Graph-Theoretic Models and Comparative Evaluations of Novel Multi-Robot Path Planning Algorithms for Collision Avoidance and Navigation Optimisation

ALWAFI, Fatma A.S., SAATCHI, Reza <http://orcid.org/0000-0002-2266-0187>, XU, Xu and ALBOUL, Lyuba <http://orcid.org/0000-0001-9605-7228>

applied sciences

MDPI

Article

Graph-Theoretic Models and Comparative Evaluations of Novel Multi-Robot Path Planning Algorithms for Collision Avoidance and Navigation Optimisation

Fatma A. S. Alwafi [1], Reza Saatchi [1,*], Xu Xu [2] and Lyuba Alboul [1,†]

[1] School of Engineering and Built Environment, Sheffield Hallam University, Howard Street, Sheffield S1 1WB, UK; fatmaalwafi@yahoo.com (F.A.S.A.)
[2] School of Computer Science, The University of Sheffield, Western Bank, Sheffield S10 2TN, UK; xu.xu@sheffield.ac.uk
* Correspondence: r.saatchi@shu.ac.uk
† Deceased author.

Abstract

A comprehensive analysis of three graph-theoretic path planning algorithms designed for multi-robotic systems (MRS) was undertaken. The algorithms were the multi-robot path planning algorithm (MRPP), central algorithm (CA), and the optimisation central algorithm (OCA). The primary objective of these algorithms is to enhance path optimality, mitigate computational complexity, and ensure robust inter-robot collision avoidance. The MRPP is a composite approach integrating the visibility graph (VG) for path generation. The CA, derived from VG principles, utilises a central baseline (CB) approach to reduce vertex count, thereby decreasing computational cost while maintaining path efficiency. The OCA extends CA by integrating obstacle expansion and safety margins to enhance collision avoidance and path optimisation. Comparative analysis through simulations in 2D polygonal environments compared the performance of these algorithms, considering their computational efficiency, path optimisation, and collision avoidance. CA and OCA demonstrated significant improvement over the VG-based approach, especially concerning optimality and optimisation. CA reduced the average path length by 4.3% compared with MRPP, while OCA achieved a 6.8% reduction over MRPP, and 2.5% over CA, demonstrating its superior balance between optimality and efficiency. MRPP offers robust connectivity, making it preferable in scenarios where communication is critical. The study's findings assist in devising MPRPP solutions.

Keywords: multi-robot path planning algorithms; visibility graph; graph Laplacian; Dijkstra's algorithm; central algorithm; optimisation central algorithm; multi-robotic collision avoidance, robotics navigation

Received: 27 January 2026
Revised: 6 March 2026
Accepted: 11 March 2026
Published: 15 March 2026

1. Introduction

Multi-Robot Systems (MRS) have emerged as a cornerstone of modern autonomous technologies, enabling coordinated and distributed operations across the industrial, environmental, and service sectors. Compared with single-robotic environments, robot teams can have superior performance, robustness, and fault tolerance, particularly in applications such as surveillance, logistics, and exploration [1,2]. Robotic path planning and navigation in complex environments pose significant challenges. These necessitate advanced

navigation approaches that balance optimality, efficiency, and scalability, and algorithms that generate collision-free paths while maintaining inter-robot communication and coordination [1,3]. The realisation of this balance is crucial for achieving the full potential of cooperative robotics in complex, real-world applications [2,4]. Path planning is a core operation in MRS. It involves determining an optimal trajectory from the robot's start position to its destination while avoiding static and dynamic obstacles [2,5]. The diverse families of path planning techniques and graph-theoretic roadmap methods, notably the Visibility Graph (VG) and Voronoi Diagram (VD) methods, remain foundational for navigation in polygonal 2D workspaces [3,5,6]. The VG approach connects all pairs of vertices that are mutually visible within a polygonal workspace, producing the shortest Euclidean paths [3]. Although this property guarantees global optimality in terms of path length, it also restricts the robots from operating with narrow clearance margins, thus leading to potentially unsafe navigation [4]. Conversely, the VD approach constructs paths that are equidistant from nearby obstacles, prioritising safety over absolute optimality [3,5]. These approaches provide explicit geometric structure, enabling efficient computation of paths using graph-search algorithms such as Dijkstra's and A*. Among the most reliable methods is Dijkstra's algorithm, which iteratively determines the shortest path between nodes in a weighted graph, where the edge weights denote associated traversal costs [3,4,7]. In multi-robot contexts, the sequential application of Dijkstra's algorithm enables each robot to compute the shortest path under the current graph constraints, including reserved or re-weighted edges representing previously planned robots. This approach guarantees individual path optimality and completeness in static environments with a fixed graph representation. However, since the method follows a decoupled planning strategy, optimality is ensured only at the individual robot level. It does not guarantee the global optimality of the entire multi-robot system with respect to collective performance metrics such as total path length or overall coordination cost, which would require joint optimisation over the composite state space [3,8]. However, algorithms such as A* and D* accelerate path planning by guiding the search toward the robot's destination using heuristic estimates [3]. When an admissible and consistent heuristic is employed, A* guarantees optimality equivalent to Dijkstra's algorithm while typically reducing the number of explored nodes [7]. Therefore, a heuristic-based search does not inherently compromise shortest-path optimality.

In this study, Dijkstra's algorithm was selected instead of A* to maintain a heuristic-independent baseline within the visibility graph framework. Since all evaluated algorithms (MRPP, CA, and OCA) share the same underlying graph representation, using Dijkstra ensured that performance differences arose solely from coordination strategy and obstacle treatment, rather than from heuristic guidance. This choice enabled a controlled and consistent comparative analysis. Accordingly, the multi-robot motion planning problem is formulated as the computation of the shortest and most collision-aware paths on a structured graph representation, ensuring individual path optimality under the given constraints [1,3,7].

Graph-based algorithms, particularly the VG algorithm, have been widely adopted due to their ability to yield globally optimal paths. However, VG's computational expense and sensitivity to obstacle density have proved challenging, as its derived paths often pass very close to obstacles. This compromises safety margins during physical implementation [5,7,9]. To mitigate this, spectral graph theory has been incorporated into planning frameworks, introducing the algebraic connectivity (λ_2) of the graph Laplacian as a quantitative measure of network robustness and inter-robot coordination [10–12]. This metric provides a valuable basis for maintaining network connectivity while guiding sequential path planning among robots. Recent developments in multi-robot planning have explored the integration of geometric, spectral, and optimisation-based methods to achieve a balance of

safety, computational efficiency, and path quality [12,13]. Building upon these foundations, this article provides a comprehensive analysis of three graph-based multi-robot path planning algorithms, i.e., MRPP [3,10], CA, and OCA [3,14]. Each algorithm represents a different operational philosophy and trade-off among optimality, connectivity preservation, and safety, and all aim to enhance path planning efficiency while retaining VG's optimality properties. MRPP integrates VG construction, λ_2-based sequencing, and Dijkstra's algorithm to prioritise both global optimality and network coherence throughout the mission. CA introduces the central baseline (CB) concept to simplify obstacle representation and reduce computational complexity. OCA extends CA, incorporating adaptive weighting and obstacle expansion techniques to enhance safety and maintain feasible trajectories. Each algorithm reflects a different balance between path optimality, computational efficiency, and safety, making it suitable for a specific operational scenario in robotics.

Recent survey studies, including Bui (2023) [9], have emphasised the growing need for hybrid geometric spectral approaches that integrate classical roadmap-based planning with connectivity-aware coordination metrics in multi-robot systems (MRS). In particular, the survey highlighted the limited number of unified comparative evaluations that systematically analyse how geometric shortest-path methods could be combined with spectral graph metrics to balance optimality, safety, and coordination performance.

The present work directly addresses this important gap by integrating visibility graph (VG)-based shortest-path planning (VG + Dijkstra) with algebraic connectivity (λ_2) for coordination-aware sequencing, and by comparatively evaluating three structured variants (MRPP, CA, and OCA) under a consistent graph-theoretic framework. Although this study focuses on static environments, it significantly contributes toward the hybrid geometric–spectral direction identified in [9] and establishes a foundation for future extensions to dynamic multi-robot systems. Therefore, this article contributes by providing a unified and in-depth analysis of three graph-based multi-robot path planning algorithms (MRPP, CA, OCA), outlining their algorithmic principles and operational differences. A rigorous comparative evaluations of the three algorithms were undertaken, demonstrating the trade-offs between their connectivity, computational efficiency, and collision avoidance. The study presents the first systematic integration of VG geometry, algebraic connectivity, and safety-weighted optimisation within a unified comparative evaluation framework. The results demonstrated that λ_2-based sequencing effectively preserves inter-robot connectivity, a capability at which the MRPP excels. Furthermore, the influence of the central baseline (CB) on path quality and computational efficiency was rigorously analysed, revealing that the CA achieves a significant reduction in runtime through targeted graph simplification. Finally, a safety-distance obstacle expansion model was introduced to enhance collision avoidance, enabling the OCA to outperform both MRPP and CA by jointly achieving improved safety margins and path efficiency.

The selection of MRPP, CA, and OCA was driven by the objective of evaluating structured improvements within roadmap-based, graph-theoretic multi-robot path planning. These algorithms share a common VG foundation but differ in coordination strategy and obstacle treatment. MRPP incorporates algebraic connectivity (λ_2) for sequencing, CA introduces obstacle reduction for computational efficiency, and OCA enhances safety through obstacle expansion. This progression enables a controlled comparative analysis within a consistent methodological framework, allowing performance trade-offs to be examined without introducing paradigm-level variability.

The remainder of this article is organised as follows: In Section 2, the related literature on roadmap-based planning and algebraic graph theory is reviewed. Section 3 details the designs and operations of the MRPP, CA and OCA. Section 4 presents the results, simulation configurations, and comparative performance analyses. Section 5 discusses the

findings, their implications, and future work, and Section 6 provides the conclusions. To ensure completeness of the article and ease of adaptation of the algorithms by the readers, the operations of the three MRPPs are included in Appendix A.

2. Literature Review

In this section, the related literature is reviewed.

2.1. Path Planning Workspace Modelling

Path planners for robots require detailed information describing the operational environment, including the obstacles' locations and workspace's characteristics (i.e., free space versus restricted areas). These influence the robot's mobility through the traversability, costs, and weights associated with regions, representing penalties for risks, energy use, or uncertainty. Path planning considerations depend on how robots interact with their environment [3,15]. In many cases, it is sufficient to minimise the path length while distinguishing traversable from non-traversable regions; however, each robot's steering, dynamics, and motion constraints must also be considered [15–17].

2.1.1. Workspace Representation and Modelling

Workspace modelling and representation are important in robotic path planning because they directly influence the feasibility and efficiency of computed trajectories. Two levels of representation are typically considered: (i) the geometric workspace, which describes physical boundaries and obstacle geometry, and (ii) the configuration space (C-space), which embeds the robot's size, shape, and kinematic constraints [3,15,16]. Some path planning algorithms require the construction of a graph that represents the robot's environment. Graph-based representations have long served as the foundation for robotic motion planning because they offer a structured and mathematically tractable framework for modelling spatial representations between free workspaces. This is characteristic of graph-search strategies within the configuration-space (C-space) search category [15,18]. In C-space modelling, obstacles in the geometric workspace are expanded by the robot's geometry to form C-space obstacles. This simplifies collision checking by allowing feasibility to be evaluated using the robot's reference point [15,19–22]. The resulting free configuration space is defined as $W_{free} = R^2 \setminus \bigcup_i O_i$, where W_{free} represents all points not occupied by the obstacles, R^2 denotes the two-dimensional Euclidean plane representing the robots' possible positions, and $\bigcup_i O_i$ is the union of all obstacle regions [3,10].

2.1.2. Roadmap-Based Planning

Path planning in robotics involves determining an efficient, collision-free path from start to goal positions while navigating an environment that may include static or dynamic obstacles. Various algorithmic strategies have been developed to achieve this task [10,21]. In roadmap-based approaches, the environment is abstracted into a graph $G = (V, E)$, where the vertices (V) represent reachable waypoints and the edges (E) denote feasible, traversable paths between them. Each vertex represents a feasible robot state, while each edge denotes a valid transition between states [10,15,23–25]. This representation captures both the structure of the configuration space and the allowable robot motions, ensuring that any resulting path is kinematically feasible when using graph-search algorithms. Roadmap-based planning is especially well-suited to environments with polygonal obstacles, where geometric relationships can be exploited to reduce computational complexity. Then, path planning becomes a graph-search problem, in which the goal is to determine the sequence of edges minimising a cost function, typically distance or time [10,18,22–25].

Depending on the planning strategy, the workspace may be represented continuously or discretised into a graph structure such as a visibility graph (VG). Continuous models preserve geometric fidelity, while graph-based models enable efficient searching using algorithms such as Dijkstra's. The algorithms presented in this article leverage roadmap structures to generate efficient, collision-free paths while integrating multi-robot connectivity constraints.

2.2. Graph-Theoretic Foundations for Path Planning Algorithms

Graph-theoretic representations form the basis for many classical and modern path planning algorithms in robotics. By abstracting the workspace into a set of discrete states and transitions, graphs provide a mathematically structured framework for reasoning about connectivity, optimality, and obstacle avoidance. In addition, graph-theoretic approaches offer several critical benefits that make them highly suitable for solving coordination, communication, and planning problems in Multi-Robot Systems (MRS) [3,10,15,22–25]. The advantages of Graph-Theoretic Planning in MRS include:

- Modularity: One of the primary strengths of graph-theoretic models lies in their modular structure. Robots, environments, and tasks can be independently represented and then integrated into a composite system graph, supporting flexible adaptation and reusability [3,6,13,26].
- Optimality: Classical algorithms such as Dijkstra's guarantee the shortest path solutions in weighted graphs. This is essential for mission-critical operations in static environments [25,27,28].
- Scalability: Spectral graph theory enables performance and robustness analysis in large-scale MRS using tools like the Laplacian matrix and algebraic connectivity, supporting scalable control and communication frameworks [11,26,29–31].
- Extensibility: Graph-theoretic frameworks are inherently compatible with distributed artificial intelligence (AI) paradigms, including consensus algorithms, reinforcement learning, and network control. This extensibility supports decentralised decision-making, dynamic task allocation, and formation control across heterogeneous robot teams [32–36].

These properties collectively highlight graph theory as a versatile and powerful backbone for the design, analysis, and deployment of multi-robot systems across diverse environments. The following sections provide a review of the main graph-theoretic concepts relevant to the algorithms developed in this study, including visibility graphs, shortest-path search, and connectivity measures [3,10,15,22].

