iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74228

A MATLAB-Based Comparative Evaluation of ACO and A* in Dynamic Environments and PSO for Static Pathfinding

Davu Manikanta[1], Ravi Jeevan[2], Pulipati Mary[3], Addanki Rajesh[4], Annavarapu Venkata Kanaka Durga Devi[5], Lanka Durga Prasad[6], Nanda Kesava Murthi[7]
[1, 2]Professor, [3, 4, 5, 6, 7]Student, Department of Electronics and Communication Engineering, Lingayas Institute of Management and Technology, Vijayawada, Andhra Pradesh, India

Abstract: Pathfinding is a critical problem in robotics, artificial intelligence, and network optimization. This paper presents a comparative evaluation of three popular algorithms: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and A* search. Implemented in MATLAB, the study analyzes their performance in terms of path length, execution time, convergence behavior, and adaptability under both static and dynamic grid environments. Results indicate that A* consistently provides the shortest paths with minimal computation time, ACO demonstrates robustness in dynamic environments, and PSO offers a balance between accuracy and efficiency. The findings contribute to the selection of suitable algorithms for real-world applications in robotics, multi-agent systems, and intelligent navigation.
Keywords: Pathfinding, Ant Colony Optimization, Particle Swarm Optimization, A* Algorithm, MATLAB

I. INTRODUCTION

Pathfinding is one of the most fundamental and widely studied problems in artificial intelligence, robotics, and computer science. It involves determining the most efficient route from a source to a destination in an environment that may contain static or dynamic obstacles. Efficient pathfinding algorithms are critical in numerous applications such as autonomous navigation for robots and self-driving vehicles, real-time strategy games, traffic management systems, wireless sensor networks, and intelligent transportation. An effective pathfinding strategy directly impacts system performance by minimizing resource consumption, reducing time delays, and enhancing decision-making capabilities.

Among the many techniques developed, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and the A* algorithm have emerged as popular and reliable approaches. Each of these algorithms is based on a distinct principle. ACO is a bio-inspired metaheuristic that simulates the foraging behavior of ants, where pheromone trails help guide the search toward optimal paths. Its decentralized and adaptive nature makes it highly effective in dynamic environments where obstacles may change over time. PSO, on the other hand, is a population-based optimization technique that models the collective behavior of bird flocks or fish schools. By iteratively updating the velocity and position of particles based on personal and group experience, PSO achieves rapid convergence toward feasible solutions while maintaining balance between exploration and exploitation. Unlike these probabilistic approaches, the A* algorithm is a deterministic heuristic-based search method. By combining the actual cost from the start node and the heuristic estimate to the goal, A* ensures the shortest possible path if the heuristic is admissible.

While A* is widely recognized for its accuracy and efficiency in structured, static environments, it often struggles in large or dynamic environments due to its exhaustive search nature. ACO, although computationally expensive, adapts better in uncertain or changing conditions. PSO provides a middle ground, producing near-optimal paths with reasonable execution time and computational efficiency.

This project focuses on a comparative evaluation of ACO, PSO, and A* within a MATLAB-based grid simulation framework. Each algorithm is implemented under identical conditions, and their performances are assessed based on path length, execution time, convergence rate, and adaptability to environmental changes. By systematically comparing these techniques, the study aims to highlight their respective strengths and limitations, providing valuable insights into selecting the most suitable algorithm for specific real-world applications. The findings contribute to the development of efficient path planning strategies in robotics, multi-agent systems, and intelligent navigation, where adaptability, accuracy, and computational efficiency are essential.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

The findings contribute not only to academic understanding but also to practical engineering solutions, where adaptive and efficient pathfinding is vital. In the future, hybrid models that combine the deterministic precision of A* with the adaptive power of ACO and PSO may emerge as robust solutions for complex, real-time environments.

II. DESIGN AND IMPLEMENTATION OF PATHFINDING ALGORITHMS (ACO, PSO, AND A*)

The comparative framework for pathfinding was developed by designing a grid-based simulation environment in MATLAB where all three algorithms—ACO, PSO, and A*—were implemented under uniform conditions. The grid was configured as a 20×30 matrix with approximately 20% of the cells randomly designated as obstacles, ensuring a realistic navigation scenario. The start node was placed at the top-left corner (1,1) and the goal node at the bottom-right corner (20,30).

