal.

Differential Drive Non-Holonomic Robots

1) **Geometry-aware Collision Avoidance:** Transitioning to a differential drive robot with non-holonomic constraints addressed the challenges associated with point and non-holonomic point robot representations. By introducing a bounding box around the robot, the implementation became more geometry-aware. This bounding box facilitated effective collision avoidance, ensuring that the robot maintained safe distances from obstacles in the environment. The incorporation of this feature marked a crucial advancement towards practical and safe trajectory planning.

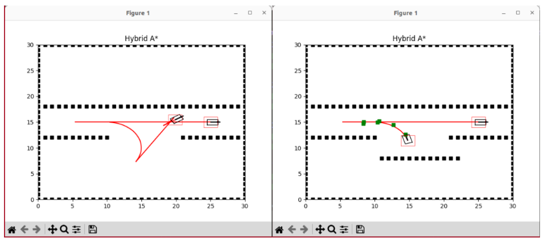


Fig. 7. Edge Case: a) The planned is path by giving agent 1 space to reach the goal, b) Path is not found considering the constraints of obstacle and conflicts.

VI. Application

Collision-free path planning in a multi-agent robot system is crucial for ensuring the safety, efficiency, and effectiveness of operations. It prevents collisions, safeguarding both the robots and their surroundings. It facilitates coordinated movement in scenarios where multiple robots work together, allowing them to complement each other's actions. Multi-agent robot scenarios without a grid-based environment are studied in this project to address dynamic and complex settings, such as:

- Real-world Navigation: Navigating through a crowded and unpredictable environment like a busy street or a shopping mall where the terrain is uneven.
- Real-world Navigation: Navigating through a crowded and unpredictable environment like a busy street or a shopping mall where the terrain is uneven.
- Search and Rescue: In disaster-stricken areas with irregular structures and obstacles, where agents need to collaborate to locate and rescue survivors.
- Underwater Exploration: Exploring the ocean floor with multiple autonomous underwater vehicles, dealing with currents, varying depths, and unknown terrain.
- Precision Agriculture: Multiple robotic agents working in agricultural fields with irregular crop layouts, varying soil conditions, and unexpected obstacles.
- Warehouse Logistics: Autonomous robots collaborating in a warehouse with changing inventory, dynamic obstacles, and diverse item shapes and sizes.

In these scenarios, agents need to plan the path considering the constraints to adapt to the complex nature of the environment.

VII. Result

Results are analyzed for two types of agents, holonomic and non-holonomic considering key points in motion planning performance check as follows:

- Path Optimality:
  - Paths found for holonomic agents were optimum paths for given start and goal locations after resolving conflicts.
  - Paths found for non-holonomic agents were sub-optimum paths for given start and goal locations after resolving conflicts.
- Conflict resolving:
  - Paths found for holonomic as well as non-holonomic agents were without conflicts for all the cases with 100% success.
- Completeness:
  - Paths found for holonomic agents guarantee finding a solution if one exist.
  - Paths found for non-holonomic agents doesnot guarantee finding a solution.

VIII. Conclusion

The improvised Conflict Based Search is a promising approach to solve the path planning approach effectively however it is not completely robust to environment. It can become computationally heavy with each increase in agent count which would make it unrealistic to operate.

IX. Future Scope

In the realm of future developments, our project aspires to achieve a significant stride by enabling each mobile robot to autonomously navigate towards dynamically defined pick-up locations, all while safely interacting with fellow robots functioning as dynamic obstacles. This ambitious goal adds layers of complexity, necessitating the design and implementation of sophisticated algorithms for seamless collaboration among multiple robotic entities.

Furthermore, we envision selection of pick-up locations to be intelligently determined based on the spatial configuration of robots and assigned tasks. This task-driven approach aims to optimize system efficiency by strategically selecting pick-up points, minimizing travel distances, and maximizing the utilization of each robot's capabilities.

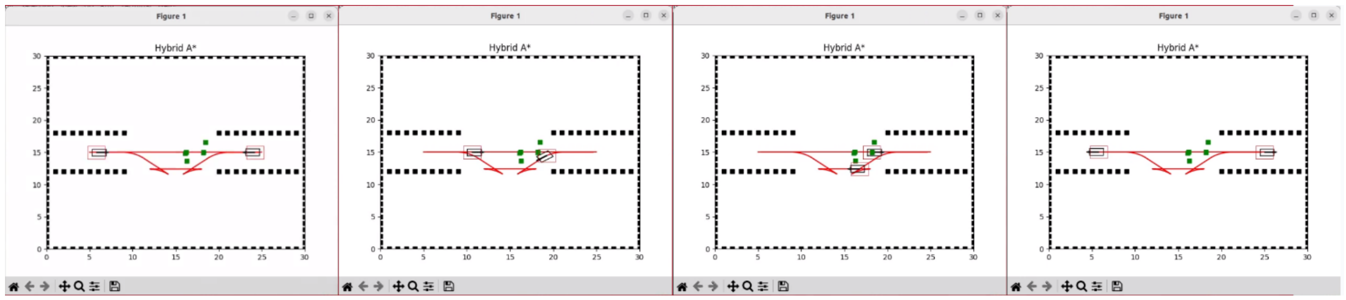


Fig. 8. Stages: a)Agents at initial position, b) Agent 1(left) is prioritized so agent 2 changed the path, c) Agent 2 planned a parallel optimum path for goal, d) Agents at goal position, (Green dots shown illustrate the conflict locations resolved/ avoided while planning path

Timeline	Task	Task allocation
Week 1 & 2 **(DONE)**	Literature Review	Shambhuraj, Prasanna, Shreya
Week 3 & 4 **(DONE)**	Simulation setup and Protoyping algorithms	Shambhuraj, Prasanna, Shreya
Week 5 & 6 **(DONE)**	Simulation and Algorithm writing	Shreya and Prasanna
Week 7 & 8 **(DONE)**	Integrating algorithm with environment	Shreya and Shambhuraj
Week 9 & 10 **(DONE)**	Debugging and Testing	Shambhuraj and Prasanna
Week 11 & 12 **(DONE)**	Validating, Buffer Time & Stretch Goal	Shreya Prasanna
Week 13 & 14 **(DONE)**	Stretch goals and Documentation	Shreya Prasanna Shambhuraj

X. Schedule and Division of Labour

References

[1] A. Bolu and Ö. Korçak, ”Path Planning for Multiple Mobile Robots in Smart Warehouse,” in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, 2019, pp. 144-150.

[2] G. Sharon, R. Stern, A. Felner, N. Sturtevant, ”Conflict-Based Search for Optimal Multi-Agent Path Finding,” in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 563-569, Toronto, Ontario, Canada: AAAI Press, 2012.

[3] E. Boyarski et al., “ICBS: The Improved Conflict-Based Search Algorithm for Multi-Agent Pathfinding,”

[4] Y. Shi, B. Hu, and R. Huang, ”Task Allocation and Path Planning of Many Robots with Motion Uncertainty in a Warehouse Environment,” in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2021, pp. 776-781.

[5] J. Li, D. Harabor, P. J. Stuckey, A. Felner, H. Ma, and S. Koenig, “Disjoint Splitting for Multi-Agent Path Finding with Conflict-Based Search”, in *ICAPS*, vol. 29, no. 1, pp. 279-283, May 2021.

XI. Appendix

Contribution of team-members till current statge of project:

1) **Literature Review:** Shambhuraj, Prasanna, Shreya
2) **2D Environment setup:** Shambhuraj, Prasanna
3) **Algorithm writing:** Shreya, Prasanna
4) **Documentation:** Shreya, Shambhuraj