2.2.1. Visibility Graphs

A full VG is generated for the environment, including: all polygonal obstacle vertices, the start and goal positions of each robot R_i, and additional intermediate vertices where direct line-of-sight is available. Each robot's start (s_i) and goal (g_i) are included as vertices in the VG. Visibility edges $e_i = (v_i, v_j)$ are constructed whenever the line segment ($(v_i v_j)$, i.e., distance between i and j) does not intersect any obstacle [10,16,17].

Construction of the Visibility Graph

The VG method used for multi-robot path planning is described as an undirected weighted graph $G = (V, E, W_E)$, where $V = \{v_1, v_2, \ldots . v_n\}$, is the set of vertices representing the configurations of the robots, as well as the starting points and endpoints of the robots' movements. V is defined as the set of all ordered pairs, (v_i, v_j), where both v_i and v_j are the elements of V, where $\{(v_i, v_j), v_i, v_j \in V\}$. $E = \{e_1, e_2, \ldots, e_n\}$, is a set of edges representing the paths between the vertices. E is a subset of all possible ordered pairs of

vertices from V [10,24]. Each edge $e_i = (v_i, v_j)$ in E represents a connection from vertex v_i to vertex v_j. This framework is fundamental to defining the structure of directed graphs; where the direction of the edge is significant, $e_{ij}, i \neq j$, exists between the vertices if robot n (i.e., R_n) interacts with robot m (i.e., R_m). This means two robots can communicate only if they are within communication distance of each other. In addition, the presence of the edge e_{ij} refers to the presence of the edge e_{ji}. Therefore, $e_{ij} = e_{ji}$ signifies that the edge is mutual and directionless. This characteristic is fundamental to undirected graphs, where the edges do not have a specific direction. W_E is a function that assigns weights (i.e., path length) to each edge in E. This notation is valuable in problems involving weighted graphs, such as finding the shortest path [3,10,18,19]. Edges join all pairs of mutually visible nodes and the edges of obstacles [10,24]. $W_E = \{w_{ij} | (i,j) \in V\}$, $w_{ij} = 0$, if $(i,j) \notin E, and\ w_{ij} > 0$ otherwise. If we consider a team of n robots, the set of neighbours of the ith robot can be defined as $n_i = \{j \in V, j \neq i | e_{i,j} \in E\}$, representing all the robots that can communicate with it. Hence, each robot is assumed to be able to interchange data with its neighbours [10,23,24]. VG constructs edges between two vertices when there is a direct line of sight between them, meaning that the line connecting the vertices does not intersect any obstacle in the environment. This guarantees optimal path lengths that are obstacle-free [20,24]. The weights of an edge represent the Euclidean distances between its vertices [21,24].

2.2.2. Shortest Path Search: Dijkstra's Algorithm

Once the VG is constructed, path planning can be formulated as a shortest-path problem [25,37]. Dijkstra's algorithm efficiently computes the shortest paths based on the weighted edges [3,10,22,24]. Given a graph $G = (V, E)$ and a weighted function $w_{i,i+1}$, Dijkstra's algorithm finds the minimum cost sequence of vertices connecting a start vertex s to a goal vertex g as

$$\widehat{W} = \arg\ min_{P=(w_0,\ldots,w_n)} \sum_{i=1}^{n} w_{w_i w_{i+1}}, and \begin{cases} w_0 = s \\ w_n = g \\ w_i w_{i+1} \in E \end{cases} \quad (1)$$

The term arg refers to the argument (of path P) that provides this minimum value, and the min term refers to the minimum value of the cost. The optimal path $\widehat{W}$ is the sequence of waypoints $P = (w_0, w_1, \ldots, w_n)$ that minimises the total distances of all edges along the path, where w_i is waypoint i. The term $w_{w_i w_{i+1}}$ is the weight (distance) between consecutive waypoints w_i and w_{i+1}. n is the number of edges along the path. In this work, Dijkstra's algorithm was used for the three path planning algorithms (i.e., MRPP, CA, and OCA) to compute the robots' trajectories across reduced or full visibility graphs.

2.2.3. Algebraic Connectivity and Laplacian Graph in Multi-Robot Systems

Recent advances in spectral graph theory have introduced tools to analyse and optimise network connectivity in multi-robot systems (MRS). The Laplacian graph, $L = D - A, L \in R^{n \times n}$ is the weighted matrix which combines the diagonal matrix D and the adjacency matrix A. It captures the essential topological properties of a communication or visibility graph. In an adjacency matrix in a weighted graph, $\in \mathbb{R}^{nxn}$ is defined as

$$A_{ij} = w_{ij} = \begin{cases} a_{ij} > 0\ if\ j \in N_i \\ 0\ otherwise \end{cases} \quad (2)$$

where w_{ij} represents the weight on the edge between nodes i and j, such as Euclidean distance or communication cost, and $w_{ij} = w_{ji}$ signifies that the edge is mutual and directionless, element a_{ij} is defined as the positive edge weight between robots i and j when a direct connection exists, and it is zero otherwise. The neighbour N_i consists of all

nodes that share an edge with node i. It is defined as the set of nodes j for which the edge (i, j) belongs to the edge set E of the graph, $N_i = \{ j \mid (i,j) \in E \}$ [15]. D is a diagonal matrix with each entry d_{ii} equals the sum of the weights of edges incident on node i, i.e.,

$$d_{ii} = \sum_{i=1}^{n} a_{ij} \quad (3)$$

The weighted diagonal matrix plays a fundamental role in distributed coordination and consensus-based algorithms, as it directly contributes to the construction of the Laplacian graph and the evaluation of network connectivity [18].

The eigenvalues of the Laplacian matrix are λ_i of L where $\lambda_i = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. The manner in which the graph Laplacian is defined is that the smallest eigenvalue, λ_1, is always zero. The key reason is that each row of L sums to zero. Consequently, this implies that the all-ones vector, $1 = [1,1, \ldots,1]^T$, is the eigenvector of L associated with the eigenvalue 0, such that $L_1 = 0$, hence, $\lambda_1 = 0$. The zero eigenvalue corresponds to a constant eigenvector across the graph. This reflects the property that the Laplacian operator measures the differences between adjacent nodes; if all node values are equal, there is no difference, so the result is zero.

Connectivity insight: The multiplicity of the zero eigenvalue equals the number of connected components in the graph. If the graph is connected, it has exactly one zero eigenvalue. If the graph consists of k disconnected components, then its Laplacian has k zero eigenvalues (i.e., the multiplicity of the zero eigenvalue is equal to k). In multi-robot networks, for a connected communication graph G:$0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$, and λ_2 (the algebraic connectivity) indicates how well the robot team is connected. So, $\lambda_1 = 0$ arises naturally from the Laplacian's structure and represents the graph's baseline connectivity. The second smallest eigenvalue of L, known as the algebraic connectivity (λ_2) measures how well connected the graph is [3,10,26–29]. A larger λ_2 implies robustness, connectivity, and resilience to link failures, i.e., a more robust, well-connected graph with many edges. The value of λ_2 ranges between 0 and the number of vertices (N), and the connectivity refers to the number of vertices in the graph if the graph is completely connected. Thus, the maximum value of $\lambda_2 = N$, and it is obtained when the entries (i, j) of the adjacent matrix (A) are all equal to 1, which means that all possible edges are present in it [10,28]. Recent research has extended these principles to multi-robot domains, introducing connectivity metrics such as algebraic connectivity (λ_2) to enhance coordination. These studies have explored hybrid approaches integrating VG with optimisation heuristics and graph theory metrics to achieve better trade-offs between safety, computational speed, and scalability [10,27,29].

In MRS coordination and path planning, maintaining a larger value for λ_2 ensures that all robots remain within communication range, thus allowing for cooperative behaviour such as formation maintenance and dynamic task allocation [10,26,38]. The algorithms that explicitly incorporate λ_2, such as MRPP, dynamically adjust planning order and edge weighting based on real-time connectivity metrics, achieving a balance between global optimality and network coherence [12,30–36]. Studies (e.g., [3,8,10]) have demonstrated that coupling geometric path planning with spectral measures significantly enhances both safety and computational efficiency, particularly in dense environments with inter-robot communication constraints [10,32–37].

3. Materials and Methods

The workspace was modelled using a metric map, providing the geometric accuracy required for constructing visibility graphs and computing Euclidean distances between waypoints. This modelling framework also supports higher-level analysis, including algebraic connectivity evaluation in multi-robot systems. Overall, accurate workspace modelling ensures that the planned trajectories are collision-free, respect robot motion

constraints, and provide the spatial fidelity needed for multi-robot coordination. All algorithms were formulated upon the VG framework and used Dijkstra's algorithm as the core path-search mechanism. The distinction among them lies in the graph simplification, sequencing, and optimisation strategies applied. The main objective of optimisation is minimising the path length (the total distance travelled by the robots) while maintaining a minimum level of connectivity in the communication graph, generating optimal and collision-free paths from source to goal destinations.

To consider collision and connectivity constraints, we considered two collision types: (i) collisions between the robots and obstacles, and (ii) inter-robot collisions (i.e., collisions between multiple robots). Each robot could determine the presence of an obstacle and measure its relative location and the distance from its boundary within the communication range. Therefore, the aim was to solve the problem of a team of multiple robots that began from an initial configuration where the team was connected (i.e., $\lambda_2 > 0$), maintaining connectivity while avoiding collisions until reaching their target positions. A collision avoidance mechanism was executed to prevent the robots from colliding with each other. Communication was defined based on the weights of the edges that determined the quality of the communication links between robots. Connectivity was preserved when λ_2 remained non-zero while each robot tracked its planned path toward its goal location. Within this connectivity-aware collision avoidance framework, the visibility graph (VG) was adopted as the underlying path planning structure. However, despite the advantages of the VG framework, certain limitations motivated further refinement.

MRPPs utilise VG because they are well-established in 2D environments and can produce shortest-path trajectories between polygonal obstacles. However, they have two critical limitations:

- They may result in paths too close to obstacles.
- Their computational cost rises with environmental complexity, making them inefficient in cluttered environments.

To address these limitations, enhanced variants were introduced within the same framework:

- The central algorithm (CA) utilised the central baseline (CB) to reduce the number of vertices and selectively considered only intersecting obstacles.
- The optimisation of the central algorithm (OCA) further enhanced safety by incorporating obstacle expansion based on safety margins.

3.1. Simulation and Comparative Evaluation

This section details the simulation framework, experimental parameters, and performance metrics used to evaluate the effectiveness of the proposed MRPP, CA, and OCAs. The simulations were designed to test each algorithm's ability to produce collision-free, computationally efficient, and scalable trajectories for multiple robots operating in static 2D polygonal environments. A series of simulations was implemented in MATLAB R2024a, utilising a custom-developed Graphical User Interface (GUI) that supported interactive workspace generation, robot deployment, and result visualisation [39–41].

3.2. Simulation Environment and Setup

To evaluate the performance of the three algorithms (MRPP, CA, and OCA), the test environment consisted of 2D polygonal maps, populated with irregular obstacles, sparse and dense. The obstacles were randomly generated to ensure variability and to challenge the algorithms' adaptability. A team of five homogeneous robots (R_1–R_5) was deployed in each simulation, modelled as small points with identical communication and actuation constraints. The robots' initial positions and goal locations were predefined and spaced to

ensure feasible, non-trivial planning scenarios. Each robot was assumed to have limited communication with its neighbouring robots within a specified range, supporting inter-robot coordination based on the neighbourhood graph. The obstacles were treated as non-traversable polygonal regions, and the algorithms were tasked with avoiding collisions while maintaining visibility and graph connectivity when required. The primary objective for each algorithm was to generate a set of collision-free paths from the initial to goal positions while minimising total path length and computational time.

3.3. Performance Metrics of the Algorithms

Each experiment assumed complete a priori knowledge of the environment, including the obstacle boundaries, start (s_i) and goal (g_i) positions of the robots, and their geometric configurations. Figure 1 is a scenario in a workspace environment with an area of 10 m × 9 m, containing 13 static polygonal obstacles ($O_1, O_2, \ldots O_{13}$) of varying sizes. Up to five robots ($R_1, R_2, \ldots, R_5$) were initialised at non-overlapping positions with five distinct target ($g_1, g_2, \ldots, g_5$) locations.

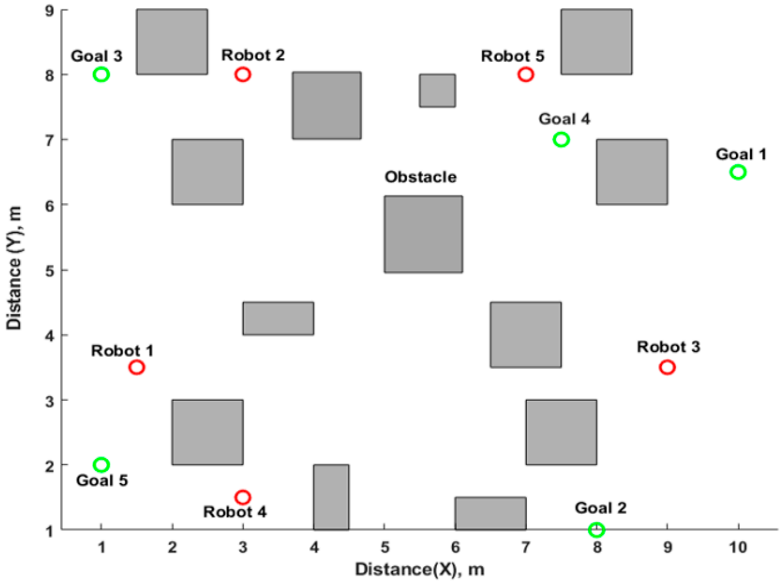

Figure 1. This scenario, in a workspace, consists of thirteen obstacles highlighted in grey, five robots highlighted with red circles, and five goals highlighted as green circles.

The following metrics were used to evaluate the algorithms:

- Path length: Total Euclidean distance from start to goal positions per robot.
- Arrival time: Time to reach the goal, assuming a constant speed.
- Connectivity (λ_2): Algebraic connectivity value posts each planning step (for MRPP).
- Central baseline: Reduce obstacles and generate waypoints (for CA).
- Safety distance: Minimum average clearance from obstacles (for OCA).
- Computation time: Time required to compute a complete set of paths.

4. Results

To illustrate the functionality of MRPP, CA, and OCA in generating optimal and collision-free paths from source to goal destinations, the workspace scenario depicted in Figure 1 was employed. This highlights the algorithms' capabilities in ensuring path efficiency, safety, and effective inter-robot coordination within a structured environment. The scenario of workspace environment measured 10 m × 9 m, containing 10 static polygonal obstacles ($O_1, O_2, \ldots, O_{10}$) of varying sizes represented as grey squares. The robots (R_1, R_2, and R_3) were initialised at non-overlapping positions with distinct three targets

($g_1, g_2,$ and g_3) locations, highlighted as red, green and blue circles, respectively (see Figure 2).

