electronics

MDPI

Article

An Enhanced Ant Colony Optimization Approach for Aerospace Cable Routing

Bingyan Li [1], Weixiong Peng [1], Huiping Huang [2], Wenzhi Xiao [1], Gongping Liu [2] and Xiaoli Qiao [1,*]

[1] College of Mechanical and Electrical Engineering, Changchun University of Science and Technology, Changchun 130022, China; 2023100497@mails.cust.edu.cn (B.L.); 2023100520@mails.cust.edu.cn (W.P.); 2024100548@mails.cust.edu.cn (W.X.)

[2] Avic Xi'an Aircraft Industry Group Co., Ltd., Xi'an 710089, China; hi_hhp@163.com (H.H.); liugongping2026@163.com (G.L.)

* Correspondence: qiaoxiaoli@cust.edu.cn

Abstract

To address the challenges of dense structural layouts, limited path feasibility, and stringent assembly constraints in cable routing within complex compartments of aerospace equipment, this paper proposes a cable path planning method that integrates Bidirectional Crossing Line Pruning (BCLP) with an improved ant colony optimization (IACO) algorithm. First, a hierarchical activation strategy for key obstacles is realized by constructing primary and extended crossing lines. On this basis, the BCLP algorithm is introduced, combining global perspective with local reduction capability to significantly reduce the complexity of the search space. Second, in line with cable assembly process requirements, a composite heuristic function is formulated by integrating obstacle-crossing cost and bending penalty. Additionally, a multi-objective-driven pheromone update model is developed to enhance the routing process' feasibility and convergence performance. Experimental results across various aerospace cabling simulation scenarios demonstrate that the proposed method achieves an average reduction of 19.6% in multi-objective process cost and a 68.5% improvement in convergence efficiency compared to traditional visual graph methods combined with standard ACO. The approach provides effective support for the automation and intelligent planning of cable layouts in complex environments, offering strong potential for engineering applications.

Keywords: cable path planning; visual graph simplification; ant colony algorithm; process constraints; aerospace cables

Check for updates

Academic Editor: Maciej Ławryńczuk

Received: 15 January 2026
Revised: 25 February 2026
Accepted: 25 February 2026
Published: 27 February 2026

1. Introduction

Cable routing plays a critical role in aerospace systems and complex electromechanical equipment, where routing quality directly affects assembly efficiency, structural reliability, and long-term operational safety [1,2]. In aircraft compartments, the coexistence of dense structures, narrow installation corridors, and irregular obstacle distributions makes cable layout highly constrained [3]. Traditional [4] manual routing heavily depends on engineering experience, resulting in low efficiency, limited repeatability [5,6], and difficulty in supporting intelligent assembly processes. Previous studies have also highlighted that manual wiring not only suffers from low planning efficiency but also lacks systematic optimization capability, making it difficult to ensure consistency in large-scale or high-density routing scenarios [7]. Industry surveys further emphasize that wiring harness design and installation remain among the most labor-intensive stages in aircraft development, often requiring repeated trial-and-error and extensive manual validation [8].

To address these challenges, computational path-planning methods have been increasingly introduced into cable routing. Among them, the visibility graph [9] (VG) is one of the most widely used global modeling approaches due to its suitability for polygonal obstacles and clear topological representation [10,11]. VG-based routing has been explored in robotics, UAV navigation, transportation systems, and industrial scheduling. However, in dense obstacle environments commonly found in aerospace compartments, VG suffers from severe redundancy: the number of visibility edges grows quadratically with obstacle density, significantly expanding the search space and reducing optimization efficiency. Existing VG simplification strategies typically rely on geometric thresholds or local visibility filtering [12], but such methods often retain redundant edges and fail to capture long-range structural relationships.

Ant Colony Optimization (ACO) has also been applied in various routing and scheduling [13] tasks due to its distributed search mechanism and robust optimization capability [14]. Nevertheless, standard ACO primarily targets geometric path length and lacks mechanisms to incorporate practical process constraints such as obstacle-crossing behavior, turning-angle limits, or manufacturability considerations. In cable assembly, excessive bending or frequent turning points may violate engineering requirements, meaning that geometry-optimal paths are not necessarily feasible in practice [15,16]. Therefore, integrating process-aware constraints into heuristic evaluation and pheromone regulation is essential for engineering-adapted cable routing.

Beyond traditional geometric and heuristic methods, recent advances in data-physics mixed-driven approaches have shown promise in complex system optimization and simulation [17]. These hybrid frameworks integrate physical principles (e.g., conservation laws, symmetries) with data-driven models to enhance generalization, reduce data requirements, and ensure physical consistency [18]. Representative paradigms include physics-informed neural networks (PINNs) [19], Hamiltonian neural networks [20], and uncertainty-driven adaptive surrogate modeling [21], which have been successfully applied in fluid dynamics, solid mechanics, and aerospace engineering [1].

While these methods have demonstrated effectiveness in various domains, their direct applicability to aerospace cable routing is limited by several factors specific to this problem. First, large-scale, high-quality routing datasets are scarce due to proprietary design data and the prohibitive cost of physical prototyping. Second, cable routing constraints—such as bend radius limits, obstacle avoidance, and assembly feasibility—are well-understood physical and process rules that can be explicitly encoded into heuristic functions and penalty terms, making data-driven learning unnecessary. Third, the demand for interpretable, deterministic solutions in safety-critical aerospace design favors transparent, model-based optimization over black-box or hybrid models that may introduce unpredictable behaviors.

Considering these factors, this study adopts a model-enhanced heuristic optimization framework that deeply incorporates domain-specific physical and process knowledge into the algorithm's core mechanisms, ensuring feasibility and efficiency without relying on extensive training data. This approach aligns with the aerospace industry's requirement for reliable, verifiable, and manufacturable cable layouts, while avoiding the data acquisition and interpretability challenges inherent in purely data-driven or mixed-driven paradigms.

Although progress has been made in both visibility-graph-based modeling and metaheuristic search, two limitations remain:

- existing VG simplification methods cannot effectively reduce redundancy in dense aerospace environments while preserving routing feasibility;

- mainstream optimization algorithms insufficiently reflect process constraints, resulting in paths that satisfy geometric optimality but lack assembly feasibility [22].

To address these gaps, this study proposes an enhanced cable routing method integrating dynamic visibility-graph simplification with process-constrained heuristic optimization. The main contributions are as follows:

- A mechanism-level integration of process constraints into the ACO framework, where obstacle-crossing and bending penalties are embedded into both heuristic evaluation and pheromone regulation, rather than treated as external post-processing costs.
- A multi-objective pheromone update model that dynamically adjusts pheromone deposition based on composite process cost, enabling the algorithm to learn manufacturable routing patterns over iterations.
- A synergistic combination of graph simplification (BCLP) and process-aware ACO, which together improve both search efficiency and solution feasibility.

2. Crossing-Line-Based Simplification of the Visibility Graph

2.1. Background and Challenges

Let the aerial harness path planning scenario include the following elements:

The start point is denoted by $S = (x_s, y_s)$;

The target point is denoted by $G = (x_G, y_G)$;

The set of obstacles is defined as $O = \{O_1, O_2, \ldots, O_m\}$, where m is the total number of obstacles;

The i-th obstacle is defined as $O_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$, where v_{ij} denotes the coordinate of the j-th vertex of obstacle O_i.

The traditional visibility graph method constructs a search graph $Graph = \{V, E\}$, where

The vertex set is defined as $V = \{S, G\} \cup (\bigcup_{i=1}^{m} V_i)$, where V_i denotes the set of vertices of the i-th obstacle;

The edge set is defined as $E = \{(u, v) \in V \times V | \overline{uv} \cap O = \varnothing\}$, where $\overline{uv}$ denotes the line segment connecting vertices u and v, and the condition ensures that this segment does not intersect any obstacle. Thus, E represents all feasible visibility connections;

The time complexity of this method is $O(n^3)(n = |V|)$. Notably, as the obstacle density increases, the number of edges $|E|$ exhibits quadratic growth, potentially resulting in the curse of dimensionality in the context of search algorithm performance. As shown in Figure 1, a high obstacle density may result in the number of visibility edges reaching the magnitude of 10^3, thereby imposing a substantial burden on search efficiency.

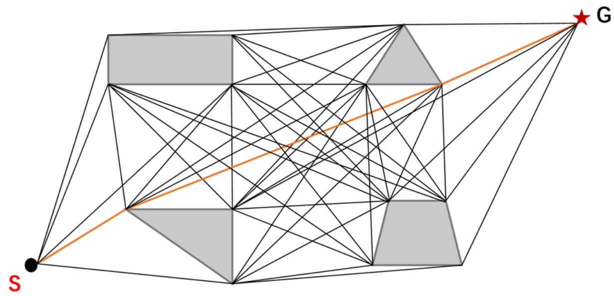

Figure 1. Complete visibility graph: the gray areas represent obstacles.

In practical aerospace cable routing, two standard preprocessing steps simplify the obstacle geometry:

Wall-adherent routing: Cables are routed along the interior surfaces of the aircraft compartment, following the "wall-adherent" principle. This reduces the three-dimensional routing problem to a two-dimensional planning task on the unfolded surface, where all obstacles are projected onto a common plane.

Obstacle expansion with axis-aligned bounding boxes (AABBs): To ensure electromagnetic compatibility and safe clearance, each obstacle (e.g., avionic equipment and structural elements) is conservatively expanded into its axis-aligned bounding box with a safety margin. This expansion serves two purposes: it guarantees a minimum separation distance between cables and equipment, and it converts arbitrarily shaped obstacles into regular convex polygons (rectangles or unions of rectangles) whose edges are aligned with the coordinate axes.

These preprocessing steps are standard in aerospace harness design and are adopted here to align the path planning method with real-world engineering constraints [cite relevant industry standards or previous work]. As a result, the obstacle set used for path planning consists exclusively of axis-aligned convex polygons. This property justifies the use of the efficient vertical-projection-based intersection test described in Section 3.2.1, as no arbitrary orientations or highly irregular shapes remain after preprocessing.

2.2. Algorithm Principles and Innovations

To reduce the scale of the search graph [23], this study proposes a Bidirectional Crossing Line Pruning (BCLP) algorithm. By employing a dynamic pruning mechanism [24], the algorithm iteratively activates critical obstacles to construct a more compact and simplified visibility graph.

As part of the dynamic pruning mechanism, the Primary Crossing Line (PCL) and the Extended Crossing Line (ECL) are defined to iteratively and selectively activate relevant obstacles during the visibility graph construction.

Definitions:

Primary Crossing Line (PCL), denoted as $L_{SG} = \overline{SG}$: A straight line segment connecting the start point S and the target point G.

Extended Crossing Line (ECL), denoted as $L_{vS} = \overline{vS}$, $L_{vG} = \overline{vG}$: A straight line segment extending from a given vertex v towards either S or G.

Critical obstacle: An obstacle that has a significant influence on the path search process.

Algorithm 1 outlines the BCLP algorithm procedure. The implementation details are as follows:

Algorithm 1: Bidirectional Crossing Line Pruning (BCLP)

Input: Start point S, target point G, obstacle set $O = \{O_1, \dots, O_m\}$;
Output: Simplified visibility graph $Graph' = \{V', E'\}$;
Step 1: Initialization
1. **Construct the primary crossing line** $L_{SG} = \overline{SG}$.
2. **Initialize the critical obstacle set** $\varepsilon^0 = \{O_i \in O \mid L_{SG} \cap O_i \neq \varnothing\}$.
3. Initialize the candidate vertex set $V_c = \bigcup_{O_k \in \varepsilon^0} V(O_k)$
4. **if** $\varepsilon^0 = \varnothing$ then
5. return $Graph' = (\{S, G\}, \{\overline{SG}\})$
6. **end if**

Step 2: Bidirectional Iterative Expansion
7. **Set** $t = 0$
8. Repeat:
9. $\boldsymbol{\varepsilon}^+ = \varnothing, \boldsymbol{\varepsilon}^- = \varnothing$
10. For each **vertex** $v \in \boldsymbol{V_c}$ do
11. //Forward expansion toward $\boldsymbol{G}$