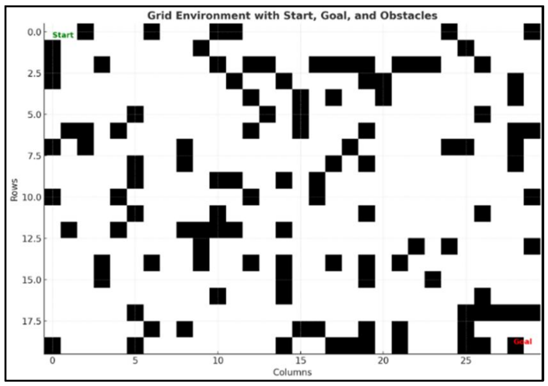
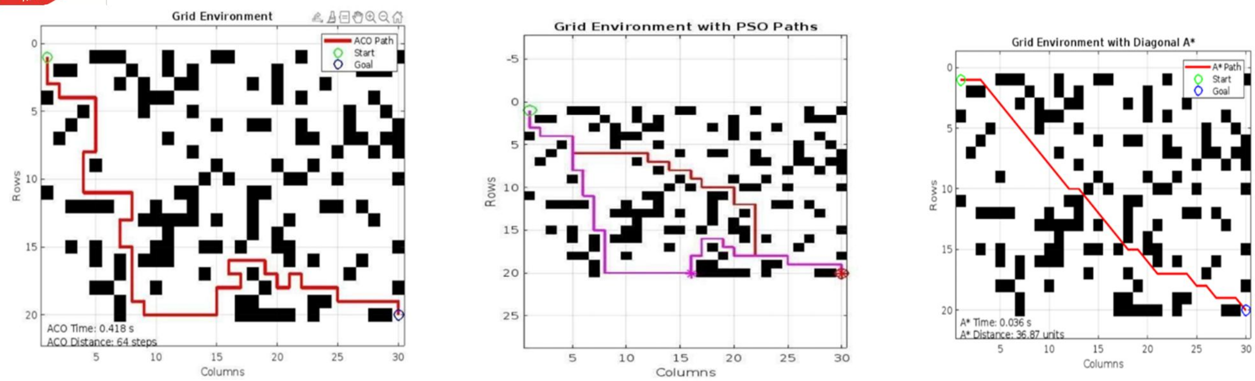


Figure.1 Grid Environment

(a) (b) (c)

Figure 2. (a) ACO Flowchart (b) PCO Flowchart (c) A* Flowchart

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com



(a) (b) (c)

Figure 3. (a)ACO Simulation Result Screenshot. (b) PCO Simulation Result Screenshot. (c) A* Simulation Result Screenshot.

In this setup, ACO mimics the foraging behavior of ants, where artificial agents deposit and follow pheromone trails. This allows the system to gradually converge towards efficient paths, even when obstacles shift dynamically. PSO, in contrast, employs a swarm of particles that continuously update their positions based on both personal and global best experiences. This helps the swarm converge rapidly toward promising paths, balancing exploration and exploitation. Finally, the *A algorithm** expands the most promising nodes based on the cost function f(n)=g(n)+h(n), guaranteeing the shortest possible path under admissible heuristics such as Manhattan distance.

Each algorithm was implemented independently but tested on the same grid environment, ensuring consistency in evaluation. The separation of algorithm modules in MATLAB provided independence and comparability, similar to how separate ground planes in antenna design ensure independent radiation characteristics. This modular approach minimizes computational interference between the algorithms and ensures unbiased results.

Given the complexity of real-world navigation, this simulation setup is well-suited for evaluating algorithm performance in autonomous robotics, wireless sensor networks, and intelligent transportation systems, where adaptability, convergence speed, and computational efficiency directly impact reliability.

Figure 2 (a) illustrates the path generated by the ACO algorithm in MATLAB. The ants initially explore multiple random paths, depositing pheromone trails as they traverse. Over successive iterations, the pheromone concentration strengthens along shorter and more efficient routes. The resulting path length is 64 steps, with an execution time of 0.418 seconds. Although slower than other methods, the ACO algorithm demonstrates excellent adaptability when obstacles are dynamically introduced, as ants naturally redirect their search, ensuring feasible solutions under changing environments.