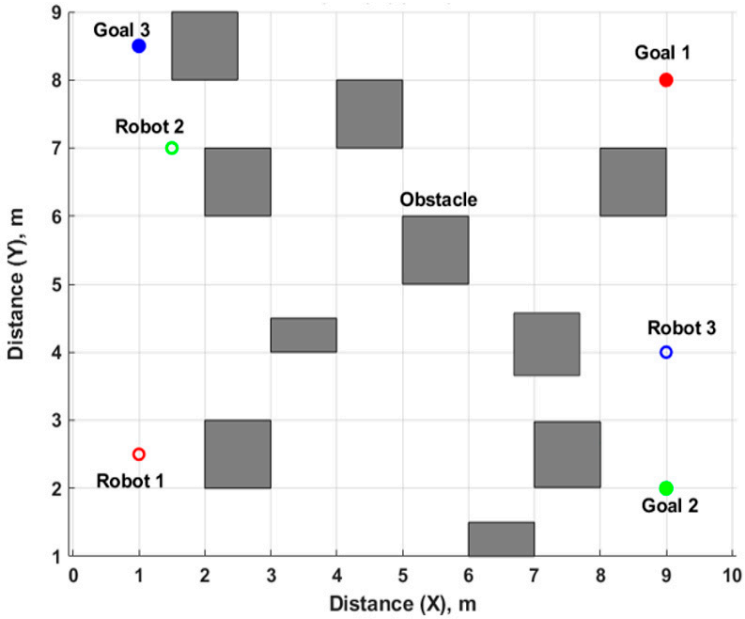

Figure 2. Scenario in a workspace environment indicating the obstacles and robots. The grey squares are the obstacles while the circles are the robots and their associated destination (goals).

Figure 3 shows the application of MRPP to find the shortest paths for three robots and their associated goals, which were determined using Dijkstra's algorithm. The red path is for R_1, the green path is for R_2, and the blue path is for R_3.

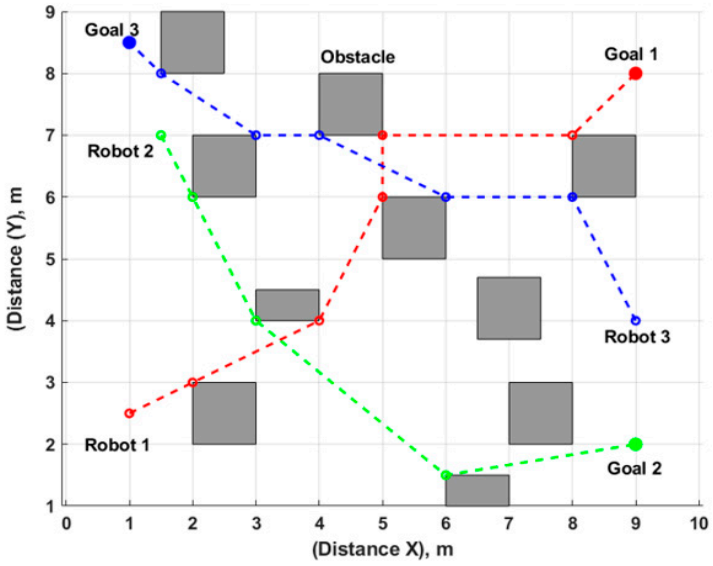

Figure 3. Path planning using MRPP. The paths for robots R1, R2 and R3 are shown in red, green and blue, respectively. The red, green, and blue circles are the robots and their goals, respectively.

Figure 4 illustrates the operation of CA. The red straight lines are CB. The red, green and blue circles are the robots and their goals, respectively. The grey squares represent the obstacles, and the small blue points are the waypoints.

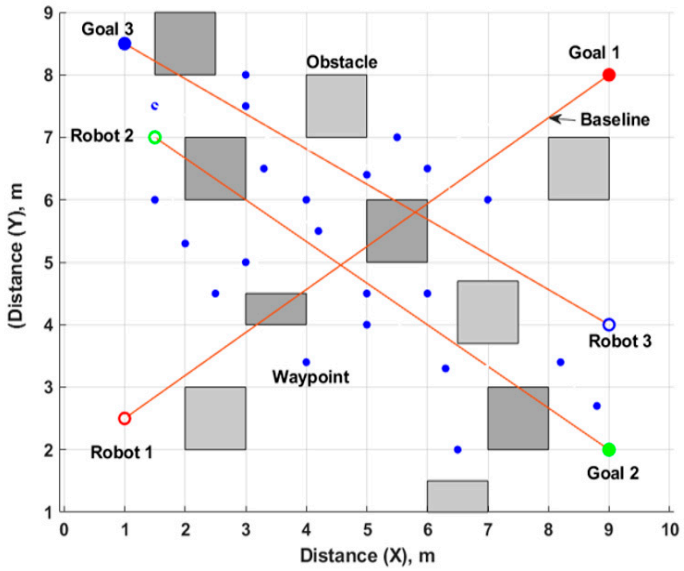

Figure 4. The steps involved in the operation of the central algorithm. The red straight lines are CB. The red, green and blue circles are the robots and their goals, respectively. The grey squares represent the obstacles, and the small blue points are the waypoints.

Figure 5 illustrates the application of CA to find the shortest paths for three robots and their associated goals, determined using Dijkstra's algorithm. The paths for R_1, R_2 and R_3 are shown in red, green and blue, respectively. The red, green and blue circles are the robots and their goals, respectively. The grey squares are the obstacles.

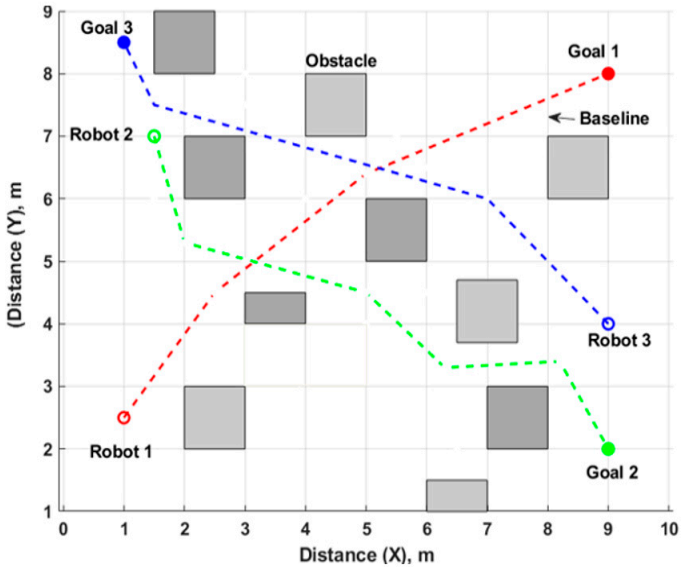

Figure 5. Path planning using CA: The paths for robots R_1, R_2 and R_3 are shown in red, green and blue, respectively.

CA offers a balance between accuracy and speed. Its reduced vertex count enables a faster computation while maintaining near-optimal path length. However, safety near obstacles is not explicitly guaranteed, motivating further improvement via the OCA. Figure 6 shows the application of OCA to find the shortest paths for three robots and the associated goals, which were determined using Dijkstra's algorithm. The paths for robots R_1, R_2 and R_3 are shown in red, green and blue, respectively. The red, green and blue circles are robots and their goals, respectively. The grey square represents the obstacles.

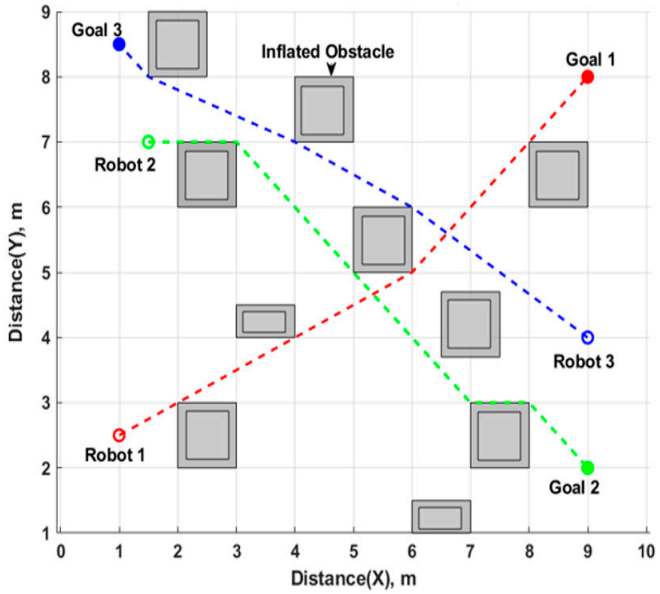

Figure 6. Paths planning using OCA: The paths for robots R_1, R_2 and R_3 are shown in red, green and blue, respectively.

These results indicate that MRPP ensures optimality in terms of distance minimisation via VG combined with Dijkstra's algorithm It also maintains the robustness of network connectivity (highest λ_2) due to its use of the sequential paths ordering, enhancing coordination and reducing potential inter-robot conflicts. Thus, λ_2 contributes to system-level robustness rather than direct path length optimisation. CA and OCA reduce computational load and enhance safety. OCA, in particular, balances path length and safety distance effectively. OCA path maintains a consistent buffer around obstacles while MRPP tends to pass through narrower corridors, and CA's trajectory occasionally approaches obstacle edges. OCA achieves substantial improvements in both computational efficiency and collision avoidance. It provides smoother, safer paths and outperforms traditional MRPPs and CAs under a high obstacle density. Although OCA incorporates obstacle expansion to guarantee a predefined safety margin (δ), the shortest-path computation along the expanded obstacle boundaries may cause the trajectory to appear visually close to obstacles in graphical representations. However, the minimum clearance is strictly enforced in configuration space and verified numerically in simulation results.

4.1. Comparative Analysis

Path length (PL) is defined as the total Euclidean distance travelled by each robot along its assigned path, i.e.,

$$L = \sum_{(i,i+1)\in P_{\min}} w_{i,i+1} = d_{i,i+1} \tag{4}$$

To calculate the average path length for each algorithm, let the number of robots (k) be 3, PL_i^A be the path length of robot i using an algorithm A, where $A \in$ {MRPP,CA,OCA}, then the total path length (PL) is

$$PL_{\text{total}}^A = \sum_{i=1}^{k} PL_i^A \tag{5}$$

and the average path length (APL) is

$$APL^A = \frac{1}{k} \sum_{i=1}^{k} PL_i^A \tag{6}$$

The values for PLs for MRPP, CA and OCA for R_1, R_2 and R_3 are shown in Table 1.

Table 1. Path length for each robot and each algorithm.

Robot	MRPP	CA	OCA
R_1	11.00	9.95	9.83
R_2	10.30	10.16	9.57
R_3	9.98	9.65	9.24
Total	31.29	29.76	28.65
Average	10.43	9.92	9.55

Total performance difference between the algorithms is determined as

$$\text{Percentage difference } = \frac{APL_{\text{A}} - APL_{\text{B}}}{APL_{\text{A}}} \times 100 \quad (7)$$

As illustrated in Table 2, using the average path lengths, the proposed OCA achieves an overall path length reduction of 8.4% compared to MRPP. This improvement is 4.9% when comparing MRPP and CA, followed by a 3.7% reduction when comparing CA and OCA.

Table 2. Path length differences comparing the algorithms.

Algorithm	Percentage Difference
MRPP versus CA	$\frac{10.43-9.92}{10.43} \times 100 = 4.9\%$
MRPP versus OCA	$\frac{10.43-9.55}{10.43} \times 100 = 8.4\%$
CA versus OCA	$\frac{9.92-9.55}{9.92} \times 100 = 3.7\%$

The traverse time for each robot, assuming a constant speed of 1 unit/second is shown in Table 3. It is calculated based on the geometric path length obtained for each robot as $T_i = L_i/\text{S}$ where T_i is the traversal time of robot i, L_i is the total path length, and S is constant robot speed (1 unit/sec). Under this assumption, the traversal time numerically equals the path length. This simplified model allowed a fair comparison of algorithm performance in terms of travel efficiency without introducing dynamic or kinematic variability.

Table 3. The traverse time for each robot and each algorithm.

Robot	Method	Path Length (m)	Arrival Time (Seconds)
R_1	MRPP	11.00	11.00
	CA	9.95	9.95
	OCA	9.83	9.83
R_2	MRPP	10.30	10.30
	CA	10.16	10.16
	OCA	9.57	9.57
R_3	MRPP	9.98	9.98
	CA	9.65	9.65
	OCA	9.24	9.24

Computational efficiency was evaluated qualitatively based on algorithmic complexity and graph size reduction. Since CA and OCA reduce the number of considered obstacle vertices via the Central Baseline (CB) method, they produce smaller graphs compared to the full visibility graph used in MRPP. Given that Dijkstra's algorithm has complexity $O(E\log V)$, reducing the number of vertices and edges directly decreases computational cost. Therefore, CA and OCA are considered computationally more efficient.

The evaluations indicated that OCA determines the path faster than MRPP and provides shorter path lengths as compared to CA. The inclusion of adaptive weighting allows OCA

to manage dense obstacle regions effectively. MRPP remains suitable for scenarios demanding strict global optimality while CA and OCA are preferable for real-time applications requiring computational efficiency.

4.2. Comparative Evaluation of the Algorithms

Table 4 provides a qualitative comparison of the MRPPs, CAs, and OCAs based on criteria including path optimality, collision avoidance, safety margin, and computational efficiency.

Table 4. A qualitative comparison of the MRPP, CA, and OCAs.

Criteria	MRPP	CA	OCA
Collision Avoidance	Good (λ_2 sequence)	Good (Waypoints sequence)	Excellent
Connectivity Awareness	Yes	No	No
Efficiency	Moderate	High	High
Path Optimality	High (VG + λ_2)	High (CB Filtered + waypoints)	High (with Margin (D_s))
Safety Distance from Obstacles	Low	Medium	High (obstacle margin)
Safety Margin	Low	Medium	High
Scalability	Medium	High	High

4.3. MRPP Performance Evaluation

Figure 7 shows the simulation results for the shortest path planning for five robots using the MRPP involving five robots operating within a two-dimensional-based environment.