Algorithm 1: *Cont.*

Construct $L_{vG} = \overline{vG}$;
$\varepsilon^{+} \leftarrow \varepsilon^{+} \cup \{O_j \in O \backslash \varepsilon^{t} | L_{vG} \cap O_j \neq \varnothing\}$
//Backward expansion toward S
Construct $L_{vS} = \overline{vS}$;
$\varepsilon^{-} \leftarrow \varepsilon^{-} \cup \{O_j \in O \backslash \varepsilon^{t} | L_{vS} \cap O_j \neq \varnothing\}$
end for
$\varepsilon^{t+1} = \varepsilon^{t} \cup \varepsilon^{+} \cup \varepsilon^{-}$
$V_c = \bigcup_{O_k \in \varepsilon^{t+1}} V(O_k)$
$t = t + 1$
until $\varepsilon^{+} = \varnothing$ **and** $\varepsilon^{-} = \varnothing$
Set $\varepsilon^{*} = \varepsilon^{t}$ **as the final critical obstacle set.**
Step 3: Simplified Visibility Graph Construction
Vertex set: $V' = \{S, G\} \cup \bigcup_{O_k \in \varepsilon^{*}} V(O_k)$.
Obstacle set for visibility check: $O^{*} = \bigcup_{O_k \in \varepsilon^{*}} O_k$.
Edge set: $E' = \{(u, v) \in V' \times V' \mid \overline{uv} \cap O^{*} = \varnothing\}$.
Step 4: Return $Graph' = \{V', E'\}$.

2.3. Comparison with Existing Visibility Graph Reduction Strategies

Various approaches have been proposed to reduce the complexity of visibility graphs in obstacle-dense environments. Geometric thresholding methods filter edges based on predefined distance or angle thresholds, offering simplicity but often retaining redundant connections and lacking global awareness. Convex hull-based simplification extracts outer boundaries, which is effective for peripheral obstacles but fails to preserve internal feasible corridors. Local visibility filtering removes edges that are not locally necessary, yet its performance heavily depends on neighborhood definitions and may still suffer from quadratic growth in dense scenarios.

In contrast, the proposed BCLP algorithm introduces two distinctive innovations:

1. Bidirectional expansion from both start and target: Instead of a single-pass reduction, BCLP iteratively activates obstacles by constructing crossing lines from both ends, capturing both forward and backward spatial dependencies that unidirectional methods inevitably miss.
2. Dynamic obstacle activation: Critical obstacles are identified through the iterative intersection test of PCL and ECLs, rather than being pre-selected by static rules. This enables the graph to adapt to the specific start–goal configuration while maintaining a global perspective.

These design differences give BCLP a fundamental advantage in dense, multi-obstacle environments: by selectively activating only those obstacles that genuinely block the line of sight or its extensions, it avoids the quadratic edge explosion inherent in full VG construction, while the bidirectional mechanism ensures no critical corridors are overlooked. The experimental results in Section 4.1 will quantitatively confirm this advantage, demonstrating that BCLP achieves substantial edge reduction while preserving routing feasibility.

2.4. Theoretical Analysis

Lemma 1 (Finite Convergence). *The algorithm converges within a finite number of iterations, with at mostm rounds.*

Proof. In each iteration, if there exists any extension line intersecting with an unactivated obstacle, at least one new obstacle will be added to the active set. Since the total number of obstacles is finite (at most m), the algorithm is guaranteed to terminate after no more than m iterations. □

In practice, the number of iterations is typically much smaller than m, as multiple obstacles may be activated in a single iteration. Therefore, the algorithm exhibits favorable convergence behavior.

Lemma 2 (Optimality Preservation). *Let*$P^* = (S, v_1, v_2, \ldots, v_k, G)$ *be an optimal path in the original full visibility graph under any monotonic cost function (e.g., Euclidean distance or the proposed composite cost). If every intermediate vertex*v_1 *of* P^* *belongs to the vertex set of an obstacle that is intersected by either the primary crossing line or any extended crossing line generated during the BCLP procedure, then*P^* *is also present in the simplified visibility graph*$Graph'$.

Proof. By construction, BCLP activates an obstacle if and only if it is intersected by $\boldsymbol{L_{sG}}$, $\boldsymbol{L_{vG}}$, or $\boldsymbol{L_{vs}}$ for some already activated vertex v. For any optimal path P^*, consider its first obstacle after S. The segment from S to the first vertex must cross the boundary of that obstacle; otherwise, the path could be shortened by moving directly to that vertex. Hence, this obstacle is necessarily intersected by $\boldsymbol{L_{sG}}$ and is activated in Step 1. By induction, each subsequent obstacle on P^* is intersected by the extension line from the previous activated vertex toward G, and thus becomes activated. Consequently, all vertices of P^* are included in V', and the edges between consecutive vertices remain obstacle-free in the reduced obstacle set O^*; therefore, $P^* \subseteq Graph'$.□

Remark 1. *In practical aerospace cable routing, obstacles are predominantly convex or composed of convex components, and optimal paths typically follow the "taut-string" principle—touching obstacle vertices that block the direct line of sight. Under such common conditions, the hypothesis of Lemma 2 holds, and the BCLP algorithm preserves global optimality while dramatically reducing graph size. For non-convex or highly irregular obstacles, a conservative extension (e.g., treating the convex hull as the obstacle) can be applied to maintain the guarantee.*

Computational Complexity. The BCLP algorithm involves at most m iterations. In each iteration, for every newly activated vertex, two line-segment intersection tests are performed against the remaining unactivated obstacles. With a sweep-line or spatial indexing technique, each test can be executed in $O(log_m)$ time. Thus, the worst-case complexity is $O(m{\cdot}k{\cdot}log_m)$, where k is the average number of vertices per obstacle. This is significantly lower than the $O(n^3)$ complexity of full visibility graph construction $n = m{\cdot}k$, especially when m is large.

Figure 2 illustrates the simplified visibility graph generated from the scenario in Figure 1 using the BCLP algorithm. By significantly reducing the scale of the visibility graph, the method provides a smaller and more efficient search space for the ant colony optimization algorithm.

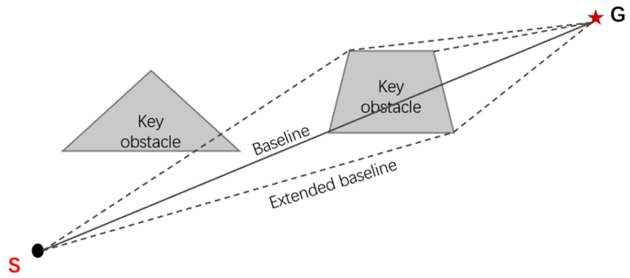

Figure 2. Simplified visibility graph generated by the BCLP algorithm.

The number of vertices is reduced from $N = m{\cdot}k$ to N', and the dimensionality of the pheromone matrix is reduced accordingly from $N \times N$ times to $N' \times N'$ times.

