2025 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Funchal, Portugal – April 2-3, 2025

# Hybrid Wavefront and Bidirectional A* Coverage Path Planning for AGVs in Photovoltaic Farm Inspection

Gabriel G.R.Castro[1], Lucas L. M. Carvalho[1], Diogo G. Marques[1], José Lima[2,3], Milena F. Pinto[1]
[1] *Federal Center for Technological Education Celso Suckow da Fonseca, Rio de Janeiro, RJ, Brazil*
[2] *Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Portugal*
[3] *INESC Technology and Science, Porto, Portugal*

2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 979-8-3315-3860-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICARSC65809.2025.10970161

***Abstract*—This work presents a novel Coverage Path Planning (CPP) framework integrating the Wavefront Algorithm with bidirectional A* to optimize path planning for Autonomous Ground Vehicles (AGVs) that inspect photovoltaic farms. The proposed framework combines the efficient propagation capabilities of the Wavefront Algorithm with the heuristic-driven optimization of bidirectional A*, comparing the results achieved to those from the individual algorithms. The framework was validated in a Software-In-The-Loop (SITL) simulation environment using Gazebo and the Robot Operating System (ROS), focusing on AGV-based inspection of ground infrastructure and cables beneath solar panels. The outcomes demonstrate significant coverage efficiency and overall robustness.**

***Index Terms*—Autonomous Ground Vehicle, Robotic Path Planning, Renewable Energy Systems.**

## I. Introduction

The detection of anomalies in photovoltaic (PV) systems has gained significant attention in recent years. For instance, Just in 2022, Raptor Maps [1] registered $82 million in losses, for the 24.5GW analyzed in the period, caused by anomalies in the equipment (nearly double from 2019, from 1,61% to 3,13% in losses). These anomalies resulted in $2.5B annually of loss if scaled to the entire industry. Such irregularities can negatively affect energy production and thermal regulation [2]. Several studies have investigated the performance and maintenance of solar panels, focusing on issues like hot spots and overheating [3]. However, few have addressed the inspection of the broader PV infrastructure, including supporting structures and cabling. Cables, in particular, play a critical role in ensuring efficient energy transmission but are vulnerable to damage from wear, environmental conditions, and theft, highlighting the importance of regular diagnostic evaluations [4]. In this context, advanced technologies such as robots have been developed to address these challenges by automating maintenance tasks.

A well-designed path-planning strategy is essential for these robotic systems to perform inspections effectively. This strategy must ensure that the robots pass in all points of a determined area of interest while avoiding obstacles [5]. This challenge falls under the Coverage Path Planning (CPP) problem category. The CPP problem is closely related to the covering salesman problem, a variant of the classic traveling salesman problem (TSP), where instead of visiting each specific city, the agent must visit a neighborhood around each city [6]. However, CPP's objective shifts from visiting neighborhoods to ensuring complete coverage of every point within a defined target area. CPP is widely applied across a diverse range of robotic applications, including agricultural monitoring [7], inspection of structures [8], autonomous aerial surveys for mapping and environmental monitoring [9], among others.

The Wavefront Algorithm, first introduced by Zelinsky et al. [10], is a grid-based method for CPP that assigns numerical labels to grid cells by propagating a wave from the goal to the start cell, ensuring complete area coverage, much like Dijkstra's Algorithm. Originally designed for offline planning, it requires predefined start and goal cells and systematically labels neighboring cells to guide the path. Shivashankar et al. [11] later extended this approach to enable online coverage in unknown environments, making it suitable for dynamic and real-time applications. Building on these foundational works, this research integrates the Wavefront Algorithm with bidirectional A* (BA*) to address limitations in handling complex and obstacle-rich environments. By combining the efficient propagation of the Wavefront Algorithm with the heuristic optimization capabilities of A*, the proposed method ensures faster and more robust path planning. In the literature, several researchers have proposed modifications to the wavefront algorithm to improve performance. Schäfle et al. [12] proposed a hybrid genetic algorithm for performing local search in a grid-based environment representation. In the work of [13], the authors have used the Focused Wave Front Algorithm for robot path planning in a 2D static environment, trying to counter the problem related to the full wave expansion. Zidane et al. [14] introduced a hybrid Wavefront and A* approach to mitigate issues such as dead ends, U-shape traps, shortest path optimization, and time efficiency in path planning. However, as noted, few studies have explored the application of these algorithms in coverage path planning for large areas, which is the focus of this work.

The proposed method combines the efficient propagation capabilities of the Wavefront algorithm with the heuristic-driven optimization power of A*. This integration enables faster and more robust path planning in complex environments containing multiple obstacles. The Wavefront algorithm is adapted to incorporate directed heuristics, while bidirectional A* serves as a fallback mechanism for handling challenging areas or optimizing trajectories. By employing bidirectional A*, the system further improves efficiency by reducing the search space through simultaneous propagation from the start and goal points. To validate the effectiveness of the proposed approach, it is implemented and tested in a Software-In-The-Loop (SITL) environment using the Gazebo simulation platform [15] and the Robot Operating System (ROS) framework. The main contributions of this research can be summarized as follows:

- Development of a hybrid approach that utilizes Wavefront and bidirectional A* algorithm for efficient and robust coverage path planning. The application of this hybrid approach is performed in a PV farm scenario to improve the inspection of lower structures, such as cables or foundations, to ensure the system's integrity.
- Assessment of the performance of the proposed framework through simulations in ROS and Gazebo, demonstrating its effectiveness and practicality for real-world scenarios such as solar panel inspection.

The remainder of this research is organized as follows: Section II outlines the mathematical foundations and provides a detailed explanation of the proposed CPP framework. Section III presents the results obtained from SITL simulations, along with a comprehensive analysis of the outcomes. Finally, Section IV concludes with final remarks and suggestions for future improvements.

## II. Proposed Approach

The process begins with receiving a map in the OccupancyGrid format, which is processed to identify occupied areas and apply a ”buffer” around them. This buffer creates a safety zone, expanding obstacles to account for the robot's size and prevent collisions. After processing, the map is converted into a graph where free cells are connected to their neighbors, enabling path analysis. The Wavefront algorithm is then used for path planning. It assigns numerical values to graph cells, starting from the destination. From the goal, a ”wave” propagates, marking adjacent cells with incremented values that indicate their distance from the target. This creates a gradient that the robot can follow efficiently to reach the destination.

When the Wavefront algorithm encounters challenges such as disconnected areas or dead ends, it falls back to the bidirectional A* algorithm. Unlike Wavefront, the bidirectional A* divides the workload between two points: the start and the goal. Both sides simultaneously explore the map, converging when their search areas meet. This approach significantly reduces the search space by focusing on two smaller regions instead of the entire space between the start and the goal. The bidirectional A* search employs a heuristic to evaluate the most promising cells based on their estimated distance

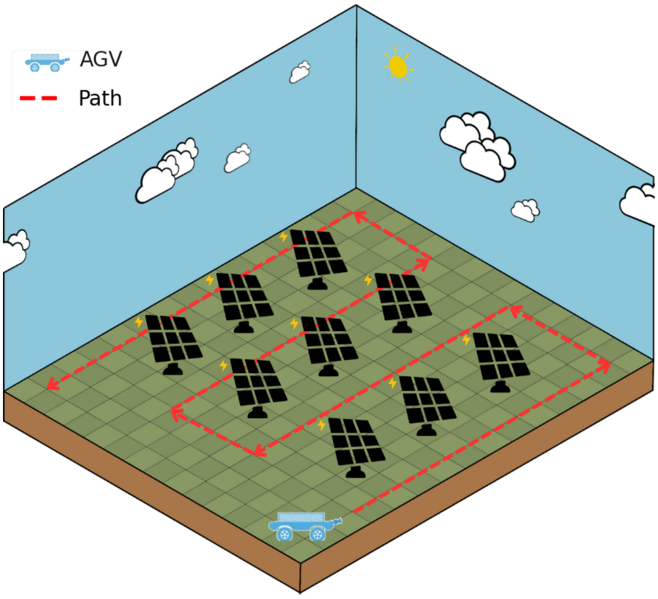



Fig. 1: General overview of the proposed solution.

to both the goal and the start. Upon finding the point of convergence, the algorithm combines the paths generated from each side to create a complete trajectory. This technique is particularly efficient in large or complex maps as it minimizes the computational load required for planning. Figure 1 presents the overall idea of the proposed approach.

### A. Problem Definition

The problem addressed in this work is the safe and efficient generation of a trajectory that covers an entire environment represented as an *occupancy grid*. This environment is modeled as a discrete grid where each cell is classified as either free or occupied. The primary objective is to generate a continuous, collision-free path that ensures complete navigation through all free cells within the environment. Mathematically, the path $\mathcal{P}$ is defined as:

$$\mathcal{P} = \{p_1, p_2, \ldots, p_n\}, \ p_i \in \mathcal{A}_{\text{free}}, \tag{1}$$

where $\mathcal{A}_{\text{free}}$ denotes the set of free cells, ensuring that all points in the path avoid obstacles. Collision avoidance is guaranteed by ensuring that the path does not intersect with any obstacle cells, defined as $\mathcal{A}_{\text{obs}}$, satisfying the condition:

$$\mathcal{P} \cap \mathcal{A}_{\text{obs}} = \varnothing.$$

To achieve this, the Wavefront algorithm is used to assign weights to the cells in the grid. This algorithm propagates a ”wave” starting from the goal point ($p_{\text{goal}}$) and moving backward toward the starting point ($p_{\text{start}}$), creating a gradient field that guides the robot's movement. It is important to note that the weights assigned to the cells represent the number of moves required to reach the goal rather than the Euclidean distance between cells. This approach simplifies the cost calculation while still ensuring effective path planning. The

2025 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Funchal, Portugal – April 2-3, 2025

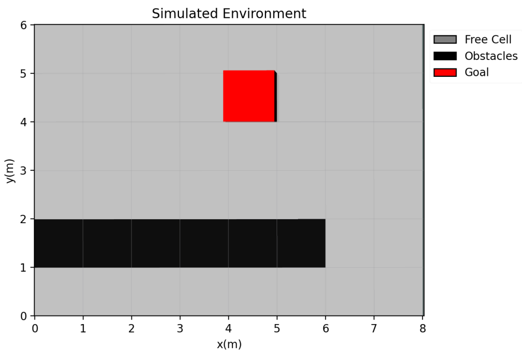
weight of each cell $w(p)$ is calculated using the following equation:

$$w(p) = \min\{w(p_{\text{adj}}) + 1\},\ p_{\text{adj}} \in \text{neighbors}(p), \tag{2}$$

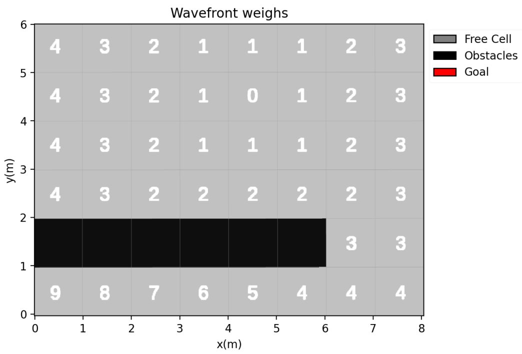
where $p_{\text{adj}}$ represents adjacent cells and neighbors$(p)$ is described as:

$$\text{neighbors}(p) = \{(x \pm 1, y), (x, y \pm 1), (x \pm 1, y \pm 1)\}. \tag{3}$$

This neighbor's definition allows diagonal movement in an eight-connected grid, providing greater flexibility in movement and potentially shorter paths. The gradient generated by the wavefront allows the robot to find, if it exists, a path that passes through all grids that belong to $\mathcal{A}_{\text{free}}$. Figure 2 shows an example of wavefront propagation in a simulated environment.




(a)




(b)

Fig. 2: Wavefront propagation in a simulated environment. (a) The simulated environment, where the red square represents the goal; (b) The simulated environment displaying the wavefront weights of each cell, propagated outward from the goal grid.

In cases where the Wavefront algorithm fails to identify a valid path, a fallback mechanism is activated, employing the bidirectional A* algorithm. This algorithm is used to compute a path from the last cell, where the Wavefront algorithm halted, to the nearest accessible free cell in the environment. Let the cell where the Wavefront propagation halts be denoted as $p_{halt}$, and the nearest free cell be nearest $p_{nearest}$, such that:

$$p_{nearest} = \arg\min_{p \in \mathcal{A}_{\text{nv}}} d(p_{halt}, p), \tag{4}$$

where $d(p_i, p_j)$ represents the manhattan distance between cells $p_i$ and $p_j$, and $\mathcal{A}_{\text{nv}}$ the set of unvisited cells within $\mathcal{A}_{\text{free}}$.The bidirectional A* algorithm is then applied to find a path satisfying:

$$\mathcal{P}_{\text{A}*} = \{p_1, p_2, \ldots, p_n\},\ p_i \in \mathcal{A}_{\text{free}}, \tag{5}$$

$$p_1 = p_{\text{halt}}, \quad p_n = p_{\text{nearest}},$$

The algorithm searches simultaneously from two directions: forward from the start point ($p_{\text{halt}}$) and backward from the goal point ($p_{\text{nearest}}$). Each search maintains its own priority queue, $Q_{\text{start}}$ and $Q_{\text{goal}}$, ordered by a cost function $f(p)$, where:

$$f(p) = g(p) + h(p). \tag{6}$$

Here, $g(p)$ is the actual cost of the path from the search's starting point to the current cell $p$, and $h(p)$ is the heuristic function that estimates the cost from $p$ to the respective goal of the search.

In each iteration:

1) A cell $p_{\text{current}}$ with the lowest $f(p)$ value is expanded from $Q_{\text{start}}$ and $Q_{\text{goal}}$, simultaneously.
2) The algorithm checks if $p_{\text{current}}$ from one direction has already been visited by the other direction. If so, the searches are considered met, and the final path is constructed by combining the partial paths.

The termination condition is met when:

$$Q_{\text{start}} \cap Q_{\text{goal}} \neq \varnothing. \tag{7}$$

The combined path $\mathcal{P}_{\text{A}*}$ is computed as:

$$\mathcal{P}_{\text{A}*} = \mathcal{P}_{\text{start}} + \mathcal{P}_{\text{goal}}^{\text{rev}}, \tag{8}$$

where $\mathcal{P}_{\text{start}}$ is the path from $p_{\text{halt}}$ to the meeting point, and $\mathcal{P}_{\text{goal}}^{\text{rev}}$ is the reversed path from $p_{\text{nearest}}$ to the meeting point. This ensures that the final path is continuous and optimal.

The cost of the final path is given by:

$$L(\mathcal{P}_{\text{A}*}) = \sum_{i=1}^{n-1} d(p_i, p_{i+1}). \tag{9}$$

When the cell $p_{\text{nearest}}$ is reached, the wavefront algorithm continues to propagate the path from this cell until either $\mathcal{A}free = \varnothing$ (no free cells remaining) or it encounters another situation where it becomes stuck with no valid neighboring cells. In such cases, the bidirectional A* algorithm is invoked again to find a path to the next valid $p_{nearest}$ cell, if one exists. Using the same environment depicted in Figure 2, Figure 3 illustrates an example in which the wavefront algorithm gets stuck, showing the path found by bidirectional A* to reach the $p_{\text{nearest}}$ cell, followed by the continuation of the path computed by the wavefront algorithm.

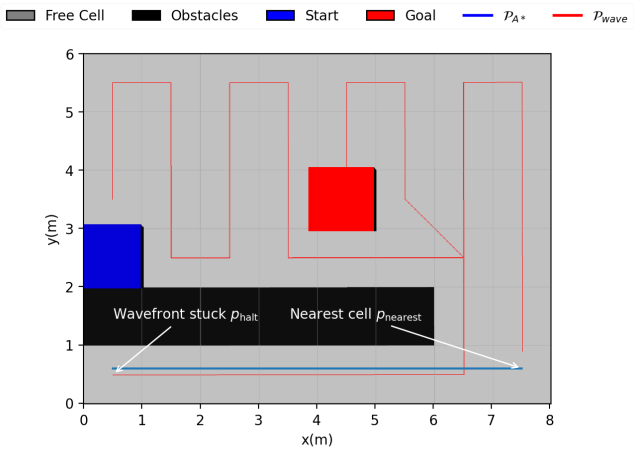



Fig. 3: Coverage path combining Wavefront and Bidercional A* algorithms

The resulting path $\mathcal{P}$ is published as a ROS message, where each waypoint includes position and orientation information. This enables the robot to navigate safely and efficiently, avoiding obstacles and adapting to the complexities of the environment. This hybrid approach, combining Wavefront and bidirectional A*, ensures complete coverage of free cells and resolves cases where Wavefront alone cannot generate a valid path. By leveraging the heuristic capabilities of A* and the global coverage of Wavefront, the method achieves computational efficiency and robustness in trajectory generation.

### B. Proposed framework

The system is designed to use the fast propagation capabilities of the Wavefront Algorithm for initial path planning and employ bidirectional A* as a fallback mechanism in challenging scenarios, such as disconnected regions or high obstacle density. The framework operates in the following stages. The environment represented as an *OccupancyGrid*, is pre-processed to identify obstacles and apply a buffer zone around them to account for the robot's dimensions. Then, a graph is constructed where free cells are nodes, and valid movements between them are edges. The Wavefront Algorithm propagates a wave from the goal cell, assigning weights to cells based on their distance from the goal. If Wavefront encounters a dead-end, the bidirectional A* is triggered to find a way out to the next near-free node. These steps are summarized in Algorithm 1. This algorithm outlines the integration of Wavefront and bidirectional A*.

**Algorithm 1:** Proposed Coverage Path Planning Framework for AGV Inspection with Iterative Search

OccupancyGrid $map$, Start $p_{\text{start}}$, Goal $p_{\text{goal}}$ Path $\mathcal{P}$
**Step 1: Map Preprocessing**
Apply a buffer to $map$ to expand obstacles considering AGV dimensions;
Convert $map$ to a graph representation $G$;
**Step 2: Wavefront Propagation**
Initialize weights $w(p) = \infty$ for all cells $p \in G$;
Set $w(p_{\text{goal}}) = 0$;
Propagate wave from $p_{\text{goal}}$ to $p_{\text{start}}$ using:

$$w(p) = \min\{w(p_{\text{adj}}) + 1\},\ p_{\text{adj}} \in \text{neighbors}(p)$$

Set $p_{\text{actual}} = p_{\text{start}}$;
Initialize the set of non-visited nodes $\mathcal{A}_{\text{nv}}$ with all navigable cells in $G$;
**while** $\mathcal{A}_{nv} \neq \emptyset$ **do**
  **Step 3: Initial Path Planning**
  Generate an initial path $\mathcal{P}_{\text{wave}}$ from $p_{\text{actual}}$ to $p_{\text{goal}}$ using Wavefront;
  **if** *Wavefront fails or $\mathcal{P}_{wave}$ is suboptimal* **then**
    **Step 4: Search for Next Nearest Node**
    Find the nearest unvisited node using:

$$p_{\text{nearest}} = \arg\min_{p \in \mathcal{A}_{\text{nv}}} d(p_{\text{actual}}, p)$$

    **Step 5: Bidirectional A* Fallback**
    Initialize two search frontiers: one at $p_{\text{actual}}$ and another at $p_{\text{nearest}}$;
    Expand both frontiers simultaneously until they meet;
    Combine paths from both sides to generate $\mathcal{P}_{\text{A*}}$;
    Mark cells in $\mathcal{P}_{\text{A*}}$ as visited;
    Update $p_{\text{actual}} = p_{\text{nearest}}$;
  **end**
  Remove visited nodes from $\mathcal{A}_{\text{nv}}$;
**end**
**Step 6: Path Publishing**
Publish $\mathcal{P}$ as a ROS *Path* message;

### C. Solution Setup

The proposed CPP approach was simulated in a software-in-the-loop (SITL) framework, running on Ubuntu 20.04 and based on Gazebo to replicate the proposed operational environment using the ROS framework that served as the backbone of the solution setup. It provided essential tools and libraries for robot communication, sensor data processing, and control [16]. The simulations were conducted on a notebook with 16GB of RAM and powered by a 2.7 GHz Core i5-5200 processor. The Ground Control Station (GCS) coordinates the UGV operations, specifically using the Nexus robot, which is equipped with various cameras and sensors. The Nexus UGV communicates with the GCS via ROS topics and messages, facilitating real-time data exchange and command execution. Figure 4 shows the simulated environment in Gazebo, which represents a solar farm with photovoltaic panels. The robot Nexus used for the tests is presented in Figure 5.

## III. Results and Discussion

Experiments were conducted by deploying the Nexus robot in the Gazebo-simulated solar photovoltaic farm developed environmentas presented in Figure 4, with an occupancy grid

2025 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Funchal, Portugal – April 2-3, 2025

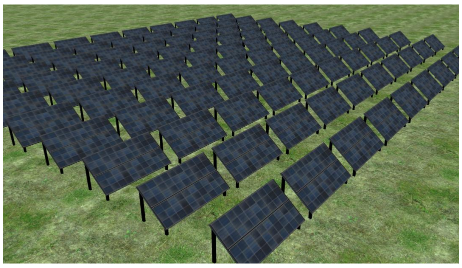


Fig. 4: World created in the Gazebo Software containing the solar panels.

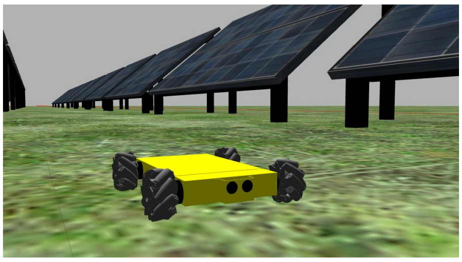


Fig. 5: Nexus robot.

map acquired through handheld telemetry and SLAM algorithms. Figure 6 illustrates this map, showcasing the acquired spatial representation.

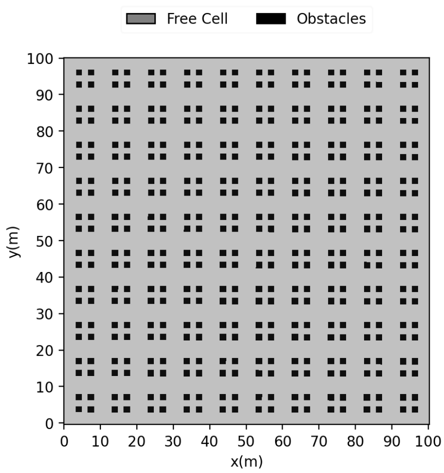
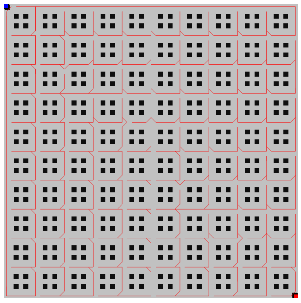
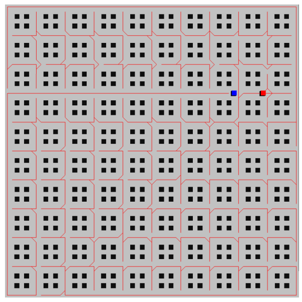



Fig. 6: Occupancy grid for gazebo-simulated solar photovoltaic farm.

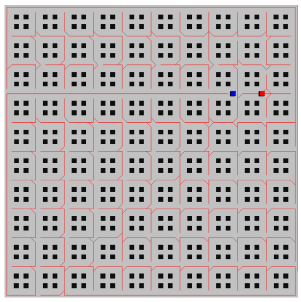
Four algorithms were employed in the experiments: a classic wavefront implementation with backtracking strategies, a wavefront combined with A*, our proposed methodology (wavefront with BA*), and a random walk algorithm. For each algorithm, random samples of start and goal points in the free space $\mathcal{A}_{free}$ were used. The objective was to generate a path that covered 100% of the grid in less than one minute. If the planning time exceeded this limit, the path terminated at the last available calculated cell. A total of two thousand simulations were performed, with each algorithm evaluated on five hundred unique start and goal pairs. To compare our proposed methodology against the other algorithms, three key metrics were selected: (i) the number of instances where the path revisited a previously visited cell; (ii) The number of times the path required the agent to change orientation; and (iii) the duration is taken to compute the path. These metrics are crucial in autonomous coverage path planning scenarios, where optimizing battery usage and computational efficiency are essential. In Figure 7, it is possible to compare the path generated for the proposed solution with other algorithms, such as pure Wavefront, hybrid Wavefront and A*, and the Random Walk algorithm. As expected, all algorithms can cover all areas of the grid.