Figure 2 (b) shows the output of the PSO algorithm. The particles converge toward the goal by updating their velocities and positions based on both their own best experience (Pbest) and the swarm's global best (Gbest). The optimal path length achieved is 48 steps, with an average execution time of 0.0068 seconds. This demonstrates PSO's capability of striking a balance between accuracy and computational efficiency. While convergence is faster than ACO, PSO may occasionally settle in near-optimal solutions rather than the absolute best, depending on swarm parameters.

Figure 2 (c) depicts the result of the *A algorithm**. By expanding nodes with the lowest cost function f(n) = g(n) + h(n), A* consistently yields the shortest path of 37 steps, with a very low execution time of 0.0037 seconds. The deterministic nature of A* ensures guaranteed optimal paths in static environments. However, when obstacles are modified dynamically, A* requires re-computation from scratch, making it less adaptive compared to ACO.

Figure 3 (a) compares the execution time of all three algorithms. The results clearly show that A* performs fastest, followed closely by PSO, while ACO is computationally the slowest due to its iterative nature.

Figure 3 (b) illustrates the path length comparison, where A* produces the shortest path, PSO a near-optimal path, and ACO the longest route among the three.

Figure 3 (c) provides a convergence graph, showing how pheromone intensity stabilizes in ACO, how swarm particles align in PSO, and how node expansions progress in A*. These trends highlight the strengths and weaknesses of each method

Overall, the results confirm that A* is best suited for static and structured environments, ACO excels in dynamic conditions requiring adaptability, and PSO serves as a middle ground for scenarios demanding both efficiency and near-optimal accuracy.

III. CONCLUSION

This work presented a MATLAB-based comparative evaluation of three widely used pathfinding algorithms: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and A*. Each algorithm was analyzed in terms of path length, execution time, convergence behavior, and adaptability within a grid-based simulation environment.

The results demonstrate that the A algorithm* consistently provides the shortest path with the least computation time, making it the most efficient method for static and well-structured environments. The ACO algorithm, though slower, exhibits strong adaptability in dynamic scenarios, where obstacles change over time, thereby ensuring reliable path discovery in uncertain conditions. The PSO algorithm offers a balanced compromise, achieving near-optimal results with moderate computational effort and stable convergence performance. From this comparative study, it is clear that the selection of an appropriate pathfinding algorithm depends on the nature of the application environment. For applications requiring guaranteed shortest paths in static grids, A* is preferable. For systems operating in dynamic and unpredictable environments, ACO is more suitable, while PSO can be applied where a trade-off between speed and optimality is acceptable.

Future work can extend this research by implementing hybrid pathfinding approaches, combining the deterministic efficiency of A* with the adaptive intelligence of ACO and PSO. Additionally, deploying these algorithms on real-time robotic platforms and multi-agent navigation systems will further validate their performance in practical scenarios.

REFERENCES

[1] Ziyao Li, “Improved Ant Colony Algorithm for Emergency Rescue Vehicle Routing,” Proceedings of International Conference on Computer and Communication Technologies, 2010.

[2] Rohan Shakiba, Mehrdad Saadatzi, Benyamin Ghojogh, “Path Planning for Humanoid Soccer Robots using Particle Swarm Optimization,” International Conference on Advanced Robotics, 2013.

[3] Risma Septiana, Yuslena Sari, Eko Didik Widianto, “Optimization of A Algorithm for Routing in Wireless Sensor Networks,”* International Journal of Computer Applications, Vol. 141, No. 12, 2016.