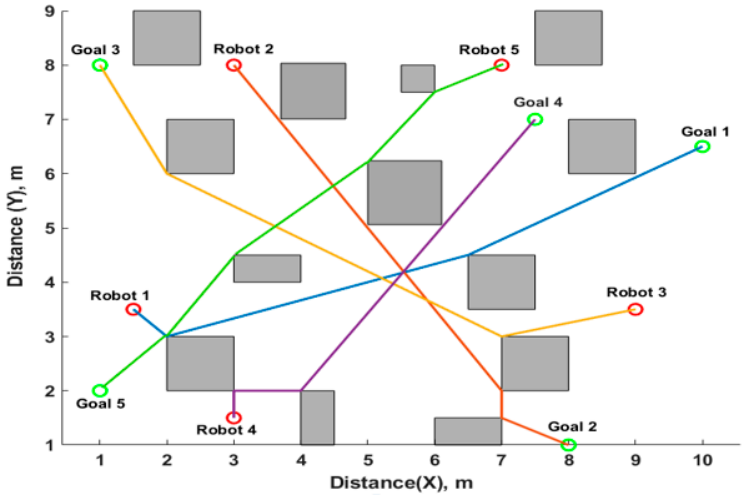

Figure 7. Simulation results of the MRPP demonstrating the computed shortest paths for all robots, with each trajectory distinctly visualised using separate colour assignments (blue, orange, yellow, purple, and green, respectively).

The key objective is to illustrate the MRPP's ability for collision-free and shortest trajectories. The environment contains static obstacles that are denoted by the grey squares distributed across the map. These areas were inaccessible to robots. The five robots are depicted by red circles representing their initial starting positions. Their respective destination targets or goals are highlighted by green circles. The underlying mechanism for

generating the optimal path segment between any two accessible vertices in the grid is Dijkstra's algorithm. This guarantees that the path calculated for each individual robot would be the geometrically the shortest route. The resultant path for each robot is displayed as a solid line, indicating its calculated movement sequence from its starting point (red circle) to its assigned goal (green circle). The overall coordination and execution displayed in this figure are governed by the MRPP. The MRPP layer utilises the shortest paths provided by Dijkstra's algorithm to ensure that all five robots can reach their respective targets without collisions with each other, or with static grey obstacles.

4.4. CA

A typical result of a CA's path planning simulation involving five robots operating within a two-dimensional-based environment is demonstrated in Figure 8. The environment contains static obstacles denoted by the grey squares. These areas are inaccessible to the robots. The five robots depicted by red circles representing their initial starting positions. Their respective destination targets or goals are highlighted by green circles.

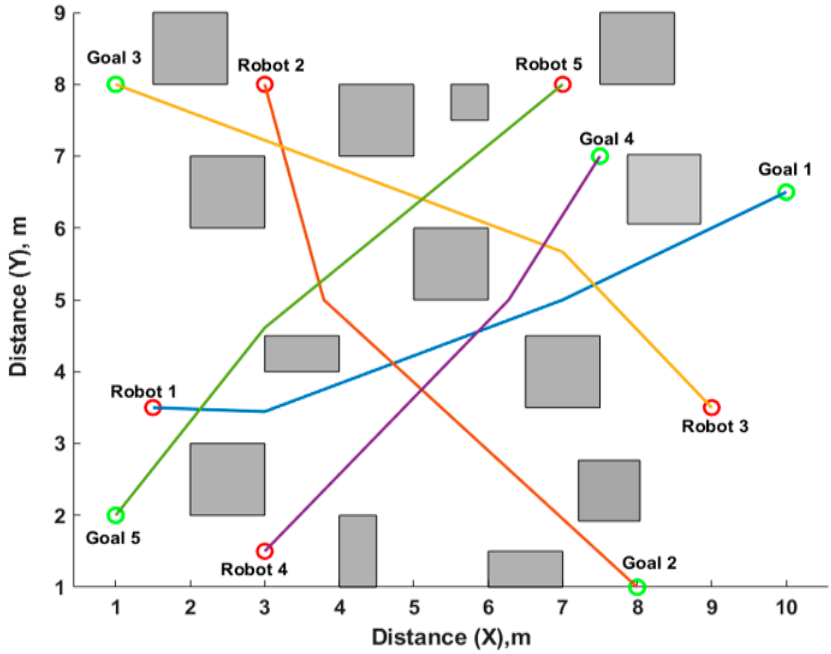

Figure 8. Simulation results of the CA demonstrate the computed shortest paths for all robots, with each trajectory distinctly visualised using separate colour assignments (blue, orange, yellow, purple, and green, respectively).

4.5. OCA

The path planning simulation result utilising the OCA and five robots is demonstrated in Figure 9.

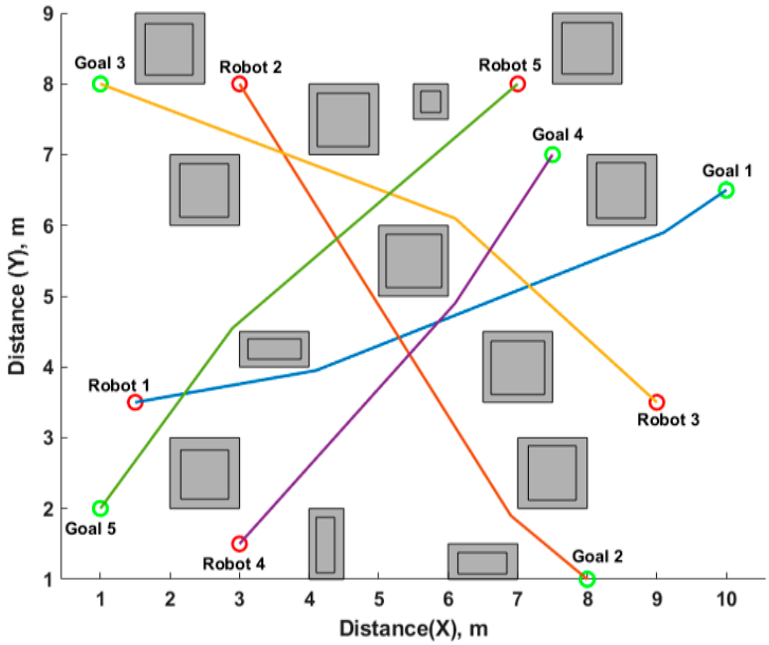

Figure 9. Simulation results from the OCA demonstrating the computed shortest paths for five robots, with each trajectory distinctly visualised using separate colour assignments (blue, orange, yellow, purple, and green).

The environment of the workspace consists of static grey squares that denote inflated, impassable obstacles within the configuration space. The figure implicitly demonstrates a bijection between the initial positions and the target goals, where each robot must successfully reach its assigned terminus. The lines connecting the red start circles to the green goal circles illustrate the final, calculated paths derived from the OCA. The simulation confirms the OCA's effectiveness in achieving reliable, high-density navigation in the workspace environment, where all five robots reach their assigned targets without collision.

4.6. Comparative Analysis of MRPP, CA, and OCAs

The path length comparison metric represents the total distance each robot travels from its start to goal positions. MRPP produced optimal paths with good connectivity but tended to yield longer paths due to full VG computations, while CA achieved a notable reduction in both path length and planning time by simplifying the obstacle space. OCA provided the safest and shortest paths, thus proving ideal for high-density or safety-critical environments. The results in Table 5 indicate that OCA achieved the shortest average path length (8.54 m), followed by CA (8.62 m) and MRPP (8.97 m), demonstrating the effectiveness of optimisation and safety-aware refinements.

Table 5. Comparison of path length (metres (m)) for MRPP, CA, and OCA for five robots.

Robot	MRPP (m)	CA (m)	OCA (m)
R1	9.48	9.15	9.09
R2	8.93	8.91	8.66
R3	10.13	9.39	9.34
R4	7.60	7.14	7.12
R5	8.69	8.53	8.47
Average	8.97	8.62	8.54

The results presented in Table 5 represent a representative trial conducted in a static environment. Because the MRPP, CA, and OCA frameworks utilise Dijkstra's algorithm, a deterministic shortest path solver on a fixed roadmap, the generated path lengths are consistent and repeatable for any given set of start and goal configurations. These graph-based methods yield identical path solutions under constant environmental constraints, ensuring the reliability of comparative metrics. While the specific values in Table 5 are derived from a single scenario, the relative performance trend (OCA < CA < MRPP) has been observed consistently across various workspace geometries. The percentage path differences for the five-robot scenario are shown in Table 6.

Table 6. Path length percentage differences for five-robot scenario and three algorithms.

Algorithm	Percentage Difference
MRPP versus CA	$\frac{8.97 - 8.63}{8.97} \times 100 = 3.8\%$
MRPP versus OCA	$\frac{8.97 - 8.54}{8.97} \times 100 = 4.8\%$
CA versus OCA	$\frac{8.63 - 8.54}{8.63} \times 100 = 1.0\%$

OCA reduced the average path length by about 1.0% compared with CA. Using experimental averages, CA achieved a 3.8% reduction over MRPP, while OCA achieved a 4.8% reduction over MRPP and 1.0% over CA. The arrival times for the five-robot scenario are provided in Table 7. OCA achieved the shortest average arrival time (8.54 s), followed by CA (8.63 s) and MRPP (8.97 s), indicating improved efficiency due to optimisation and safety-aware planning. MRPP had the longest arrival time, reflecting its more complex, potentially less safe routes. CA was faster than MRPP, and OCA performed best at minimising arrival time.

Table 7. Arrival time (in seconds) for MRPP, CA, and OCA for five robots.

Robot	MRPP	CA	OCA
R1	9.48	9.15	9.09
R2	8.93	8.91	8.66
R3	10.13	9.39	9.34
R4	7.60	7.14	7.12
R5	8.69	8.53	8.47
Average	8.97	8.63	8.54

Simulated experiments for each algorithm demonstrated that all three algorithms successfully produced collision-free paths. MRPP achieved the most consistent global optimality but required the longest computation time due to sequential path generation. CA achieved a balance between path length and speed, while OCA outperformed both by generating the shortest paths with less computation time. The strong correlation between path length and arrival time confirms that the temporal gains are a direct result of the optimised geometric paths rather than random fluctuations in processing speed. Table 8 summarises the comparative performance metrics.

Table 8. Overview of key differences between MRPP, CA, and OCA for 5-robot scenario.

Feature/Algorithm	MRPP (Multi-Robot Path Planning Algorithm)	CA (Central Algorithm)	OCA (Optimisation Central Algorithm)
Base Method	Visibility Graph + Dijkstra + Algebraic Connectivity	Central Baseline + Reduced VG + Dijkstra	CA + Obstacle Safety Margin + Dijkstra
Graph type	Full Visibility Graph (all pairs)	Central Axis subset (reduced node connections)	Same as CA but with safety/margin modifications
Safety Distance from Obstacles	Low	Medium	High (obstacle margin)
Obstacle handling	All obstacles are considered	Only those intersecting Central Baseline	Obstacles expanded with safety margin
Path Planning	Dijkstra on full VG	Dijkstra on reduced VG	Dijkstra with modified edge weights
Path Optimality	High (VG + Dijkstra + λ_2)	High (selective obstacle avoidance)	High (VG + CB + obstacle expansion)
Computational Efficiency	Moderate (full visibility graph)	High (simplified obstacle set)	High (efficient + safe (D_s))
Time complexity	High ($O(n^2)$ visibility edges)	Reduced	Comparable to CA
Collision Avoidance	Good Via algebraic connectivity and path correction (sequence via λ_2)	Good Via Waypoint (W) generation through central Baseline (sequence via W)	Excellent Via Safety Distance (D_s)) and buffered collision zones. (D_s) avoids all collisions)
Real-world feasibility	Lower in dense scenes	Medium	High

4.7. Overall Assessment of MRPP, CA, and OCA Performance

To provide an overall assessment of algorithmic performance across the two experimental scenarios (scenario with three robots and scenario with five robots) the path lengths were averaged. This aggregate analysis enables an overall comparison of MRPP, CA, and OCA. The average path lengths for the three algorithms are provided in Table 9.

Table 9. Averaged path lengths for 3- and 5-robot scenarios.

Algorithm	Scenario 1	Scenario 2	Average
MRPP	10.43	8.97	9.70
CA	9.92	8.63	9.28
OCA	9.55	8.54	9.05

The associated percentage reduction differences for the three algorithms are shown in Table 10.

Table 10. Percentage path reductions for the three algorithms when the values for 3- and 5-robot scenarios were averaged.

Algorithm	Percentage Difference
MRPP versus CA	$\frac{9.70 - 9.28}{9.70} \times 100 = 4.33\%$

MRPP versus OCA	$\frac{9.70 - 9.05}{9.70} \times 100 = 6.75\%$
CA versus OCA	$\frac{9.28 - 9.05}{9.28} \times 100 = 2.48\%$

Averaging the two experimental scenarios, the mean path lengths are 9.70 m for MRPP, 9.28 m for CA, and 9.05 m for OCA. Compared with MRPP, CA achieves a 4.3% reduction in path length, while OCA provides a total reduction of 6.8%. Furthermore, OCA improves on CA by 2.5%, demonstrating its superior balance between optimality and safety.

5. Discussions

The comparative analysis of the three proposed algorithms revealed trade-offs between computational complexity, path optimality, safety, and real-world feasibility. The results confirmed that graph-theoretic strategies combining visibility-based methods can deliver scalable, safe, and communication-aware path planning for complex multi-robot systems. The study validated the suitability of MRPP, CA, and OCA for path planning in obstacle-rich environments and established a foundation for future extensions into dynamic and three-dimensional domains. The OCA stood out as the most robust option when safety and optimisation were critical objectives. In the following sections, a more thorough discussion of these issues is provided.

5.1. Performance Evaluations of the Algorithms

MRPP prioritises path optimality and connectivity by integrating VG, Dijkstra's algorithm, and algebraic connectivity (λ_2). It ensures effective inter-robot coordination and avoids collisions by leveraging λ_2-based sequencing to maintain the robustness of the robot communication network. Concurrently, it utilises VG to generate short, safe, and efficient paths. However, its major limitation lies in the computational burden due to full VG construction and sequential planning, making it less scalable in complex or dynamic environments.

CA and OCA represent advancements that build upon the strengths of the VG method while effectively addressing its inherent limitations, particularly concerning obstacle proximity and safety. A key distinction lies in their approach to path generation relative to obstacles. CA introduces a more computationally efficient approach by narrowing the planning space to relevant obstacles intersecting a CB. While it simplifies calculations for minimal distance and reduces planning time, it may compromise slightly on safety, as it does not explicitly maintain communication links or enforce obstacle clearance margins (the paths generated may pass close to obstacle corners). This can pose a safety risk in practical applications. Nevertheless, it offers a strong balance of speed and path feasibility.

OCA, incorporating safety margins through obstacle expansion, produces smoother, safer, and more reliable paths in cluttered environments. As an improvement over CA, OCA incorporates a fixed safety distance. This ensures that paths curve around obstacles, actively maintaining a safe distance and significantly enhancing path safety. This makes OCA especially valuable for real-world deployments where sensors may be imperfect and obstacle boundaries uncertain. The trade-off is a marginal increase in computation time as compared to CA. While CA guarantees a path that does not intersect obstacles, OCA further ensures that each robot maintains a specified clearance from obstacles. This makes OCA particularly well-suited for real-world applications where factors such as environmental uncertainty and the need for robust safety margins are critical considerations.

All three algorithms consistently employ Dijkstra's algorithm to compute shortest paths. This foundational choice ensures path optimality within their respective graph constructions and allows for adaptation to dynamic robot placements, although the extent of this adaptability varies with the specific algorithm's design. Overall, OCA balances path optimality, efficiency, and computational feasibility. CA is more efficient than MRPP while retaining a good performance. MRPP is best suited where robust connectivity (via λ_2) is critical.

Summary of advantages and disadvantages

MRPP:

- Advantages: Ensures shortest paths via Dijkstra's algorithm, maintains inter-robot connectivity, effective in static and well-known environments.
- Disadvantages: High computational cost, limited scalability, lacks adaptability to dynamic changes.

CA:

- Advantages: Efficient path generation, significantly reduced planning time, well-suited for dense environments.
- Disadvantages: May sacrifice path safety near obstacles; does not maintain connectivity awareness, less robust under environmental uncertainty.

OCA:

- Advantages: High safety due to margin buffers, ideal for real-world applications, preserves CA's efficiency while improving safety.
- Disadvantages: Marginally higher computational cost than CA, not yet adaptive to dynamic obstacle movement.

Overall Ranking: OCA → CA → MRPP.

- Best Overall: OCA is safest, shortest, and most application ready.
- Most Efficient: CA offers a strong trade-off between performance and computation.
- Most Theoretically Robust: MRPP excels in coordination and connectivity metrics.

5.2. Qualitative Comparison with Existing Path Planning Methods

Although the focus of this study was a comparative analysis between MRPP, CA, and OCA within a unified VG framework, it was also important to position these algorithms relative to the commonly used path planning algorithms reported in the literature. Classical graph-based approaches such as the Dijkstra and A* algorithms provide optimal shortest-path computation, but they do not inherently address multi-robot coordination or connectivity preservation, which are critical aspects in multi-robot systems [9,42–44]. Sampling-based methods such as probabilistic road map (PRM) and rapidly exploring random tree (RRT) offer efficient exploration in complex environments, but they typically generate non-optimal paths or require smoothing to achieve trajectory efficiency [9,45,46]. In contrast, MRPP, CA, and OCA extend the visibility graph framework to incorporate coordination, obstacle handling strategies, and safety considerations in multi-robot navigation. Therefore, while the classical algorithms primarily focus on single-robot optimal path computation, the proposed framework emphasises multi-robot coordination, safety-aware navigation, and computational efficiency within the same visibility graph structure. This qualitative comparison situates the proposed algorithms within the broader context of robotic path planning research.

5.3. Real-World Applications of Graph-Theoretic Multi-Robot Path Planning

MRPP graph-theoretic multi-robot path planning algorithms, including MRPP, CA, and OCA, offer broad applicability across diverse real-world scenarios where coordinated

action, operational efficiency, and stringent safety protocols are paramount. Their inherent adaptability renders them particularly suitable for deployment in a range of static and semi-structured environments [3,47–49].

Industrial automation and warehousing: In controlled industrial environments such as automated warehouses, assembly lines, and robotic fulfilment centres, path planning must prioritise efficiency and safety under high traffic densities [47–51]. CA and OCA are highly suitable due to their capacity for efficient path generation. The computational efficiency offered by CA and the safety-enhanced routing provided by OCA are particularly valuable in ensuring smooth, collision-free operations within high-density robotic environments.

Static and known workspaces: Environments with fixed layouts, such as laboratories or grid-based facilities, benefit from the deterministic nature of algorithms like MRPP, CA, and OCA. These algorithms ensure collision-free and optimally coordinated navigation, enabling robots to execute parallel tasks with minimal interference [10,52,53].

Communication-aware coordination: Robust inter-robot communication is fundamental for cooperative task execution. MRPP and CA offer scalable solutions that enable robots to coordinate across operational regions, while reducing unnecessary motion [3,10,48]. These algorithms preserve connectivity, thereby ensuring robots function as a cohesive unit [54–60].

Search-and-rescue and disaster response: In dynamic and partially structured environments such as collapsed buildings or post-disaster zones, robots must operate under uncertainty while maintaining communication. MRPP's emphasis on connectivity makes it well-suited for such missions, where communication robustness is critical for team survival and operational success [10,47,52,57].

The principles of MRPP and inter-agent spacing directly translate to scenarios such as autonomous vehicle platooning and intelligent intersection coordination. Algorithms like OCA are instrumental in ensuring that vehicles maintain safe distances while minimising overall travel time, even in dynamic and partially observable traffic environments.

Autonomous vehicle coordination: The principles behind multi-robot coordination apply directly to autonomous vehicle platooning and intelligent intersection control. OCA supports safe inter-vehicle spacing and dynamic path adjustments, improving both safety and efficiency in traffic networks [13,61].

Military and hazardous missions: in high-risk domains such as minefield navigation, reconnaissance, or surveillance in hostile terrain, algorithms like OCA are critical. They deliver mission efficiency while upholding strict safety standards in the absence of reliable external control [47,52,53].

Traffic networks and intelligent transport systems: Urban mobility systems face challenges like multi-robot environments. Graph-based MRPP techniques provide real-time re-routing, connectivity maintenance, and optimal inter-agent spacing, key enablers for autonomous traffic coordination and congestion mitigation [13,54–56].

Each of these application areas presents distinct operational priorities, ranging from paramount safety and rapid computational speed to unwavering communication robustness. The algorithms discussed offer tailored strengths that can be judiciously matched to these varying and specific demands.

5.4. Algorithms Challenges and Future Research Directions

Despite the promising outcomes and advancements presented, several critical challenges remain within the domain of multi-robot path planning using graph-theoretic approaches. Addressing these areas represents a significant opportunity for future research. Future work will involve adapting these algorithms for dynamic environments, automated parameter tuning, and decentralised multi-robot coordination.

Dynamic environments: A primary challenge involves enhancing the adaptability of these algorithms to environments characterised by moving obstacles or requiring real-time map updates. Current methods often struggle to maintain efficiency and optimality in such highly dynamic settings [47,50,51].

Decentralised planning: The MRPP, while effective, relies on sequential planning, which inherently introduces a degree of centralization. Real-time updates to visibility graphs and the continuous maintenance of algebraic connectivity during robot movement remain computationally intensive. Future research should prioritise the development of truly distributed or fully decentralised algorithms that can uphold safety and coordination without the overhead of centralised control [10,48,54].

Communication-aware coordination: Maintaining robust inter-robot communication is essential for cooperative task execution in multi-robot systems (MRS). MRPP and CA support scalable deployment over large operational regions, enabling synchronised behaviour and reduced redundant motion [3,47]. These algorithms ensure topological connectivity, often quantified via algebraic connectivity (λ_2), which sustains collaborative task engagement [54,60].

Scalability: While algorithms such as CA and OCA offer notable reductions in computational complexity compared to traditional methods, scaling them effectively to very large robot teams operating in real-time dynamic environments presents an ongoing and open challenge [53,57,60,61].

By addressing these challenges, future work can substantially extend the applicability, robustness, and intelligence of graph-theoretic approaches, thereby significantly enhancing the autonomy of multi-robot systems in increasingly complex real-world settings. These areas constitute a substantial agenda for continued research and innovation.

6. Conclusions

This study presented a comprehensive analysis of three graph-theoretic multi-robot path planning algorithms, i.e., MRPP, CA, and OCA. Each algorithm was evaluated and analysed in terms of its underlying principles, computational workflow, and performance within cluttered 2D polygonal environments. The algorithms were further explored through controlled simulations using common metrics, including path length, computation time, safety index, path smoothness, and algebraic connectivity. The findings demonstrate that MRPP provides the shortest paths and the robust inter-robot connectivity, benefiting from its full visibility graph and λ_2-based sequencing. However, this comes at the expense of increased computational time and reduced scalability. CA achieves rapid computation by constructing a reduced visibility graph focused on a central baseline. This approach is suitable for time-critical applications but exhibits weaker safety margins due to its lack of obstacle expansion or clearance optimisation. OCA delivers the best balance among all the criteria evaluated. By incorporating a weighted cost function and explicit safety distance expansion, OCA produces smooth, collision-free, and computationally efficient paths, outperforming CA in safety and approaching MRPP in quality while maintaining lower computational cost, making it the most practical for real-world deployment.

Overall, the results confirmed that the integration of geometric roadmaps with spectral connectivity measures and safety-aware optimisation yields a robust and efficient solution for multi-robot navigation in cluttered environments. These insights contribute to the development of research indicating that hybrid approaches combining visibility graphs, Laplacian connectivity, and adaptive weighting are essential for enabling coordinated, safe, and scalable robotic systems.

Author Contributions: Conceptualization, F.A.S.A., R.S., X.X. and L.A.; methodology, F.A.S.A., R.S., X.X. and L.A.; software, F.A.S.A., R.S. and L.A.; validation, F.A.S.A., R.S. and L.A.; formal

analysis, F.A.S.A., R.S. and L.A.; investigation, F.A.S.A., R.S., X.X. and L.A.; resources, F.A.S.A., R.S., X.X. and L.A.; data curation, F.A.S.A., R.S., X.X. and L.A.; writing—original draft preparation, F.A.S.A., R.S. and X.X.; writing—review and editing, F.A.S.A., R.S. and X.X.; visualisation, F.A.S.A., R.S. and X.X.; supervision, R.S., X.X. and L.A. Author L.A. passed away prior to the publication of this manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Please directly contact the author for access to the data.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CA	Central algorithm
CB	Central baseline
MRPP	Multi-robot path planning algorithm
MRS	Multi-robotic systems
OCA	Optimisation central algorithm
PL	Path length
PRM	Probabilistic road map
RRT	Rapidly exploring random tree
UGI	User graphic interface
VD	Voronoi diagram
VG	Visibility graph

Symbols

The following symbols are used in this manuscript:

G	Graph
V	Vertex
E	Edges
W_{ij}	Weighted edge
λ_2	Second smallest eigen value of Laplacian matrix
R_i	Robot
D_s	Safety Distance
δ	Minimum obstacle clearance along the edge.
α ,β	Weighting parameters balancing path length and safety.
O_i	Obstacles
s_i	Start positions of robots
g_i	Goals positions of robots
W_{free}	Free Workspace
n	Number of edges in the path.
$\widehat{W}_{\widehat{w}_i\widehat{w}_{i+1}}$	Weighted (distance) of edge $(\widehat{w}_i, \widehat{w}_{i+1})$

Appendix A. Operations of MRPP, CA and OCA

For completeness of the paper, in this appendix, the operations of the three graph-theoretic path planning algorithms (i.e., MRPP, CA and OCA) are described. The planning algorithms incorporate graph-theoretic representations in different ways, aligning their structural design with specific path planning and connectivity requirements [3,9,10,14,37]. These differences reflect the unique design objectives and operational strategies of each algorithm. The approaches demonstrate the versatility and importance of graph-theoretic representations in robotic path planning.

A.1. Problem Definition

Multi-robot motion planning concerns finding paths for multiple robots in an environment. The aim is to plan the movements of the robots from the start position (s_i) to the target position (g_i) in a workspace environment through a sequence of steps that will be described in the next sections [1,25,33,34].

A.2. Algorithms' Inputs and Outputs

The inputs to a planning framework for multiple robots consist of the following:

- Environmental graph (G): A graph *G (V, E)* where $|V| = N$.
- The vertices *V* of the graph are all the possible positions or waypoints available to the robots within the workspace.
- The edges *E* represent all feasible, collision-free paths (lines-of-sight) connecting the positions of R robots.
- Robot team (R): A set of n robots, denoted as R_1, R_2..., R_n.
- Configuration states: For each robot R_i, two specific states are defined: a start position $s_i \in V$, and a goal position $g_i \in V$.

The output is a path plan for each robot, ensuring all robots reach their goal locations without collisions [10,22,35].

A.3. Multi-Robot Path Planning Algorithm (MRPP)

The MRPP integrates three foundational techniques:

(i) Visibility graphs for modelling the environment and identifying feasible shortest connections.
(ii) Dijkstra's algorithm for computing optimal paths between nodes from start to goal positions.
(iii) Algebraic connectivity λ_2 to assess communication robustness and to guide the sequential planning of robot paths [3,10,36].

The procedure begins with constructing a VG representing all visible connections between obstacle vertices, the robot's start and goal positions [10,37]. The Laplacian matrix of the graph is then computed, and λ_2 is used to determine network connectivity. Path planning proceeds sequentially, ensuring each robot's path is optimised while avoiding any collision with previously assigned paths [3,9,10,27,38]. It assumes full a priori knowledge of the environment, including obstacle geometry and the start and goal positions of each robot.

A.3.1. MRPP Description

Inputs: Robots' start positions (s_i), goal positions (g_i), and polygonal obstacles (O_i).
Outputs: visibility graph (VG), optimal paths from source si to goal (destination) gi.

(i) Establish a free workspace (W_{free}): Construct the free configuration space and map all obstacle boundaries, robot start position, and goal position. The free workspace is the entire 2D plane minus all obstacles. $W_{\text{free}} = R^2 \setminus \bigcup_i O_i$, where W_{free} is the subset of the two-dimensional Euclidean plane (R^2) that is not occupied by obstacles. R^2 is therefore the entire 2D workspace where the robots operate. It represents all possible points (x,y) in a continuous plane. $\bigcup_i O_i$ denotes the union of all obstacle regions. The free workspace is represented by an environmental graph $G_{env} = (V_{env}, E_{env})$ that provides a graph-based abstraction of the obstacle-free space for path planning, where V_{env}, is the set of vertices and E_{env} the set of edges.

(ii) Determine robots' Start (s_i) and Goal (g_i) positions: Identify: $s_i = \{s_i | s_i \in W_{free}, i = 1 \ldots, n\}, g_i = \{g_i | g_i \in W_{free}, i = 1 \ldots, n\}$ and record the number, geometry, and positions of all obstacles.

(iii) Partition the workspace graph: Represent the environment as an undirected weighted graph G_{env}. Divide this graph into two disconnected components of subgraphs $G_{env} \leftarrow \{G1, G2\}$, $G_1 \cap G_2 = \emptyset$. These subgraphs correspond to distinct connected regions of the free workspace that are not mutually reachable.

(iv) Connectivity enhancement via λ_2: Let L be the Laplacian matrix of the current environment (or communication) graph. For all possible candidate edges (w_{ij}) that could link the two components: $w_{ij} = (edge\ between\ (v_i, v_j)$, where $v_i \in G_1$ and $v_j \in G_2$. Evaluate the algebraic connectivity of the Laplacian: $\lambda_2 = \lambda_2(L)$. Select the edge (or set of edges) that maximises λ_2, thereby increasing network robustness and improving global mobility of the workspace. For each candidate edge (w_{ij}), form the updated Laplacian $L' = L + L(w_{ij})$, where $L(w_{ij})$ is the Laplacian contribution of adding edge (i, j). The optimal edge is selected as

$$w^*_{ij} = \arg \max_{w_{ij}} \lambda_2(L + L(w_{ij})) \quad \text{(A1)}$$

The selected edge (w^*_{ij}) is the one that yields the largest increase in λ_2, thereby strengthening graph connectivity, improving robustness to disconnection, and enhancing the global traversability of the workspace.

(v) Optimisation process: The optimisation phase refines the initial paths to ensure the path length remains near optimal. The robot team maintains sufficient communication connectivity, typically expressed as keeping $\lambda_2 < \lambda_{min} > 0$, where λ_{min} is a predefined threshold that ensures reliable inter-robot communication. The optimisation proceeds through the following operations. The value of λ_{min} is selected based on communication range, network density, and robustness requirements of the multi-robot system.

a) Evaluate connectivity during motion: At each planning iteration, the communication graph (G_c) is updated according to the robots' current positions, and its λ_2 is recomputed: $\lambda_2 = \lambda_2(L(G_c))$. If $\lambda_2 < \lambda_{\min}$, this indicates that robots are at risk of communication loss. If $\lambda_2 \geq \lambda_{\min}$, the team remains well-connected. G_c denotes the communication graph, which represents the communication links between robots at a given planning iteration. The Laplacian matrix. $L(G_c)$ of this graph is used to compute the algebraic connectivity.

b) Connectivity-aware path adjustment: If the connectivity requirement $\lambda_2 \geq \lambda_{min}$ is not satisfied, the algorithm introduces path modifications to enforce team coherence: Adjust robot positions to keep them inside the communication range of neighbours. The algorithm introduces temporary edges into the communication graph to bolster network connectivity. It prejudices robot motion toward regions that maintain or increase λ_2. This ensures that $\lambda_2^{(new)} > \lambda_2^{(old)}$, thereby ensuring that the communication graph remains robust throughout the mission, the network is harder to disconnect, and information can flow more reliably among robots, where $\lambda_2^{(old)}$: connectivity before adjustment, $\lambda_2^{(new)}$: connectivity after adjustment. The path modification step successfully improves team connectivity while the robots move, reducing the risk of communication loss.

(vi) Construct the visibility graph (VG): Create the visibility graph $VG = (V, E)$, where V includes obstacle vertices, start points, and goal points $V = s_i, \cup g_i, \cup$ {all obstacle vertices}, and E includes all pairs of vertices with line-of-sight connectivity (collision-free edges) $(v_i, v_j) \in E$ iff $\overline{v_i v_j} \subset W_{free}$.

(vii) Compute shortest paths using Dijkstra's algorithm: For each robot R_i, compute the waypoints $(\widehat{W}_i)$ = Dijkstra (VG, s_i, g_i). This yields an ordered waypoint sequence,

$\widehat{W}_i = \{w_0, w_1, \ldots, w_n\}$, where $w_0 = s_i, w_n = g_i$. The output is the set of optimal collision-free paths for all robots across the visibility graph, enhanced by improved graph connectivity obtained through λ_2-based edge augmentation, $P = \{P_1, P_2, \ldots, P_n\}$, where P is the optimal path for each robot [3,10].

The operations of the MRPP are also illustrated in Figure A1.

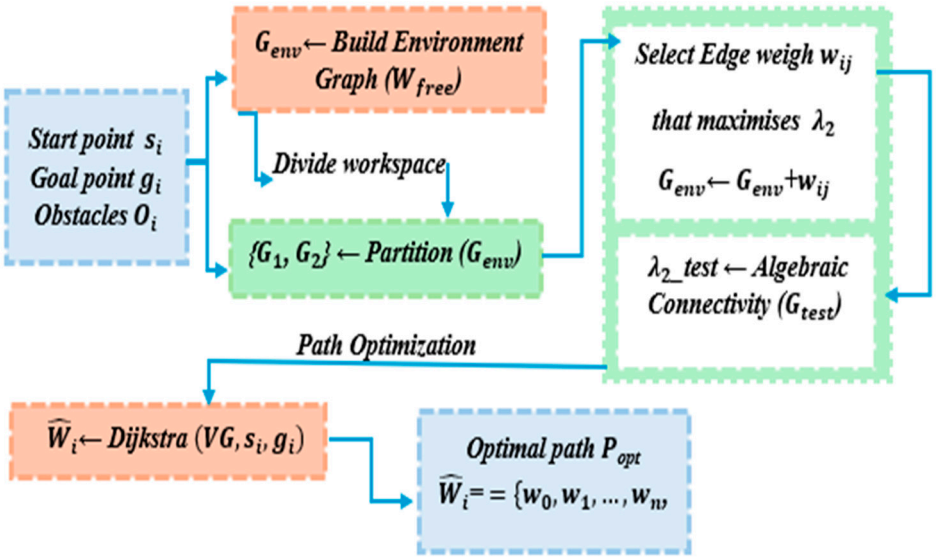

Figure A1. The operation of the MRPP.

Pseudocode placeholder: Algorithm A1: multi-robot path planning algorithm (MRPP).

Algorithm A1. Multi-robot path planning algorithm (MRPP).

Inputs: *Start positions of robots:* $s_i = \{s_1, s_2, \ldots, s_n\}$,, *Goal positions of robots:* $g_i = \{g_1, g_2, \ldots, g_n\}$,, *and set of polygonal obstacles:* $O_i = \{O_1, O_2, \ldots, O_n\}$,

Outputs: *VG, and Optimal paths from each* s_i *to* g_i

1. *Construct Free Workspace:* W_{free} ← *BuildFreeSpace* (O_i)

2. *Initialise Robot Start and Goal Positions: Identify all* $s_i \in S$ *and* $g_i \in G$. *Extract obstacle geometry and positions*

3. *Partition Workspace Graph:* G_{env}← *Build Environment Graph* (W_{free}). {G_1, G_2} ← *Partition* (G_{env})// *two disconnected components*

4. *Connectivity Enhancement using* λ_2: *Candidate Edges* ← *All possible edges between vertices in* G_1 *and* G_2. *For each edge* $w_{ij} \in$ *Candidate Edges:* G_{test} ← G_{env}+ w_{ij}. λ_2_test ← *Algebraic Connectivity* (G_{test}). *Select* w_{ij} *that maximises* λ_2_test: G_{env}← G_{env}+w_{ij}

5. *Construct Visibility Graph (VG): VG* ← *Build Visibility Graph (O, S, G)*

6. *Compute Shortest Paths (Dijkstra): For each robot* R_i:$\widehat{W}_i$← *Dijkstra* (VG, s_i, g_i)

7. *Output Final Paths:* $\widehat{W}_i$= = $\{w_0, w_1, \ldots, w_n\}$, $w_0 = s_i, w_n = g_i$.

8. *Return Final Paths:* $P = \{P_1, P_2, \ldots, P_n\}$.

Key Features for MRPP:

- Sequential planning: The paths of robots are planned sequentially. The order is determined by λ_2 values to reduce potential conflicts.

- Edge weight adjustment: Edge weights are dynamically modified based on previously planned paths to minimise overlap and prevent collisions.
- Optimal path assurance: Utilises Dijkstra's algorithm on well-constructed VG, ensuring the shortest and safest feasible paths.
- Suitable for static, fully known environments.
- Maintains inter-robot communication.
- Avoids collision without complete re-planning.

A.4. Central Algorithm (CA)

The CA builds on the VG principle but introduces a central baseline (CB) strategy to drastically reduce computational complexity. Instead of connecting every visible pair of vertices, CA constructs connections only through those obstacles intersecting with a predefined baseline that lies between the start and goal points. This approach simplifies VG by minimising unnecessary vertices and edges. Paths are then generated through waypoints defined along the CB, ensuring obstacle avoidance and shorter travel distances. This results in significantly reduced computational complexity while maintaining near-optimal path quality and enhancing efficiency [3,14].

A.4.1. CA Description

Figure A2 illustrates the waypoint calculation, where the Central Baseline joins the start point S and goal point g intersect an obstacle, the edges of which are shown as dotted lines, at point $u(x, y)$.

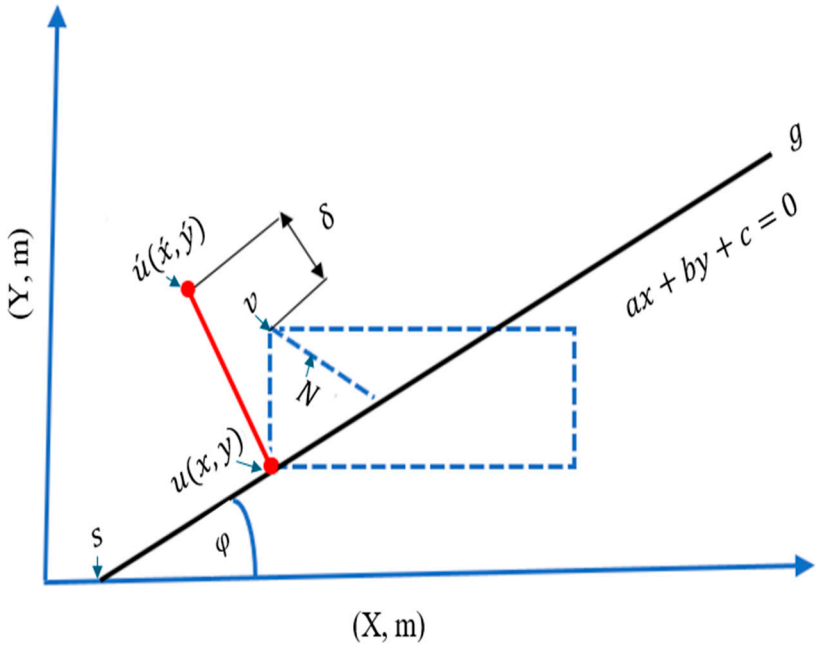

Figure A2. Computation of waypoint (u') at intersection point (u).

It is required to calculate a point $u'(x', y')$ that creates an orthogonal line to the central baseline at point $u(x, y)$. The normal distance between the point $u'(x', y')$ and the straight line must exceed the maximum distance N, where N denotes the maximum perpendicular distance between the CB and any vertex v of the intersecting obstacle. The waypoint must lie outside the obstacle by at least a safe clearance distance δ. The distance N is computed as $N = \frac{|(g-s)X\,(s-v)|}{|g-s|}$, or alternatively, if the CB is represented in implicit line form: $ax + by + c = 0$. Then, the perpendicular distance from a point (x, y) to the CB is

$$N = \left|\frac{ax + by + c}{\sqrt{a^2 + b^2}}\right| \tag{A2}$$

In the given equations $u,\ u', v, s, and\ g$ are position vectors for the intersection point, waypoint, vertex, start, and goal locations, whereas *a, b,* and *c* are constants, and *x, y* are variables. Let φ be the inclination angle of the CB with respect to the positive x-axis. Since the slope of the CB is: $M = \tan(\varphi)$, we have $\sin(\varphi) = \frac{\text{0pposite}}{\text{Hypotenuse}}, \cos(\varphi) = \frac{\text{Adjacent}}{\text{Hypotenuse}}$.

A unit vector orthogonal to the CB is

$$\hat{u}_{\perp} = \begin{bmatrix} -cos(90 - \varphi) \\ sin\,(90 - \varphi) \end{bmatrix} \tag{A3}$$

Thus, the waypoint u' is computed on either side of the intersection point as $u' = u \pm (N + \delta)\,\hat{u}_{\perp}$:

$$u' = u \pm (N + \delta) \begin{bmatrix} -cos(90 - \varphi) \\ sin\,(90 - \varphi) \end{bmatrix} \tag{A4}$$

where N is the maximum distance between the CB and any vertex v, and δ is the safe clearance distance. Or explicitly as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \pm (N + \delta) \begin{bmatrix} -cos(90 - \varphi) \\ sin(90 - \varphi) \end{bmatrix} \tag{A5}$$

For each intersection point (u), two symmetric waypoints are generated, $\acute{u}^{(+)} = u + d\,\hat{u}_{\perp}, and\ \acute{u}^{(-)} = \mathrm{u} - d\,\hat{u}_{\perp}$, where $\acute{u}^{(\pm)} = \hat{w}_i^{(\pm)}$, and $d = (N + \delta) > 0$ is the safety offset (clearance distance). Each obstacle produces four waypoints because its CB intersection consists of two points, and each point yields two orthogonal offsets, $\acute{u}_i = \{\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4\}$. Thus, the full waypoint set is

$$\widehat{W}_i = \{s_i, g_i\} \cup \left\{ \hat{w}_i^{(+)}, \hat{w}_i^{(-)} \mid \mathrm{u} \in O_i \right\} \tag{A6}$$

Two waypoints $\hat{w}_i, \hat{w}_{i+1} \in \widehat{W_i}$ are connected by an edge if the straight segment between them does not intersect any obstacle (i.e., they are visible (Vis)):

$$(\hat{w}_i, \hat{w}_{i+1}) = \begin{cases} \hat{w}_{i,i+1}, if\ \overline{(\hat{w}\hat{w}_{\iota+1}} \cap\ O_i\ =\ \emptyset\ \forall i \\ \emptyset, Otherwise \end{cases} \tag{A7}$$

i. Create a reduced graph $G'(V', E')$ that is constructed from set of all vertices (waypoints generated from the relevant obstacles $\hat{O}_i$, the start, the goal) and visible connections among them, where $V' = \widehat{W}_i$, and

$$E' = \{\ \hat{w}_i, \hat{w}_{i+1}\ \mid \mathrm{Vis}\ (\hat{w}_i, \hat{w}_{i+1}) = \hat{w}_{i,i+1} | (i, i+1) \in V'\} \tag{A8}$$

ii. An edge $(\hat{w}_i, \hat{w}_{i+1}) \in E'$ is included in the graph if and only if the straight-line segment connecting the two waypoints lies entirely in the free workspace, i.e., $\hat{w}_{\iota}\bar{\hat{w}}_{\iota+1} \cap O_i = \emptyset$ for all obstacles O_i.

iii. Apply Dijkstra's algorithm on G' to obtain the shortest feasible path. $P = \mathrm{Dijkstra}(G', S_i, g_i)$. The optimal CA-based path is

$$P = \arg \min_{P \in \mathcal{P}(S_i, g_i)} \sum_{i=0}^{\mathrm{n}-1} \widehat{W}_{\hat{w}_i \hat{w}_{i+1}}\ \text{s.t.} \begin{cases} \hat{w}_0 = s_i, \hat{w}_n = g_i, \\ (\hat{w}_i, \hat{w}_{i+1}) \in E' \forall i = 0, \ldots, n-1, \\ s_{\hat{w}_i \hat{w}_{i+1}} \geq D_s \forall i = 0, .., n-1. \end{cases} \tag{A9}$$

The $arg\ min$ returns the sequence of waypoints (P) that minimises the total cost across the reduced visibility graph produced by CA. Feasibility requires each consecutive waypoint pair to be visible (edge in E') and to have clearance at least distance D_s. s.t is a

standard abbreviation for "subject to" (under the constraints that) the following constraints:

a) The path $P(s_i, g_i)$: set of all feasible paths from start (s_i) to goal (g_i).
b) n: number of edges in the path.
c) $\widehat{W}_{\widehat{w}_i\widehat{w}_{i+1}}$: weighted (distance) of edge $(\widehat{w}_i, \widehat{w}_{i+1})$
d) Each consecutive pair of waypoints $(\widehat{w}_i, \widehat{w}_{i+1})$ must be a valid edge in the graph.
e) E': edge set of the planning graph
f) The clearance $s_{\widehat{w}_i\widehat{w}_{i+1}}$: minimum clearance to obstacles along the edge.
g) D_s: required safety distance

Figure A3 provides an illustrative example of waypoint generation using the CA in a workspace containing seven static polygonal obstacles (shown in grey).