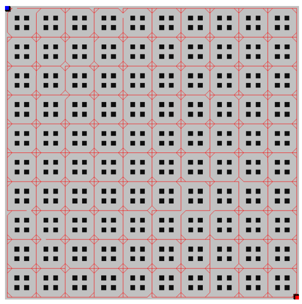
(a) Wavefront Classic



(b) Wavefront + A*



(c) Proposed Method (Wavefront + BA*)

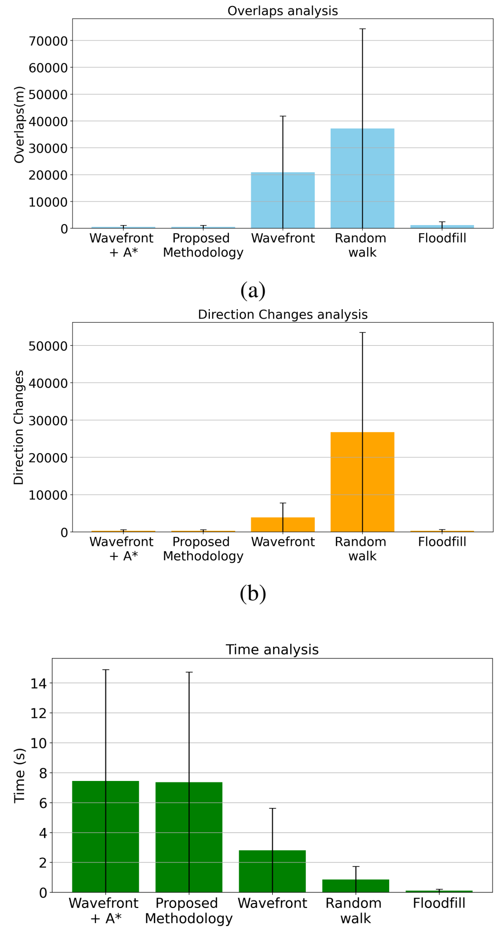


(d) Random Walk

Fig. 7: Generated Path by Four Algorithms: (a) Wavefront Classic, (b) Wavefront with A*, (c) Proposed Method (Wavefront with BA*), (d) Random Walk.

Figure 8 provides a comparative analysis that clarifies the performance differences among the algorithms. Table I presents a quantitative analysis comparing our proposed methodology with the other algorithms. It can be observed that, during simulations, our proposed methodology performed similarly to the wavefront + A* algorithm in terms of overlaps (-0.13%) and direction changes (-0.33%), but with a slightly faster solution (8.74%). When compared to Wavefront and Random walk, Wavefront + BA* outperforms them significantly in terms of overlap and direction change. When compared to the floodfill algorithm, it outperforms in terms of overlaps (126.28%) and direction changes (6.32%). These results highlight the effectiveness of our approach in generating more efficient and smoother paths.

**2025 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Funchal, Portugal – April 2-3, 2025**




(c)

Fig. 8: Mean analysis: (a) Overlap, (b) Direction changes, (c) Time consumption.

TABLE I: Comparison with other methods.

| Method | Overlaps(%) | Direction Changes(%) | Time(%) |
|---|---|---|---|
| Wavefront + A* | -0.13 | -0.33 | 8.74 |
| Wavefront | 6039.85 | 1875.00 | -32.40 |
| Random walk | 6528.33 | 8382.51 | -85.44 |
| Floodfill | 126.28 | 6.32 | -98.49 |

## IV. Conclusion and Future Work

This research presented a hybrid approach integrating the Wavefront algorithm with bidirectional A* for CPP of AGVs in photovoltaic farms. Experiments conducted in simulated environments demonstrated an improved performance compared to traditional algorithms. The outcomes showed enhanced coverage efficiency, reduced execution time, and minimized directional changes, highlighting the robustness and scalability of the proposed approach and validating its potential to optimize operations in complex and dynamic environments. In future works, it is intended to deploy the proposed algorithm in real photovoltaic farms with a real robot to validate its effectiveness under practical conditions.

## V. Acknowledgements

This work was supported by national funds through FCT/MCTES (PIDDAC): CeDRI, UIDB/05757/2020 (DOI: 10.54499/UIDB/05757/2020) and UIDP/05757/2020 (DOI: 10.54499/UIDB/05757/2020); SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020) The authors would like to thank the following Brazilian Federal Agencies, CEFET/RJ, FAPERJ, CNPq, INESC P&D Brasil, and IPB for supporting this research. This work was carried out with the support of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## References

1 Maps, R. (2023) 2023 global solar report. [Online]. Available: https://raptormaps.com/resources/2023-global-solar-report#download

2 Vega Díaz, J. J., Vlaminck, M., Lefkaditis, D., Orjuela Vargas, S. A., and Luong, H., "Solar panel detection within complex backgrounds using thermal images acquired by uavs," *Sensors*, vol. 20, no. 21, p. 6219, 2020.

3 Jaffery, Z. A., Dubey, A. K., Haque, A. *et al.*, "Scheme for predictive fault diagnosis in photo-voltaic modules using thermal imaging," *Infrared Physics & Technology*, vol. 83, pp. 182–187, 2017.

4 Herrmann, W., Eder, G., Farnung, B., Friesen, G., Köntges, M., Kubicek, B., Kunz, O., Liu, H., Parlevliet, D., Tsanakas, I. *et al.*, "Qualification of photovoltaic (pv) power plants using mobile test equipment," *IEA-PVPS T13-24: 2021*, 2021.

5 Galceran, E. and Carreras, M., "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

6 Arkin, E. M. and Hassin, R., "Approximation algorithms for the geometric covering salesman problem," *Discrete Applied Mathematics*, vol. 55, no. 3, pp. 197–218, 1994.

7 Castro, G. G. d., Berger, G. S., Cantieri, A., Teixeira, M., Lima, J., Pereira, A. I., and Pinto, M. F., "Adaptive path planning for fusing rapidly exploring random trees and deep reinforcement learning in an agriculture dynamic environment uavs," *Agriculture*, vol. 13, no. 2, p. 354, 2023.

8 Biundini, I. Z., Pinto, M. F., Melo, A. G., Marcato, A. L., Honório, L. M., and Aguiar, M. J., "A framework for coverage path planning optimization based on point cloud for structural inspection," *Sensors*, vol. 21, no. 2, p. 570, 2021.

9 Biundini, I. Z., Melo, A. G., Pinto, M. F., Marins, G. M., Marcato, A. L., and Honorio, L. M., "Coverage path planning optimization for slopes and dams inspection," in *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 2*. Springer, 2020, pp. 513–523.

10 Zelinsky, A., Jarvis, R. A., Byrne, J., Yuta, S. *et al.*, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proceedings of international conference on advanced robotics*, vol. 13. Citeseer, 1993, pp. 533–538.

11 Shivashankar, V., Jain, R., Kuter, U., and Nau, D., "Real-time planning for covering an initially-unknown spatial environment," in *Twenty-Fourth International FLAIRS Conference*, 2011.

12 Tobias Rainer Schafle, M. M. and Uchiyama, N., "Generation of optimal coverage paths for mobile robots using hybrid genetic algorithm," *Journal of Robotics and Mechatronics*, vol. 33, pp. 11–23, 2021.

13 Ghai, B. and Shukla, A., "Wave front method based path planning algorithm for mobile robots," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*, Satapathy, S. C. and Das, S., Eds. Springer International Publishing, 2016, pp. 279–286.

14 Zidane, I. M. and Ibrahim, K., "Wavefront and a-star algorithms for mobile robot path planning," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017*, Hassanien, A. E., Shaalan, K., Gaber, T., and Tolba, M. F., Eds. Cham: Springer International Publishing, 2018, pp. 69–80.

15 Koenig, N. and Howard, A., "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

16 Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org