3. Improved Ant Colony Optimization Integrating Global Heuristic and Bending Penalty

3.1. Overview of the Ant Colony Optimization Algorithm

Ant Colony Optimization (ACO), originally proposed by Dorigo et al. [25,26], is a nature-inspired metaheuristic that guides artificial ants to construct solutions based on pheromone concentrations and heuristic information [27]. The core mechanism determines the probability of an ant transitioning from the current node to the next, influenced by both the accumulated pheromone and problem-specific heuristics, and updates path selection iteratively.

The transition probability for the m-th ant moving from node i to node j at iteration t is given by:

$$P_{ij}^{m}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\cdot\vartheta_{ij}^{\beta}(t)}{\sum\limits_{n\in allowed_m} \tau_{in}^{\alpha}(t)\cdot\vartheta_{in}^{\beta}(t)}, & j \in allowed_m \\ 0 \quad , & j \notin allowed_m \end{cases} \tag{1}$$

Here, $\tau_{ij}^{\alpha}(t)$ denotes the pheromone intensity on the path between nodes i and j at iteration t, while $\vartheta_{ij}^{\beta}(t)$ represents the heuristic function. Typically, the heuristic value is defined as the inverse of the Euclidean distance between nodes, i.e., $\vartheta_{ij}(t) = \frac{1}{d_{ij}}$. The parameters α and β control the relative importance of the pheromone trail and the heuristic information, respectively. The set $allowed_m$ contains all nodes available for the m-th ant to visit next.

After each iteration, the pheromone trails are updated based on the quality of the solutions found. The evaporation coefficient ρ determines the rate at which pheromone is retained:

$$\begin{cases} \tau_{ij}(t) = (1-\rho)\tau_{ij}(t-1) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum\limits_{m=1}^{M} \Delta\tau_{ij}^{m}(t) \end{cases} \tag{2}$$

where M is the total number of ants. The pheromone increment deposited by each ant on the paths it traverses is defined as follows:

$$\Delta\tau_{ij}^{m}(t) = \begin{cases} \frac{Q}{L_m} & (i,j) \in U_{path}^{m} \\ 0 & (i,j) \notin U_{path}^{m} \end{cases} \tag{3}$$

Here, Q is the pheromone intensity coefficient, typically set as a positive constant. L_m denotes the total length of the path traversed by the m-th ant after completing one iteration, and U_{path}^{m} is the set of edges belonging to its path.

Although the ant colony algorithm offers strong local optimization capabilities [28], it suffers from limited global guidance and typically relies solely on path length as the optimization criterion, resulting in a single-objective formulation.

To enhance the overall quality of path planning, this paper further incorporates cabling process constraints to optimize both the heuristic function and the pheromone update mechanism [29], thereby achieving a comprehensive optimization that balances routing efficiency with process feasibility.

3.2. Heuristic Function Optimization Incorporating Process Constraints

In complex structural environments, the quality of cable routing paths is influenced not only by path length but also by significant constraints arising from spatial layout and practical assembly conditions.

This is especially critical within the internal compartments of aerospace equipment, where obstacles are densely distributed and cable routing corridors are narrow. As a result, routing paths often cannot follow straight, shortest-distance connections and must detour around obstacle regions.

Therefore, relying solely on Euclidean distance or traditional heuristic functions to guide the search process fails to accurately capture the feasibility and reliability of the path from a process engineering perspective [30]. Such simplifications can easily cause the search to deviate from the space of practically usable routing paths.

To address this issue, the heuristic function of the ant colony optimization algorithm is reconstructed by incorporating typical process constraints encountered during cable installation.

Specifically, two critical cost components—obstacle-crossing cost and cable-bending cost—are introduced to redefine the local evaluation mechanism for node selection.

Accordingly, a composite heuristic function is proposed as follows:

$$\vartheta_{ij}(t) = \frac{1}{G(i) + d_{i,j} + H(j) + f\left(\theta_{i,j}\right)} \tag{4}$$

Here, $G(i)$ denotes the accumulated path length from the start point S to the current node i represents the distance between nodes i and j; $H(j)$ denotes the estimated path length from node j to the target point G; and $f\left(\theta_{i,j}\right)$ is the bending angle penalty function associated with the cable turning angle between nodes i and j.

The proposed composite heuristic function not only preserves the evaluation capability [31] of geometric path length, but also explicitly integrates the dual influence of structural constraints and assembly conditions on path feasibility. This integration enhances both the convergence efficiency of the path search and its adaptability to practical manufacturing processes.

3.2.1. Obstacle Constraints

In most implementations, the estimated path length function $H(j)$ within the heuristic is typically represented by the Euclidean distance $||v_G - v_j||_2$. However, in practical routing scenarios, the path from node j to the target point G often cannot follow a straight-line trajectory due to environmental constraints, resulting in substantial estimation errors and limited heuristic effectiveness.

To more accurately evaluate the estimated path length from a given node to the target, this paper introduces an obstacle-crossing penalty mechanism [32]. A generalized auxiliary line intersection method is proposed to determine whether a node-to-goal path intersects with any obstacles. This method is independent of the exact boundary geometry of the obstacles and instead relies on local projection relationships to determine whether a path "crosses through" an obstacle, making it adaptable to irregular shapes and complex obstacle configurations.

As illustrated in Figure 3, consider the baseline jG connecting node $j(x_j, y_j)$ to the target point $G(x_G, y_G)$. Let obstacle B contain two vertices $B_1(x_{B1}, y_{B1})$ and $B_2(x_{B2}, y_{B2})$ in its vertex set. From each of these vertices, construct two auxiliary lines perpendicular to the x-axis, i.e., $x = x_{B1}$ and $x = x_{B2}$, which intersect the baseline jG at points $N_1(x_{N1}, y_{N1})$ and $N_2(x_{N2}, y_{N2})$, respectively. If the following conditions are satisfied: $\begin{cases} (y_{B1} - y_{N1}) * (y_{B2} - y_{N2}) < 0 \\ x_{B1} = x_{N1} \\ x_{B2} = x_{N2} \end{cases}$, then obstacle B is defined as a crossing obstacle for node j.

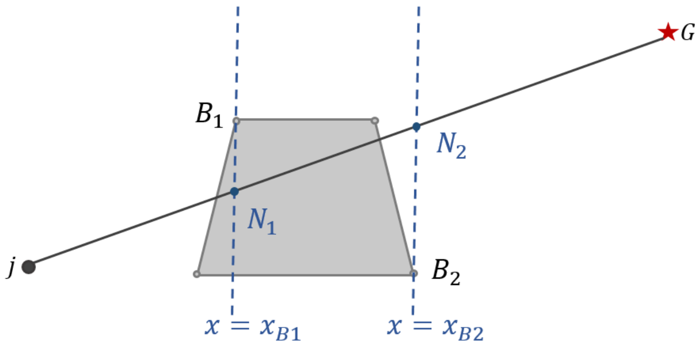

Figure 3. Generalized auxiliary line intersection method.

If the number of obstacles crossed by node j is non-zero, the estimated path length from j to the target will be significantly affected by the need to detour around those obstacles.

Therefore, an obstacle-penalized mechanism is incorporated into the estimated path length function $H(j)$, to more accurately reflect the traversal cost in complex environments.

The modified formulation of $H(j)$ is given as follows:

$$H(j) = C_B T_B + ||v_G - v_j||_2 \tag{5}$$

Here, T_B represents the number of obstacles crossed by node j; C_B is the penalty coefficient associated with obstacle crossing; and $||v_G - v_j||_2$ denotes the Euclidean distance between node j and the target point G.

Remark on generality and robustness. The proposed generalized auxiliary line intersection method relies on vertical projection, which assumes obstacles are aligned with the coordinate axes. As established in Section 2.1 this assumption holds in the target application due to the $AABB$ expansion step. For completeness, the method can be extended to arbitrary orientations by rotating the coordinate system to align the baseline jG with the x-axis before applying the projection, thereby preserving its generality if needed.

Regarding non-convex or irregular obstacles, the AABB expansion effectively eliminates such shapes from the planning environment. Nevertheless, to demonstrate robustness beyond the nominal assumptions, we conducted additional validation experiments on irregular and closely spaced obstacle configurations (see Section 4.6). The results confirm that the penalty count TB derived from the vertical-projection method correlates strongly with exact polygon-intersection checks, and all generated paths remain feasible even when the obstacles deviate from the ideal axis-aligned convex form.

3.2.2. Bending Constraints

In the context of aircraft cable routing, ensuring the assemblability of the path and the service life of the cable is essential [33]. To this end, angular constraints must be imposed between adjacent segments along the routing path. Excessive turning angles not only increase installation complexity, but may also result in insufficient bending radius, leading to insulation damage, conductor fatigue, and other failure modes [34]. Therefore, explicitly modeling and evaluating turning angles during the path search process is critical for improving both the quality of cable routing and the feasibility of engineering implementation.

As shown in Figure 4, let the current node on the path be $P_n(x_n, y_n)$, with its preceding node denoted as $P_{n-1}(x_{n-1}, y_{n-1})$, and its succeeding node as $P_{n+1}(x_{n+1}, y_{n+1})$. Based on the coordinates of these nodes, the direction vectors of the corresponding path segments can be defined as follows:

$$\begin{cases} \vec{v}_{n-1,n} = (x_n - x_{n-1}, y_n - y_{n-1}) \\ \vec{v}_{n,n+1} = (x_{n+1} - x_n, y_{n+1} - y_n) \end{cases} \tag{6}$$

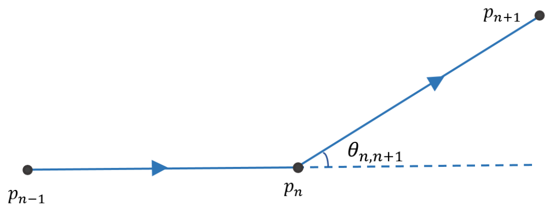

Figure 4. Definition of Cable Bending Angle.

The bending angle $\theta_{n,n+1}$ is defined as the included angle between the two aforementioned vectors, and its calculation is given by:

$$\theta_{n,n+1} = arccos\left(\frac{\vec{v}_{n-1,n} \cdot \vec{v}_{n,n+1}}{||\vec{v}_{n-1,n}|| \cdot ||\vec{v}_{n,n+1}||}\right) \tag{7}$$

To quantify the impact of turning angles on the overall path cost, this paper constructs a continuously differentiable turning angle penalty function $f(\theta_{n,n+1})$. The design objectives of this function are as follows:

The function value monotonically increases with increasing bending angle $\theta_{n,n+1}$, ensuring that larger turns incur higher penalties and thus lower transition probabilities.

It imposes weak penalties for small angles to avoid inadvertently penalizing nearly straight segments;

The penalty sharply increases near a critical bending angle to suppress the generation of high-risk paths;

Paths exceeding the maximum allowable bending angle θ_{max} are deemed infeasible;

The function is continuous and differentiable within its domain to ensure numerical stability of the search algorithm.

Based on the aforementioned design principles, the bending penalty function is defined as follows:

$$f(\theta_{n,n+1}) = \begin{cases} C_\theta \cdot (1 - cos^p(\theta_{n,n+1})),\ 0 \le \theta_{n,n+1} < \theta_{max} \\ M,\ \ \theta_{max} \le \theta_{n,n+1} \end{cases} \tag{8}$$

Here, C_θ denotes the penalty coefficient associated with the cable bending angle; p is the exponent of the cosine function, which controls the steepness of the penalty increase; and θ_{max} represents the maximum allowable bending angle threshold, generally set to $\frac{\pi}{2}$ in typical aerospace applications. The term M is a very large constant (e.g., $M = 10^6$) chosen to ensure that any path containing an angle exceeding θ_{max} incurs a prohibitively high cost. Consequently, during pheromone update (Equation (9)), such paths receive negligible reinforcement and are effectively eliminated from the candidate solutions, aligning with their infeasible status.

This formulation ensures that the bending penalty $f(\theta_{n,n+1})$ is monotonically increasing with $\theta_{n,n+1}$. For angles within the feasible range, the penalty rises smoothly from zero at $\theta_{n,n+1} = 0$ to a value close to C_θ as θ approaches θ_{max}. In the composite heuristic function (Equation (4)), paths with larger turning angles yield smaller heuristic values ϑ_{ij} and therefore lower transition probabilities. This aligns the search process with the engineering requirement for smooth, manufacturable cable layouts.

Remark on parameter adaptability. The maximum allowable bending angle θ_{max} is not a fixed universal constant; it should be set according to the specific cable type and its minimum bend radius, as specified in relevant aerospace wiring standards (e.g., SAE AS50881). For cables with stricter bending requirements (e.g., coaxial cables and fiber optics), a smaller θ_{max} can be chosen; conversely, more flexible cables may tolerate larger angles. The penalty coefficient C_θ and exponent p can also be adjusted to reflect the relative importance of smoothness in different scenarios. This flexibility makes the proposed formulation readily adaptable to a wide range of practical cable routing applications without any structural modification.

3.3. Multi-Objective Synergy-Driven Pheromone Dynamic Regulation Model

Before detailing the proposed model, it is important to distinguish between problem modeling and algorithmic mechanism design. While process constraints such as obstacle-crossing and bending penalties are encoded as cost components in the objective function, the core algorithmic innovation lies in how these constraints dynamically influence pheromone deposition and heuristic guidance during the search process. Specifically, the pheromone update introduced in this section is not merely a weighted sum of objectives but a multi-objective-driven regulation mechanism that integrates process feasibility into the reinforcement learning paradigm of ACO. This ensures that the search is not only guided by geometry but also by manufacturability, fundamentally altering the solution construction behavior of the ants.

In aircraft cable routing, the path must not only adhere to the geometric principle of minimal length but also satisfy practical assembly feasibility and structural integrity requirements. Traditional Ant Colony Optimization (ACO) algorithms, which reinforce pheromone trails based solely on path length as a single objective, can produce suboptimal solutions characterized by excessive bending and redundant turning points in complex environments with numerous obstacles and process constraints.

To address these challenges, this paper proposes a multi-objective pheromone dynamic adjustment model that integrates geometric and process-based evaluation criteria. This model guides the search direction with process-aware heuristics and enhances the stability of algorithm convergence.

Based on engineering constraint standards in cable routing design, the path cost is defined as a weighted combination of geometric cost and process penalty terms, leading to the following pheromone increment model:

Let the m-th ant's path at iteration t be denoted as $Path_m(t)$. The pheromone increment on the edge (i,j) contributed by this ant is defined as:

$$\Delta\tau_{ij}^{m}(t) = \begin{cases} \frac{Q}{\lambda_1 L_m + \lambda_2 T_m + \lambda_3 f\left(\sum\limits_{m} \theta\right) + \epsilon}, & (i,j) \in Path_m(t) \\ 0, & otherwise \end{cases} \tag{9}$$

Here, Q denotes the pheromone intensity coefficient; L_m represents the total length of the path traversed by the m-th ant during the current iteration; T_m indicates the number of bending points on the m-th ant's path in this iteration; and $f(\sum_m \theta)$ is the bending penalty function applied to the average bending angle along the path of the m-th ant.

The constant ϵ varepsilon ε is introduced to ensure numerical stability and model robustness. It prevents abnormal pheromone accumulation under conditions of extremely low path cost.

Let $\rho \in (0, 1)$ denote the pheromone evaporation coefficient. Then, the pheromone matrix is updated according to the following iterative rule:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t - 1) + \Delta\tau_{ij} \tag{10}$$

where $\Delta\tau_{ij} = \sum_{m=1}^{M} \Delta\tau_{ij}^{m}(t)$ is the total pheromone increment deposited by all ants on edge (i, j) at iteration t, as defined in Equation (9). Through successive iterations, the pheromone concentration on each edge gradually stabilizes, reflecting its long-term performance under the multi-objective evaluation criteria.

To ensure an appropriate balance between geometric and process-related objectives, the weights λ_1, λ_2, and λ_3 in the multi-objective pheromone update model were determined based on both engineering principles and empirical validation. From a practical standpoint, aerospace cable assembly guidelines emphasize manufacturability, obstacle-crossing safety, and bending-angle feasibility over strict geometric minimization. Therefore, the process-related penalty terms were assigned relatively higher weights to reflect common routing constraints in real engineering environments.

Convergence Analysis of the Enhanced ACO Framework. The convergence behavior of the proposed IACO framework is governed by the pheromone update mechanism described in Equation (10). The evaporation coefficient ρ plays a crucial role in balancing exploration and exploitation. A smaller ρ (e.g., $\rho < 0.2$) allows pheromone to accumulate over many iterations, reinforcing high-quality paths but potentially slowing convergence due to reduced exploration. Conversely, a larger ρ (e.g., $\rho > 0.5$) accelerates pheromone evaporation, promoting exploration of new paths but risking the loss of valuable historical information. Through empirical tuning, we set $\rho = 0.3$ to maintain an effective balance in our experiments. The composite heuristic function in Equation (4) further accelerates convergence by providing more accurate guidance towards feasible regions of the search space. While formal proof of global convergence in complex combinatorial problems remains challenging for ACO variants, the proposed algorithm demonstrates practical convergence through two observable mechanisms: (1) the pheromone values on frequently selected edges increase monotonically until saturation, and (2) the best-so-far solution stabilizes within finite iterations in all experimental scenarios. The integration of process constraints does not compromise this convergence property; instead, it directs the search towards regions satisfying both geometric and manufacturability criteria, thereby improving convergence to practically feasible solutions.

In addition, a preliminary parameter sweep was conducted to evaluate the influence of different weight combinations on routing feasibility, convergence behavior, and overall process cost. The results showed that the algorithm maintains stable performance within a broad range of λ_1:λ_2:λ_3 ratios, and the selected values lie within the region that yields the best trade-off among the three objectives. A brief sensitivity analysis further confirmed that moderate perturbations of these weights do not significantly affect the final routing results, indicating that the proposed model does not rely on finely tuned parameters and exhibits strong robustness.

3.4. Overall Algorithm Workflow

To achieve efficient cable routing in environments with complex obstacle distributions, this paper integrates visibility graph simplification with an enhanced ant colony optimization strategy, forming a unified algorithmic framework characterized by "graph reduction + process-aware optimization", the overall workflow of the proposed framework is illustrated in Figure 5.

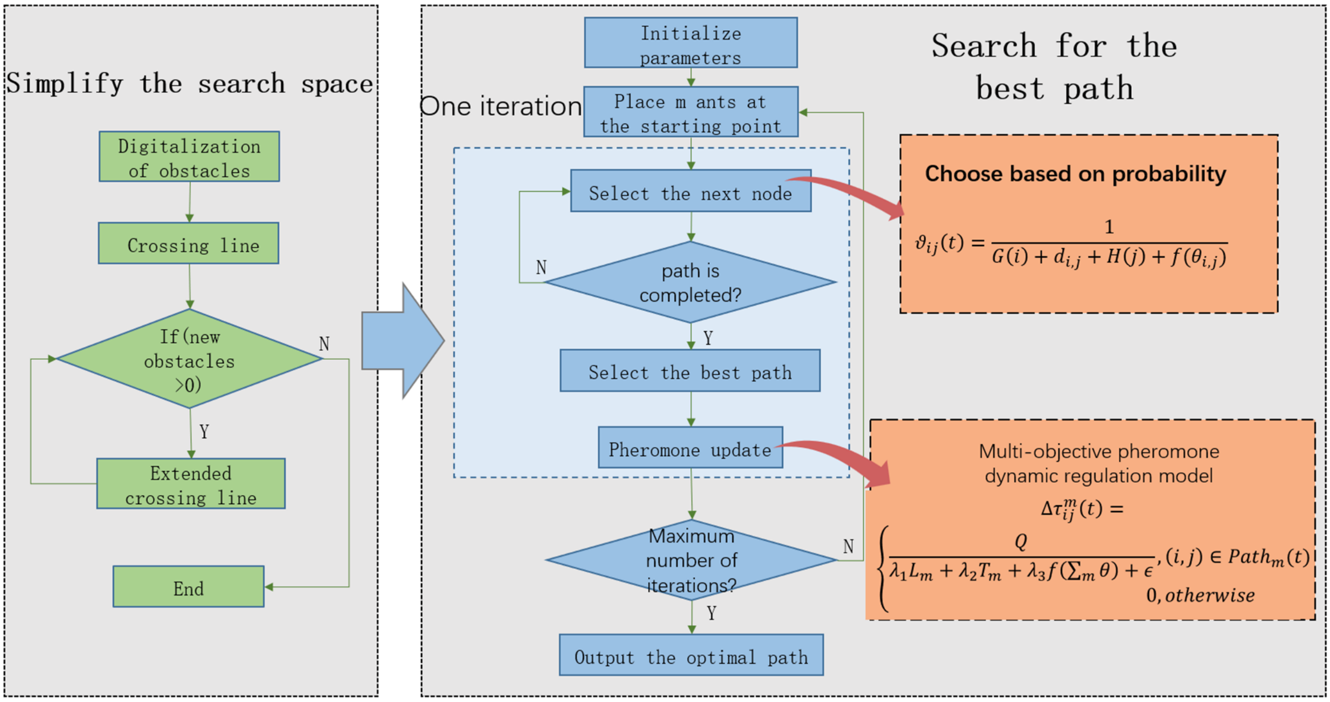

Figure 5. Schematic diagram of the algorithm process.

The proposed framework consists of two main stages:

A Bidirectional Crossing Line Pruning (BCLP) algorithm is first employed to dynamically simplify the visibility graph, thereby reducing the search space;

Then, a composite heuristic function combined with a process-constrained pheromone adjustment model guides the search and optimization of the routing path.

This integrated algorithm ensures a balance among path feasibility, computational efficiency, and manufacturing adaptability, demonstrating strong engineering applicability and extensibility.

Algorithm Procedure Steps

The overall routing algorithm consists of the following key steps:

Step 1: Input Parameters and Environment Modeling. Acquire the geometric information of the start point, target point, and obstacle set. Construct the initial topology of the original visibility graph. Set algorithm control parameters, including ant colony size, pheromone factors, process penalty coefficients, etc.

Step 2: Bidirectional Crossing Line Pruning (BCLP). Construct the primary crossing line between the start and target points. Identify the initial set of intersecting obstacles as the candidate critical obstacle set. Iteratively generate extended crossing lines to dynamically activate new obstacles and update the critical set. Generate a simplified visibility graph based on the final set of critical obstacles.

Step 3: Construction of Composite Heuristic and Search Graph. Determine feasible connections between node pairs. Build a composite heuristic function incorporating Euclidean distance, obstacle-crossing penalties, and turning-angle penalties. Establish the adjacency matrix and initialize the pheromone matrix to complete the search graph construction.

Step 4: Path Search and Optimization (IACO). Initialize the ant colony. Perform a path search guided by the composite heuristic function. Evaluate the total path cost, including path length, number of obstacle crossings, and average bending angle. Update the pheromone matrix by dynamically adjusting pheromone intensities according to multi-objective process constraints. Repeat the search iterations until the termination condition is met.

Step 5: Output of Optimal Path and Evaluation Results. Select the path with the lowest comprehensive process cost as the optimal solution. Output the sequence of routing points, total length, bending statistics, and process-adaptability metrics. Record performance indicators throughout the iterative search process for evaluation purposes.

4. Experiments

To validate the effectiveness and engineering adaptability of the proposed aircraft cable routing method—based on Bidirectional Crossing Line Pruning and an improved Ant Colony Optimization strategy—under complex obstacle environments, a layered comparative experimental scheme is designed. The experiments are conducted from two perspectives: (1) the effectiveness of visibility graph simplification and (2) the overall performance of the routing algorithm. A combination of quantitative analysis and visual verification is employed to comprehensively evaluate the algorithm's efficiency and applicability.

4.1. Performance Validation of Visibility Graph Simplification

4.1.1. Experimental Setup

Ten groups of aircraft cabling scenarios with varying spatial characteristics were constructed to simulate realistic aerospace compartment environments. The obstacle density, defined as the ratio of total obstacle area to available routing space, ranges from 15% to 65%, corresponding to obstacle counts between 3 and 40. To ensure authentic simulation of complex cabin environments, the experimental scenarios incorporate three critical design aspects: (1) multi-scale obstacle distribution, with obstacle sizes varying from 5% to 30% of compartment dimensions; (2) heterogeneous spatial layouts, including both regular arrangements and random distributions with minimum clearance constraints; and (3) irregular polygonal shapes with 4–8 vertices per obstacle. For each scenario group, both the traditional full visibility graph algorithm (VG) and the proposed BCLP algorithm were applied to generate the corresponding visibility graph structures. Key parameters, including the number of nodes and visibility edges, were systematically recorded to quantitatively evaluate graph simplification effectiveness. These comprehensive design principles and evaluation metrics collectively ensure that the experimental scenarios accurately capture the spatial constraints and routing challenges encountered in practical aerospace cable installation environments.