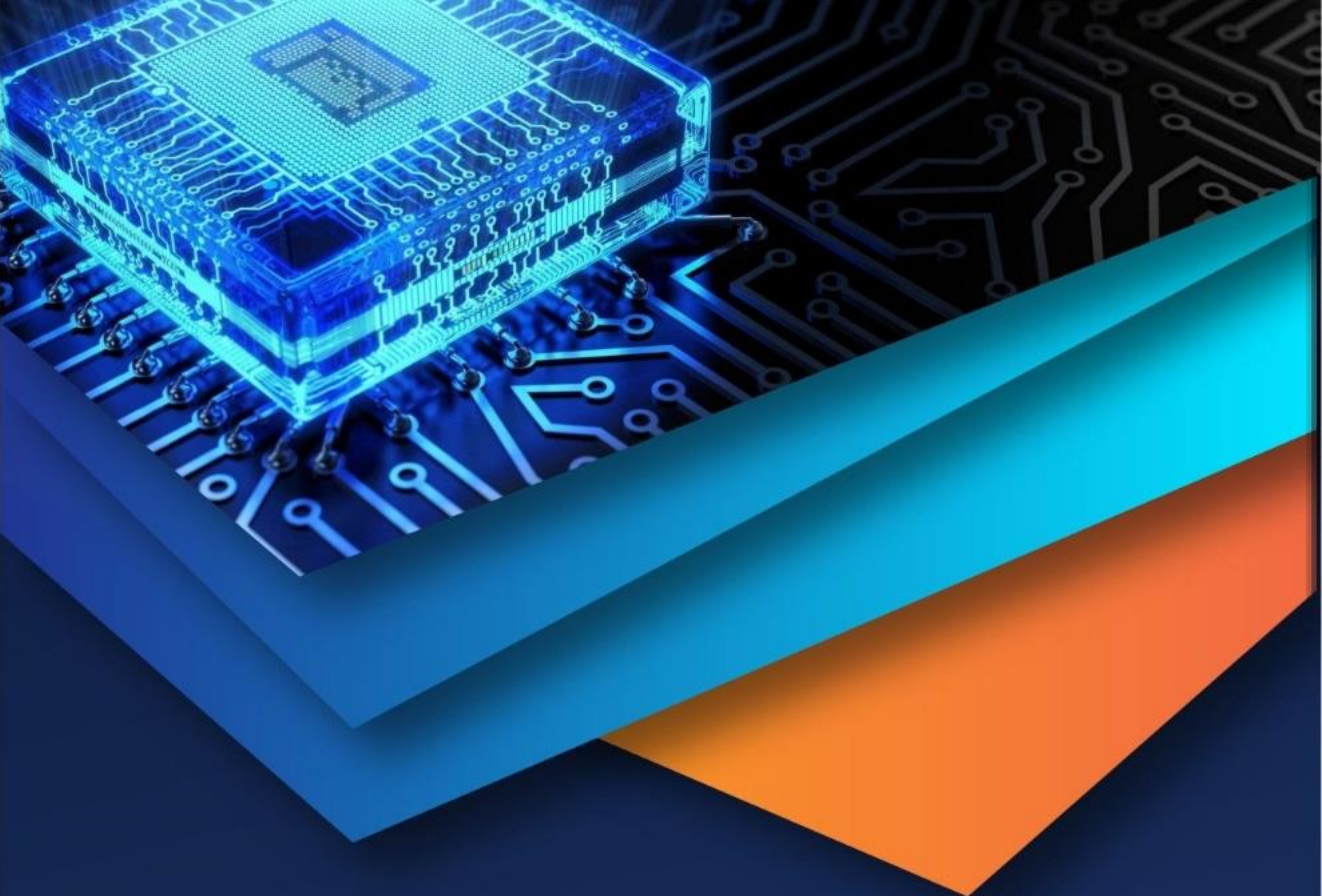
[4] Peng-Fei Li, Shu-Qin Li, Qian-Sheng Fang, “Improved Ant Colony Algorithm for Ship Weather Routing,” International Journal of Naval Architecture and Ocean Engineering, Vol. 10, No. 2, 2018.

[5] Jinyu Tang, Xiaodong Liu, Wenlong Fu, “An Improved Ant Colony Algorithm with Weighted Penalty Strategy,” Journal of Intelligent & Fuzzy Systems, Vol. 38, 2020.

[6] Dorigo, M., Maniezzo, V., and Colorni, A., “Ant System: Optimization by a Colony of Cooperating Agents,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 26, No. 1, pp. 29–41, 1996.

[7] Kennedy, J., and Eberhart, R., “Particle Swarm Optimization,” Proceedings of the IEEE International Conference on Neural Networks, 1995.

[8] Hart, P. E., Nilsson, N. J., Raphael, B., “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, pp. 100–107, 1968.



10.22214/IJRASET

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY