© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715558

Collision-Free Multi-Robot Navigation Through Independent Path Computation and Local Coordination

ONKARINI HATWAR[1], ASTHA SAKARKAR[2], HITESHRI POTARKAR[3], SAHIL BARTINE[4], SAMAY BHARADKAR[5]

[1, 2, 3, 4, 5] Department of Computer Technology, Kavikulguru Institute of technology and science (KITS), Ramtek, Nagpur, India.

Abstract - This paper presents a decentralized approach for collision-free navigation in multi-robot systems, where each robot independently computes its own path while coordinating locally with neighboring robots to avoid collisions. Unlike centralized path planning methods, the proposed approach eliminates reliance on a global controller and instead emphasizes robot-level autonomy and local interaction. Each robot initially plans a path using a classical single-agent planner and dynamically adjusts its motion when potential collisions are detected. Local coordination strategies such as temporary waiting, speed adjustment, and resolve conflicts in real time. To validate this proposed framework, we developed a comprehensive multirobot simulation environment using Python. The system was tested on various grid maps with static obstacles to evaluate performance metrics such as path efficiency and collision counts. The approach improves scalability, robustness, and adaptability in dynamic environments. Simulation-based experiments demonstrate that the proposed method achieves zero collision rates while maintaining acceptable path efficiency and computational performance. The results indicate that independent path computation combined with local coordination is a practical and effective solution for decentralized multi-robot navigation.

Keywords: Multi-Robot Navigation, Collision-Free Path Planning, Decentralized Systems, Autonomous Robots, Local Coordination

I. INTRODUCTION

Multi-robot systems are increasingly deployed in applications such as warehouse automation, surveillance, and search-and-rescue operations. In these scenarios, robots must reach their respective goals efficiently while sharing a common workspace. The primary challenge lies in ensuring collision-free navigation as the number of robots increases.

Centralized multi-robot path planning approaches compute joint trajectories for all robots [2][3] but suffer from scalability limitations, high computational complexity, and single-point failure risks. In contrast, decentralized approaches allow robots to operate autonomously using local information, offering improved robustness and scalability. However, purely reactive decentralized methods may lead to deadlocks or inefficient paths.

This work addresses these challenges by combining independent path computation with local coordination. Each robot plans an optimal path using the A* algorithm while resolving conflicts through local interactions, achieving safe and scalable navigation without centralized control.

II. PROBLEM FORMULATION

Consider a set of (N) mobile robots operating in a twodimensional workspace populated with static obstacles. Each robot (i) is assigned a unique start position (s_i) and goal position (g_i). Robots are modeled as circular agents with radius (r). A collision occurs if the distance between any two robots becomes less than (2r).

The objective is to generate collision-free trajectories such that all robots reach their respective goals while relying only on local sensing or limited communication. The system operates without centralized supervision, emphasizing decentralized decision-making and scalability.

Among various path planning techniques, the A* algorithm is one of the most widely adopted heuristic search methods[1] in artificial intelligence–based

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715558

navigation systems. By incorporating heuristic information into the search process, the A* algorithm effectively eliminates a large number of unnecessary search nodes, thereby significantly improving computational efficiency. As a result, it is particularly well suited for solving shortest-path problems in static environments. Using the A* algorithm, a robot can efficiently determine an optimal motion trajectory from its initial position to the target position in an obstacle-free or known environment, producing a path that balances accuracy and computational performance.

2.1. A ALGORITHM PRINCIPLE*
The A* algorithm is a highly efficient path optimization technique that offers faster convergence compared to classical graph search algorithms such as Dijkstra's method.

A. Algorithm Overview
The A* algorithm is a heuristic search method widely used for optimal path planning in static environments. It evaluates nodes based on a cost function that balances the actual path cost and an estimated cost to the goal.

The evaluation function is defined as
[f(n) = g(n) + h(n)]
where (g(n)) represents the accumulated cost from the start node to node (n), and (h(n)) denotes a heuristic estimate of the cost from node (n) to the goal.

B. Heuristic Function
In grid-based environments, the Manhattan distance heuristic is commonly employed and is given by
[h(n) = |x_n - x_g| + |y_n - y_g|]
where ((x_n, y_n)) and ((x_g, y_g)) are the coordinates of node (n) and the goal, respectively. This heuristic is admissible, as it never overestimates the true cost, thereby guaranteeing optimality of the A* solution.

*C. Justification of A**
Advanced planners such as D*[17], Theta*[18] ,and cooperative algorithms like M*[19] address dynamic or joint-state planning problems but incur higher computational overhead. In this work, classical A* is selected due to its deterministic behavior, computational efficiency, and suitability for independent path computation in decentralized multirobot systems.

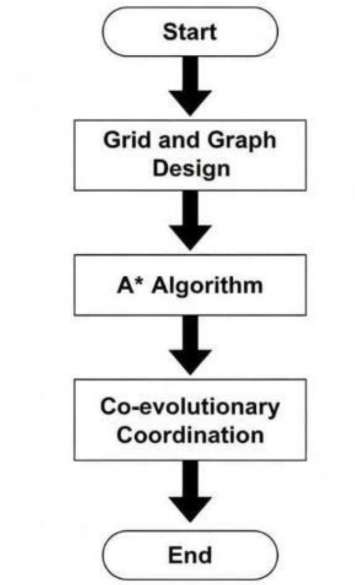

Fig 3. Data Flow Diagram of Multi Agent Path Planning

3.1 Independent Path Computation
Initially, each robot computes a path from its start location to its goal using a classical single-robot path planning algorithm such as A*. This planning phase considers only static obstacles and ignores other robots, allowing efficient individual path generation.

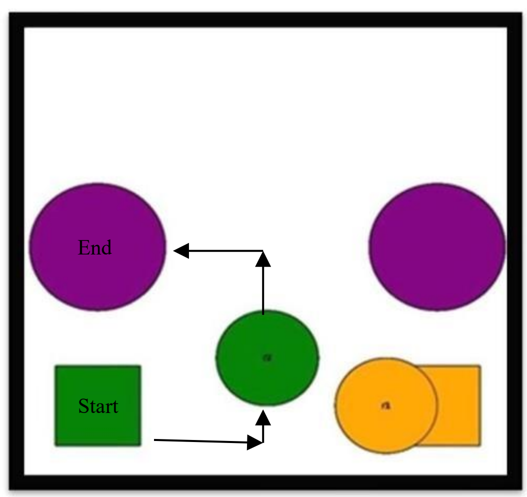

Fig 3.1 Independent path planning

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715558

The figure depicts the initial configuration and navigation objectives of the agents within a shared workspace. The green square indicates the starting position of an agent, while the green circular marker represents the agent's initial physical location. The purple circular regions denote the assigned goal positions, and the remaining geometric shapes correspond to static obstacles that constrain the feasible motion space. Each agent independently computes an optimal path from its start location to the corresponding goal using the A* path planning algorithm, which evaluates candidate trajectories by minimizing a cost function composed of the accumulated travel cost and a heuristic estimate of the remaining distance to the destination. This approach enables efficient generation of shortest feasible paths while ensuring obstacle avoidance, and during execution, the agent follows the planned trajectory while maintaining a safe separation from obstacles and other agents in the environment.

3.2 Collision Prediction

During navigation, robots continuously monitor nearby robots using local sensing or short-range communication. By analyzing the relative positions and velocities of neighboring robots, each robot predicts potential collisions within a short time horizon.

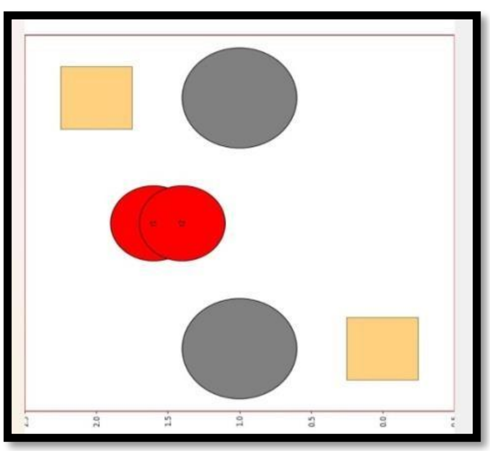

Fig 3.2 Collision Prediction of multi robot.