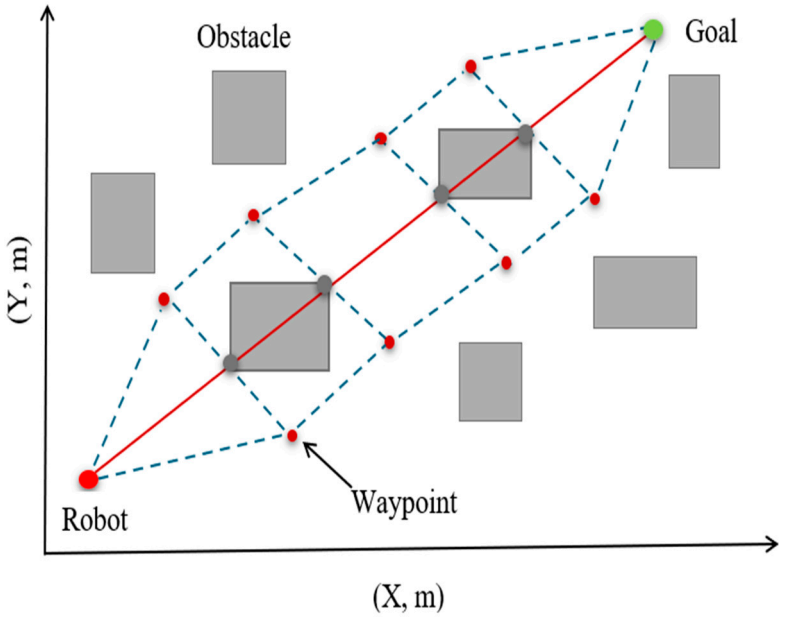

Figure A3. An illustrative example of waypoint generation using the CA in a workspace containing seven static polygonal obstacles (shown in grey).

The CB, depicted as a red line, represents the direct straight-line connection between the robot's start position (red point) and goal position (green point). Only the obstacles that intersect the CB are considered relevant for path computation. For each intersecting obstacle, four waypoints (small red markers) are generated in the local collision-free region, positioned symmetrically around the CB. These waypoints serve as critical vertices that define feasible detour options. By connecting the start position, the generated waypoints, and the goal position, the CA produces two distinct collision-free path alternatives around the obstacle geometry. The operations of the CA are illustrated in Figure A4.

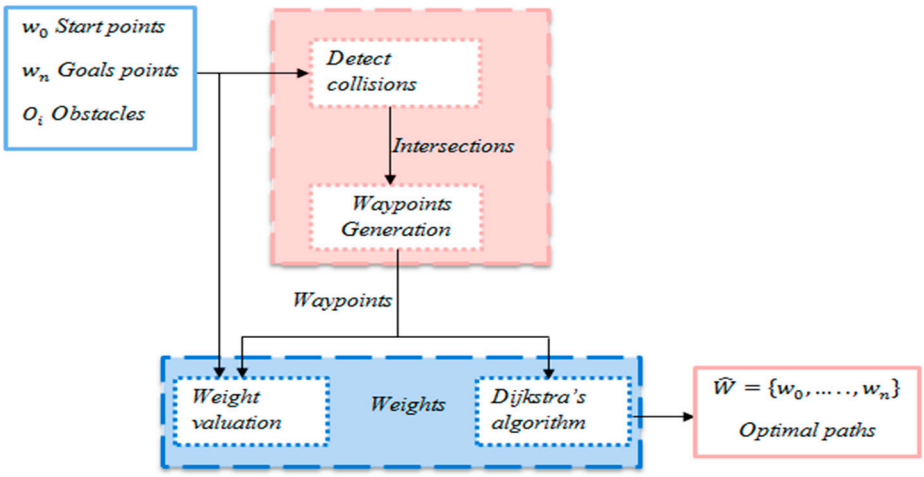

Figure A4. The operation of the CA.

Pseudocode placeholder: Algorithm A2: central algorithm (CA)

Algorithm A2. The central algorithm.

Input: *Start* S_i, *Goal* g_i, *Obstacle set* O_i

Output: *Reduced graph G′ and shortest path P*

1. Define Central Baseline CB (S, G)

2. For each obstacle O_n *in* O_n*: if* O_i *intersects CB: generate waypoint vertices* w_i *near intersection*

3. Construct reduced visibility graph G′(V′, E′)

4. P ← Dijkstra (G′, s, g)

5. Return P

Key feature: CA is computationally efficient because it drastically reduces the number of vertices and edges.

A.5. Optimisation Central Algorithm (OCA)

OCA enhances CA by introducing optimisation-based weighting, safety distance control, and path correction mechanisms [3,14]. OCA builds safety distance (*Ds*) to ensure safer navigation. *Ds* is the average minimum distance from any obstacle. It modifies cost functions and graph edges adaptively based on obstacle proximity and inter-robot distance. This algorithm produces general solutions and acceptable results for different maps because it generates waypoints around obstacles in workspace. This provides a self-adjusting mechanism for real-time path correction, thereby enhancing path smoothness, redundant motion, and ensures optimality in comparison to CA and traditional VG methods.

A.5.1. CA Description

Inputs: start point (s_i), goal point (g_i), obstacles (O), and weights (α, β).
Output: Optimised safe path (P_{opt}).

i. Obstacle expansion: Obstacles are artificially enlarged using a safety distance (e.g., δ).
ii. Angular adjustment: Path curvature is modified to maintain clearance from expanded obstacles.
iii. Construct the reduced visibility graph $G'(V', E')$ using these principles.

A.5.2. Safety Distance and Obstacle Expansion

As shown in Figure A5, to avoid collisions, each obstacle is expanded by a safety distance (D_s), calculated as: $D_S = PQ = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$, where (x_0, y_0) are the Cartesian coordinates of point P, and (x_1, y_1) are the Cartesian coordinates of point Q. Figure A5 provides the expression that computes the Euclidean distance between points P and Q, which in this context represents the safety distance (D_S), which is the straight-line distance between point $P(x_0, y_0)$ and point $Q(x_1, y_1)$ in the 2D workspace.

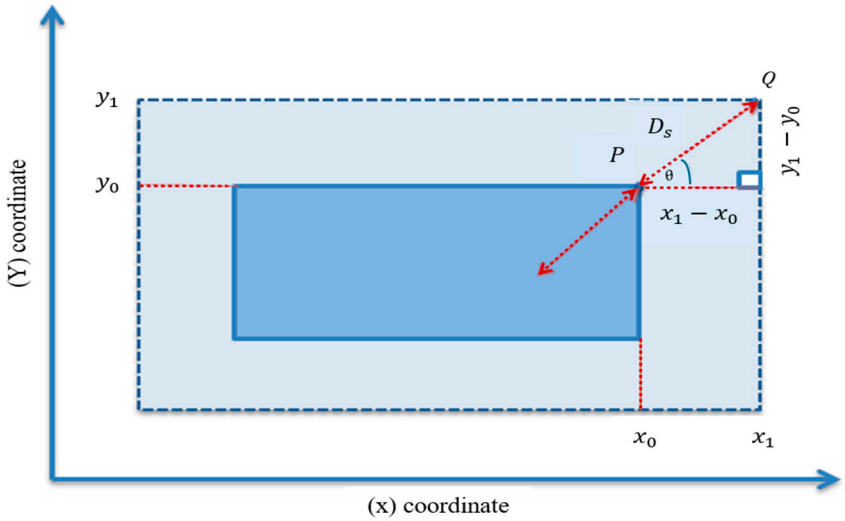

Figure A5. The approach to computes the Euclidean distance between points *P* and *Q*, which in this context represents the safety distance (D_S) which is the straight-line distance between point $P(x_0, y_0)$ and point $Q(x_1, y_1)$ in the 2D workspace.

Alternatively, using polar coordinates, the Cartesian coordinates $(x_1 - x_0)$ and $(y_1 - y_0)$ can be converted to the polar coordinates r and θ using trigonometric functions sine and cosine as follows:

$$r = D_s, (x_1 - x_0) = r\cos\theta\,, and\ (y_1 - y_0) = r\sin\theta \tag{A10}$$

$$r = \sqrt{(rcos\theta)^2 + (rsin\theta)^2} \tag{A11}$$

This expansion ensures that the planned paths maintain a buffer around obstacles proportional to D_s.

A.5.3. Weight Function Definition

For each edge: $E = e_{ii+1}{:}w_{i,i+1} = \alpha d_{i,i+1} + \beta\left(\frac{1}{\delta_{i,i+1}}\right)$, where $d_{i,i+1}$ is the Euclidean distance between vertices i and j (in metres). $\delta_{i,i+1}$ reflects the computed safety distance (D_s). It is the minimum clearance from obstacles along the edge connecting vertices i and $i+1$ (in metres). α represents the weight assigned to the path length (coefficient for distance, reflecting preference for shorter paths). β represents the safety weight (the importance of maintaining a safe distance from obstacles).

- Apply an iterative optimisation loop to minimise the total distance:

$$P_{min} = \underset{(i,i+1)\in p}{min} \sum w_{i,,i+1} \tag{A12}$$

where P is a path, defined as a sequence of connected edges between the start and goal vertices. $\sum w_{i,i+1}$ is the total distance of the path sum of all edge weights along $P, and\ w_{i,i+1}$ is the weight of moving from vertices $i\ and\ i+1$. $min_{\,(i,i+1)\in P}$ is the

optimisation operator to find the path $P_{\min}$ that yields the minimum total distance among all possible paths between the start and goal nodes. $P_{\min}$ is the optimal path (the sequence of edges with the lowest cumulative distance.

- Introduce obstacle expansion margins to prevent near-obstacle travel.
- Execute Dijkstra's search using updated weights to obtain a safe and efficient path.

$$\widehat{W} = \arg\ min_{P=(w_0,\ldots,w_n)} \sum_{i=0}^{n-1} \left[\alpha d_{w_i w_{i+1}} + \beta \left(\frac{1}{\delta_{w_i w_{i+1}}} \right) \right] \quad \text{(A13)}$$

Subject to (s.t)

a. The path $P = (w_0, \ldots, w_n)$, set of all paths from start $w_0 = s_i$, $w_n = g_i$
b. $d(w_i, w_{i+1})$: Euclidean distance between consecutive waypoints.
c. $\delta(w_i, w_{i+1})$: minimum obstacle clearance along the edge.
d. α, β: weighting parameters balancing path length and safety.
e. Repeat for all robots sequentially or in parallel (depending on communication constraints).

The operations of the OCA are illustrated in Figure A6.

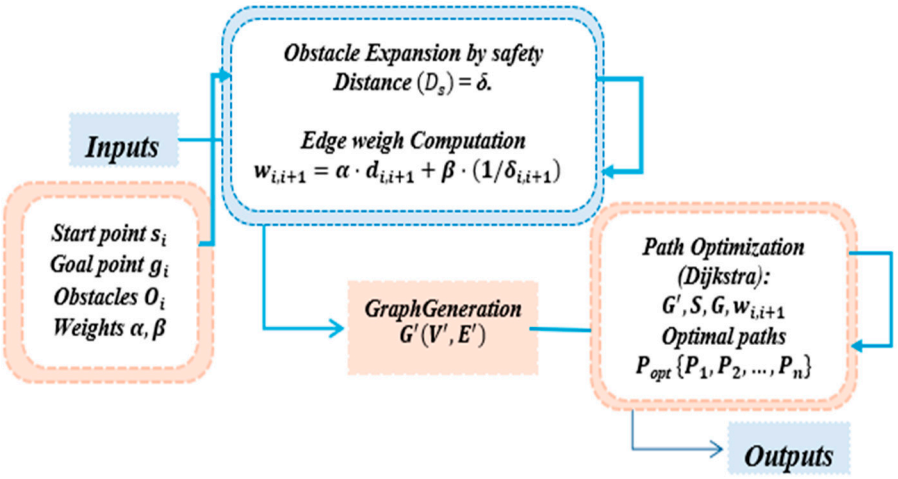

Figure A6. The operation of the OCA.

Pseudocode placeholder: Algorithm A3: optimisation central algorithm (OCA)

Algorithm A3. Optimisation algorithm for OCA.

***Inputs**: Start point S, Goal point G, Obstacles O, and Weights α, β*

***Output:** Optimised safe path P_{opt}*

1. *Graph Generation (via OCA): G (V, E)*
2. *Obstacle Expansion by safety Distance δ*
3. *Edge weigh Computation: For each edge $e_{ij} \in E'$: $w_{i,i+1} = \alpha \cdot d_{i,i+1} + \beta \cdot (1/\delta_{i,i+1})$*
4. *Construct graph $G'(V', E')$*
5. *Path Optimization (Dijkstra): $G', S, G, w_{i,i+1}$*

Optimised safe path P_{opt}

Advantages:

- Ensures high safety standards.
- Avoids close-proximity paths that may be risky in real-world applications.
- Retains path optimality and computational feasibility, i.e., selects the most efficient and safe route by minimising the total weighted cost accumulated along the candidate paths between the start and goal nodes.

References

1. Xu, K.; Xing, J.; Sun, W.; Xu, P.; Yang, R. Multi-Robot Collision Avoidance Method in Sweet Potato Fields. *Front. Plant Sci.* **2024**, *15*, 1393541.
2. Sellers, T.; Lei, T.; Carruth, D.W.; Luo, C. Synergizing Graph-Based Methods with Biologically Inspired Algorithms for Enhanced Robot Path Planning Efficiency. In *Unmanned Systems Technology XXVI*; SPIE: Bellingham, WA, USA, 2024; Volume 13055, pp. 52–64.
3. Alwafi, F.A.S. Social Graphs and Their Applications to Robotics. Ph.D. Thesis, Sheffield Hallam University, Sheffield, UK, 2022. Available online: https://shura.shu.ac.uk/31906/ .
4. Lei, T.; Sellers, T.; Luo, C.; Carruth, D.W.; Bi, Z. Graph-Based Robot Optimal Path Planning with Bio-Inspired Algorithms. *Biomim. Intell. Robot.* **2023**, *3*, 100119.
5. Zhang, W.; Li, J.; Yu, W.; Ding, P.; Wang, J.; Zhang, X. Algorithm for UAV Path Planning in High Obstacle Density Environments: RFA-Star. *Front. Plant Sci.* **2024**, *15*, 1391628.
6. AbuJabal, N.; Rabie, T.; Baziyad, M.; Kamel, I.; Almazrouei, K. Path Planning Techniques for Real-Time Multi-Robot Systems: A Systematic Review. *Electronics* **2024**, *13*, 2239.
7. Al-Metari, M.; Aziz, N.A.H.; Sadeq, L. Path Planning with Obstacle Avoidance. Bachelor's Thesis, College of Engineering and Applied Sciences, American University of Kuwait, 2021. Available online: https://www.researchgate.net/publication/384178197_Path_planning_with_obstacle_avoidance_for_a_mobile_robot_supporting_industrial_processes (accessed on 26 January 2026).
8. Sánchez-Ibáñez, J.R.; Pérez-del-Pulgar, C.J.; García-Cerezo, A. Path Planning for Autonomous Mobile Robots: A Review. *Sensors* **2021**, *21*, 7898.
9. Bui, H.D. A Survey of Multi-Robot Motion Planning. *arXiv* **2023**, https://doi.org/10.48550/arXiv.2310.08599.
10. Alwafi, F.A.S.; Xu, X.; Saatchi, R.; Alboul, L. Development and Evaluation of a Multi-Robot Path Planning Graph Algorithm. *Information* **2025**, *16*, 431. https://doi.org/10.3390/info16060431.
11. Malli, I.; Bechlioulis, C.P.; Kyriakopoulos, K.J. Robust Distributed Estimation of the Algebraic Connectivity for Networked Multi-Robot Systems. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 9155–9160. Available online: https://ieeexplore.ieee.org/document/9561201/authors#authors (accessed on).
12. Murayama, T. Dynamical Improvement of Network Connectivity for Multi-Robot Robustness Using Perturbed Algebraic Connectivity. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Fukushima, Japan, 11–14 January 2021. Available online: https://ieeexplore.ieee.org/document/9382705 (accessed on).
13. Pasham, S.D. Using Graph Theory to Improve Communication Protocols in AI-Powered IoT Networks. *Res. Anal. J.* **2024**, *7*, 17–48.
14. Alwafi, F.; Xu, X.; Alboul, L.; Penders, J. Visibility Graph (VG)-Based Path Planning Method for Multi-Robot Systems. In Proceedings of the UKRAS22 Network Robotics and Autonomous Systems 2022, Wales, UK, 26 August 2022; pp. 60–61. Available online: https://uk-ras.org.uk/publications/ukras22-robotics-unconstrained-environments (accessed on 26 January 2026).
15. Zhang, H.Y.; Lin, W.M.; Chen, A.X. Path Planning for the Mobile Robot: A Review. *Symmetry* **2018**, *10*, 450.
16. Yu, J.; LaValle, S.M. Optimal Multi-Robot Path Planning on Graphs: Complete Algorithms and Effective Heuristics. *IEEE Trans. Robot.* **2016**, *32*, 1163–1177.
17. Yu, J. Intractability of Optimal Multi-Robot Path Planning on Planar Graphs. *arXiv* **2015**, https://doi.org/10.48550/arXiv.1504.02072.
18. Blasi, L.; D'Amato, E.; Mattei, M.; Notaro, I. Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph. *Appl. Sci.* **2020**, *10*, 5613.
19. Hu, Y.; Xie, F.; Yang, J.; Zhao, J.; Mao, Q.; Zhao, F.; Liu, X. Efficient Path Planning Algorithm Based on Laser SLAM and an Optimized Visibility Graph for Robots. *Remote Sens.* **2024**, 16, 2938. https://doi.org/10.3390/rs16162938.
20. Niu, H.; Savvaris, A.; Tsourdos, A.; Ji, Z. Voronoi-Visibility Roadmap-Based Path Planning Algorithm for Unmanned Surface Vehicles. *J. Navig.* **2019**, *72*, 850–874. https://doi.org/10.1017/S0373463318001005.
21. Al-Kamil, S.J.; Szabolcsi, R. Optimizing Path Planning in Mobile Robot Systems Using Motion Capture Technology. *Results Eng.* **2024**, *22*, 102043.
22. Solis, I.; Motes, J.; Sandström, R.; Amato, N.M. Representation-Optimal Multi-Robot Motion Planning Using Conflict-Based Search. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4608–4615.

23. Hvězda, J. Comparison of Path Planning Methods for a Multi-Robot Team. Master's Thesis, Czech Technical University in Prague, Prague, Czech Republic, 2017. Available online: https://dspace.cvut.cz/bitstream/handle/10467/69497/F3-DP-2017-Hvezda-Jakub-Comparison_of_path_planning_methods_for_a_multi-robot_team.pdf (accessed on 26 January 2026).
24. Lee, W.; Choi, G.H.; Kim, T.W. Visibility Graph-Based Path-Planning Algorithm with Quadtree Representation. *Appl. Ocean. Res.* **2021**, *117*, 102887.
25. Zhao, D.; Zhang, S.; Shao, F.; Yang, L.; Liu, Q.; Zhang, H.; Zhang, Z. Path Planning for the Rapid Reconfiguration of a Multi-Robot Formation Using an Integrated Algorithm. *Electronics* **2023**, *12*, 3483. https://doi.org/10.3390/electronics12163483.
26. Fiedler, M. Algebraic Connectivity of Graphs. *Czechoslov. Math. J.* **1973**, *23*, 298–305. Available online: https://snap.stanford.edu/class/cs224w-readings/fiedler73connectivity.pdf (accessed on 26 January 2026).
27. Griparic, K. Algebraic Connectivity Control in Distributed Networks by Using Multiple Communication Channels. *Sensors* **2021**, *21*, 5014. https://doi.org/10.3390/s21155014.
28. Zhao, W.; Deplano, D.; Li, Z.; Giua, A.; Franceschelli, M. Algebraic Connectivity Control and Maintenance in Multi-Agent Networks Under Attack. *arXiv* **2024**, arXiv:2406.18467. https://doi.org/10.48550/arXiv.2406.18467.
29. Defoort, M.; Veluvolu, K.C. A Motion Planning Framework with Connectivity Management for Multiple Cooperative Robots. *J. Intell. Robot. Syst.* **2014**, *75*, 343–357.
30. Woosley, B.; Dasgupta, P.; Rogers, J.G.; Twigg, J. Multi-Robot Information Driven Path Planning Under Communication Constraints. *Auton. Robot.* **2020**, *44*, 721–737.
31. Murayama, T. Distributed Control for Bi-Connectivity of Multi-Robot Network. *SICE J. Control. Meas. Syst. Integr.* **2023**, *16*, 1–10.
32. Matos, D.M.; Costa, P.; Sobreira, H.; Valente, A.; Lima, J. Efficient Multi-Robot Path Planning in Real Environments: A Centralized Coordination System. *Int. J. Intell. Robot. Appl.* **2025**, *9*, 217–244.
33. Zhou, C.; Li, J.; Shi, M.; Wu, T. Multi-Robot Path Planning Algorithm for Collaborative Mapping under Communication Constraints. *Drones* **2024**, *8*, 493. https://doi.org/10.3390/drones8090493.
34. Ma, Y.; Mao, Z.; Wang, T.; Qin, J.; Ding, W.; Meng, X. Obstacle Avoidance Path Planning of Unmanned Submarine Vehicle Using Improved Firework–Ant Colony Algorithm. *Comput. Electr. Eng.* **2020**, *87*, 106773.
35. Liang, D.; Liu, Z.; Bhamra, R. Collaborative Multi-Robot Formation Control and Global Path Optimization. *Appl. Sci.* **2022**, *12*, 7046. https://doi.org/10.3390/app12147046.
36. Capelli, B.; Fouad, H.; Beltrame, G.; Sabattini, L. Decentralised Connectivity Maintenance with Time Delays Using Control Barrier Functions. *arXiv* **2021**, https://doi.org/10.48550/arXiv.2103.12614.
37. Ma, H. Graph-Based Multi-Robot Path Finding and Planning. *Curr. Robot. Rep.* **2022**, *3*, 77–84.
38. Chitikena, H.; Sanfilippo, F.; Ma, S. Robotics in Search and Rescue (SAR) Operations: An Ethical and Design Perspective Framework for Response Phase. *Appl. Sci.* **2023**, *13*, 1800. https://doi.org/10.3390/app13031800.
39. Govindaraju, M.; Fontanelli, D.; Kumar, S.S.; Pillai, A.S. Optimized Offline-Coverage Path Planning Algorithm for Multi-Robot for Weeding in Paddy Fields. *IEEE Access* **2023**, *11*, 109868–109884. https://doi.org/10.1109/ACCESS.2023.3322230.
40. MathWorks. *Matlab and Simulink for Engineering Systems (Mobile Robotics Simulation Toolbox)*; MathWorks: Natick, MA, USA, 2025. Available online: https://www.mathworks.com/ (accessed on 26 January 2026).
41. MathWorks®. MATLAB Central. 2025. Available online: https://www.mathworks.com/matlabcentral/fileexchange/ (accessed on 26 January 2026).
42. Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R. Conflict-Based Search for Optimal Multi-Agent Path Finding. *Artif. Intell.* **2015**, *219*, 40–66.
43. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. https://doi.org/10.1007/BF01386390.
44. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. https://doi.org/10.1109/TSSC.1968.300136.
45. Kavraki, L.E.; Švestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. https://doi.org/10.1109/70.508439.
46. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning,* Department of Computer Science, Iowa State University: Ames IA 50011, USA, 1998, https://msl.cs.illinois.edu/~lavalle/papers/Lav98c.pdf (accessed on 10 March 2026)
47. Peng, S. Multi-Robot Path Planning Combining Heuristics and Multi-Agent Reinforcement Learning. *arXiv*, **2023**, arXiv:2306.01270. https://doi.org/10.48550/arXiv.2306.01270.

48. Tang, W.; Li, C.; Wu, J.; Zhu, Q. Decentralized Communication-Maintained Coordination for Multi-Robot Exploration: Achieving Connectivity and Adaptability. In Proceedings of the 2024 IEEE/RS J International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024. Available online: https://ieeexplore.ieee.org/document/10802832 (accessed on 26 January 2026).
49. Jathunga, T.; Rajapaksha, S. Improved Path Planning for Multi-Robot Systems Using a Hybrid Probabilistic Roadmap and Genetic Algorithm Approach. *J. Robot. Control. (JRC)* **2025**, *6*, 715–733. https://doi.org/10.18196/jrc.v6i2.25572.
50. Dewangan, R.K.; Shukla, A.; Godfrey, W.W. A solution for priority-based multi-robot path planning problem with obstacles using ant lion optimization. *Mod. Phys. Lett. B* **2020**, *34*, 2050137. https://doi.org/10.1142/S0217984920501377.
51. Banik, S.; Banik, S.C.; Mahmud, S.S. Path Planning Approaches in Multi-Robot System: A Review. *Eng. Rep.* **2025**, *7*, e13035.
52. Suresh, A.; Laux, M.; Brinkmann, W.; Danter, L.C.; Kirchner, F. Enhancing Heterogeneous Multi-Robot Teaming for Planetary Exploration. *Eng. Proc.* **2025**, *90*, 112. https://doi.org/10.3390/engproc2025090112.
53. Tafrishi, S.A.; Svinin, M.; Tahara, K. A Survey on Path Planning Problem of Rolling Contacts: Approaches, Applications and Future Challenges. *arXiv* **2024**, arXiv:2501.04442. https://doi.org/10.48550/arXiv.2501.04442.
54. Chaabi, H. Distributed Multi-Robot Exploration with Connectivity Maintenance Under QoS Constraints. Ph.D. Thesis, Université de Lille, Lille, France, 2025. Available online: https://hal.science/tel-05379346 (accessed on 26 January 2026).
55. Zhou, M.; Li, J.; Wang, C.; Wang, J.; Wang, L. Applications of Voronoi Diagrams in Multi-Robot Coverage: A Review. *J. Mar. Sci. Eng.* **2024**, *12*, 1022.
56. Shaoul, Y.; Mishani, I.; Vats, S.; Li, J.; Likhachev, M. Multi-Robot Motion Planning with Diffusion Models. *arXiv* **2024**, arXiv:2410.03072. https://doi.org/10.48550/arXiv.2410.03072.
57. Zhao, L.; Chen, B.; Hu, F. Research on Swarm Control Based on Complementary Collaboration of Unmanned Aerial Vehicle Swarms Under Complex Conditions. *Drones* **2025**, *9*, 119. https://doi.org/10.3390/drones9020119.
58. Li, W.; Wang, Z.; Mai, R.; Ren, P.; Zhang, Q.; Zhou, Y.; Xu, N.; Zhuang, J.; Xin, B.; Gao, L.; et al. Modular Design Automation of the Morphologies, Controllers, and Vision Systems for Intelligent Robots: A Survey. *Vis. Intell.* **2023**, *1*, 2.
59. Jensen, J.H.; Sørensen, K.P.B.; Sejersen, J.L.F.; Sarabakha, A. Multi-Agent Path Planning in Complex Environments Using Gaus ian Belief Propagation with Global Path Finding. *arXiv* **2025**, arXiv:2502.20369.
60. Li, J.; Wang, C.; Li, B.; Wei, Q.; Song, L.; Huang, D. Maintaining Connectivity in Coverage Control: A Distributed Algebraic Connectivity Estimation Approach Using Control Barrier Functions. *Aerosp. Syst.* **2025**, 1–11. https://doi.org/10.1007/s42401-025-00424-3.
61. Bai, R.; Yuan, S.; Li, K.; Guo, H.; Yau, W.Y.; Xie, L. REALM: Real-Time Line-of-Sight Maintenance in Multi-Robot Navigation with Unknown Obstacles. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025. https://doi.org/10.1109/ICRA55743.2025.11128211.