This Fig describe the yellow-colored regions represent the source positions from which the robots initiate their motion. The red-colored blocks denote static obstacles that restrict the available movement space and create narrow passages. The red circular agents labeled r1 and r2 are moving along nearby trajectories, and their overlapping positions indicate a predicted collision. This situation highlights the necessity of collision prediction and coordination mechanisms in multi-robot path planning to ensure safe and uninterrupted navigation.

3.3 Local Coordination and Conflict Resolution

When a potential collision is detected, robots resolve conflicts through local coordination strategies. These include temporary speed reduction, waiting at safe locations, or locally replanning a short segment of the path. All decisions are made independently based on local observations, ensuring decentralized operation.

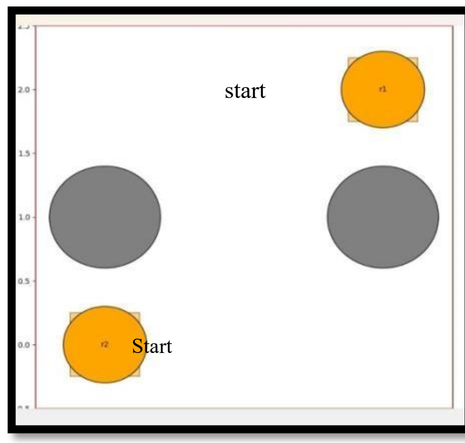

Fig 3.3.1 local coordination and conflit resolution

The Fig describe the environment after the agents have initiated movement from their respective source locations. The yellow circular agents have successfully navigated through the workspace while maintaining collision-free trajectories with respect to the static obstacles. The absence of overlap between the agent boundaries and the gray obstacle regions confirms that obstacle avoidance constraints are strictly satisfied throughout the motion. This figure demonstrates the capability of the agents to move freely within the environment, transitioning from source regions toward their intended positions without any obstacle interference, thereby validating the effectiveness of the adopted navigation and collision-

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715558

avoidance strategy.

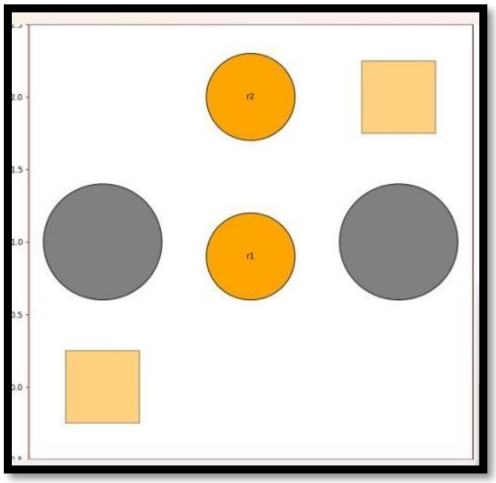

Fig.3.3.2 Local coordination and conflict resolution by moving to destination.

The Fig illustrates the initial configuration of the navigation environment. The square blocks denote the source locations from which the agents are deployed, while the yellow circular shapes represent the mobile agents at their starting positions. The gray circular regions correspond to static obstacles that occupy fixed locations in the workspace and restrict the feasible motion space. At this stage, both agents (labeled *r1* and *r2*) are positioned near their respective source regions, and no motion has yet occurred. The spatial distribution of obstacles creates constrained corridors, emphasizing the need for effective path planning to ensure safe navigation.

III. ALGORITHM OVERVIEW

The overall navigation process follows these steps:

1. Initialize robot positions and goals.
2. Each robot independently computes an initial path.
3. Robots move along their paths.
4. Robots detect nearby robots and predict collisions.
5. Local coordination strategies are applied if conflicts arise.
6. Robots update their motion and continue navigation.
7. The process repeats until all robots reach their goals.

IV. EXPERIMENTAL SETUP

The proposed approach was evaluated in a two dimensional environment with static obstacles. Experiments were conducted with varying numbers of robots to assess scalability. Performance was evaluated using metrics such as collision count, average path length, travel time, and goal-reaching success rate.

Table 1. Performance comparison between existing multi-robot navigation approaches and the proposed method.

Method	Planning Type	Scalability	Collision Handling	Centralized Control	Deadlock Risk
Centralized Planning	Global	Low	Explicit	Required	None
Reactive Avoidance	Local	High	Implicit	Not Required	High
Prioritized Planning	Semi-Global	Medium	Explicit	Partial	Medium
Learning-Based Methods	Adaptive	Medium	Learned	Optional	Uncertain
Proposed Method	Hybrid	High	Local Coordination	Not Required	Very Low

V. RESULTS AND DISCUSSION

The results obtained from the implementation of multi-robot path planning are highly informative in understanding how robots can operate in dynamic and static environments. This section provides a comprehensive and detailed analysis of the outputs, focusing on both the single robot scenario and the multi-robot coordinated navigation scenario. The purpose of this analysis is to highlight the significance of the results in demonstrating the effectiveness of the system architecture and to connect the observed outcomes with the underlying algorithms and techniques such as A* search, co-evolutionary coordination, and Bezier smoothing.

VI. CONCLUSION

This paper presented a decentralized collision-free multi-robot navigation approach based on independent path computation and local coordination. The method eliminates the need for centralized control while ensuring safe and scalable navigation. Future work will focus on real-world robot implementation and the integration of learning-based collision prediction techniques.

ACKNOWLEDGMENT

The authors express their sincere gratitude to the Computer Technology Department at Kavikulguru Institute of Technology & Science (KITS), Ramtek, for providing the necessary infrastructure and support to conduct this research on python based virtual assistant we are thankful to our colleagues and research advisors for their invaluable guidance and constructive feedback throughout the project. Special appreciation is extended to the IEEE and arXiv communities for their extensive resources, which enriched our literature review.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, Cambridge, UK, 2006.

[2] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multiagent navigation," *IEEE International Conference on Robotics and Automation*, pp. 1928–1935, 2008.

[3] M. Bennewitz, W. Burgard, and S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems," *IEEE International Conference on Robotics and Automation*, pp. 271–276, 2001.

[4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[5] Y. Guo and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," *IEEE International Conference on Robotics and Automation*, pp. 2612–2619, 2002.

[6] J. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, no. 3, pp. 171–194, 1999.

[7] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.

[8] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[9] J. Čáp, P. Novák, M. Vokřínek, and R. Švestka, "Multi-agent pathfinding with continuous time," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 3, pp. 549–579, 2016.

[10] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[11] T. Fraichard and H. Asama, "Inevitable collision states—A step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[13] H. Choset et al., *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.

[14] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163–1177, 2016.

[15] A. Kolling, N. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on HumanMachine Systems*, vol. 46, no. 1, pp. 9–26, 2016.

[16] Koenig S, Likhachev M. Fast replanning for navigation in unknown terrain. IEEE Transactions on Robotics. 2005; 21(3):354-363.

[17] Ferguson D, Stentz A. Using interpolation to improve path planning: The Field D* algorithm. Journal of Field Robotics.2006; 23(2): 79-101.

[18] Daniel K, Nash A, Koenig S, Felner A. Theta*: Any-Angle Path Planning on Grids. Journal of Artificial Intelligence Re-search. 2010 Oct; 39: 533579.

[19] Wagner G, Choset H. M*: A complete multirobot path planning algorithm with performance bounds. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2011. pp. 3260-3267.

[20] Wagner G, Choset H. Subdimensional expansion for multirobot path planning. Artificial Intelligence. 2015; 219:1-24.

[21] Standley TS. Finding Optimal Solutions to Cooperative Pathfinding Problems. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligences; 2010. pp. 173-178.

[22] Standley TS, Korf RE. Complete Algorithms for Cooperative Pathfinding Problems. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence; 2011.pp. 668-673.