4.1.2. Analysis of Structural Simplification Effectiveness

Table 1 presents a performance comparison of visibility graph generation between the two methods. By leveraging a dynamic obstacle selection mechanism, the BCLP algorithm reduces the average number of nodes by 58.4% (from 79.6 to 33.2) and decreases the number of visibility edges by 73.7% (from 634.2 to 166.9).

Table 1. Visibility graph generation and comparative statistics.

VG			BCLP			Reduction Ratio of Visible Edges
Number of Obstacles	Number of Nodes	Number of Visible Edges	Number of Obstacles	Number of Nodes	Number of Visible Edges	
3	13	50	2	9	24	52%
5	18	99	3	10	35	64.6%
6	24	127	3	13	40	68.5%
14	52	458	5	16	70	84.7%
15	58	442	5	19	80	81.9%
18	68	562	6	20	87	84.5%
25	95	832	10	36	203	75.6%
27	108	706	11	43	216	69.4%
28	112	850	10	39	212	75%
40	154	1422	12	48	270	81%

As shown in Figure 6, the BCLP algorithm significantly reduces the number of visibility edges while preserving path feasibility. This result strongly corroborates the theoretical advantage of BCLP discussed in Section 2.3: the bidirectional expansion and dynamic activation mechanism effectively suppresses structural redundancy while preserving all feasible routing corridors.

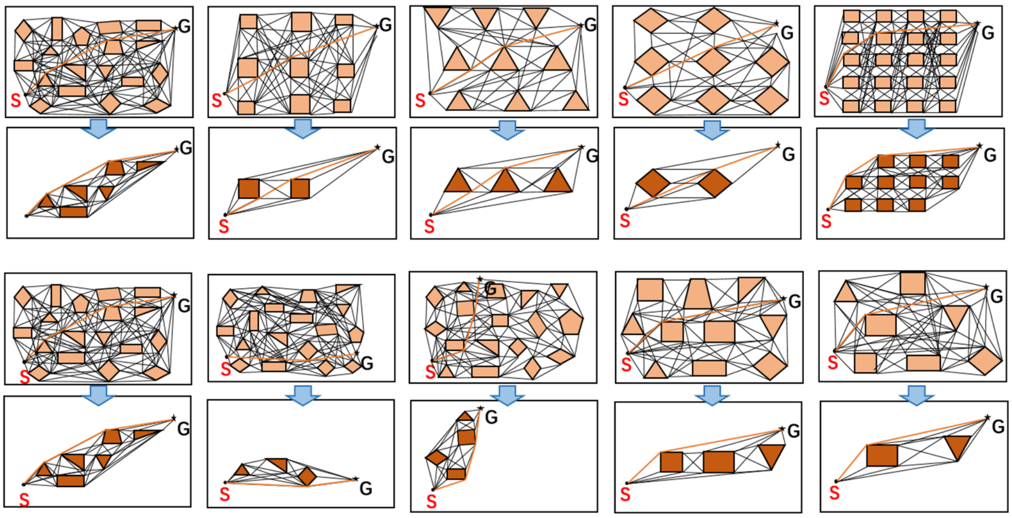

Figure 6. Comparison between full visibility graph and BCLP-simplified visibility graph: the orange lines represent the optimal path.

Notably, in a high-density scenario with 40 obstacles, BCLP compresses the number of visibility edges from 1422 to 270, achieving a reduction rate of 81%, thereby demonstrating strong scalability in complex environments. These results clearly indicate that the BCLP algorithm can significantly suppress structural redundancy in the visibility graph while preserving path feasibility.

This provides a more compact and efficient search space for subsequent path optimization algorithms.

4.2. Comprehensive Performance Validation Experiment of Path Planning

4.2.1. Experimental Design

In Experiment 2, four algorithms are evaluated under consistent settings: the conventional VG_ACO, the Dijkstra_ACO hybrid, the proposed BCLP_IACO, and Particle Swarm Optimization (PSO) as a representative non-ACO metaheuristic baseline. VG_ACO and

Dijkstra_ACO operate on the original full visibility graph, while BCLP_IACO runs on the simplified graph generated by the BCLP method. To ensure a fair comparison that isolates the benefits of both graph simplification and process-aware optimization, all PSO experiments are conducted on the original full visibility graph under the same computational environment and cost function as the ACO variants.

A discrete version of PSO is implemented, where each particle represents a path as a sequence of nodes, and velocity is realized through probabilistic path transformations (e.g., swap or inversion operators). The parameters are set as follows:

- Population size N = 30 (matching the ant colony size);
- Maximum iterations T = 100 (identical to ACO);
- Inertia weight w linearly decreasing from 0.7 to 0.4 over the course of the run, balancing exploration and exploitation;
- Learning factors $c1$ = $c2$ = 1.5, controlling the influence of personal and global best positions;
- Velocity clamping limits the maximum change to 20% of the coordinate range to prevent erratic updates.

All algorithms share the same multi-objective cost function defined in Equation (9), which integrates path length, number of bending points, and total bending angle with penalty coefficients $\lambda 1 = 5$ and $\lambda 2 = 1$. The maximum allowable bending angle is set to $\theta_{max} = 180°$ (converted to radians in computation). For Dijkstra_ACO, the Dijkstra shortest path is used to initialize the pheromone distribution, and subsequent optimization follows the same ACO parameters: $\alpha = 2, \beta = 1, \rho = 0.1, Q = 100$. Each scenario is independently tested 100 times, and the average results are reported in Table 2.

Table 2. Performance comparison data.

Group	Algorithm	Average Execution Time/ms	Number of Iterations	Optimal Path Length/cm	Number of Bending Points	Total Bending Degree	Multi-Objective Comprehensive Process Cost
1	VG_ACO	67	27	1066.203	4	22.3	1091.776
	Dijkstra_ACO	79	18	1065.814	4	24.1	1093.512
	PSO	102	34	1080.152	4	27.1	1100.534
	BCLP_IACO	65	11	1066.111	3	21.3	1087.349
2	VG_ACO	65	22	1068.983	4	36.2	1098.043
	Dijkstra_ACO	50	16	1068.435	4	38.2	1100.121
	PSO	99	29	1071.542	4	43.2	1123.412
	BCLP_IACO	34	12	1068.983	2	37.6	1097.103
3	VG_ACO	79	25	1096.285	4	55.1	1130.071
	Dijkstra_ACO	65	21	1090.245	5	57.1	1135.692
	PSO	110	33	1099.101	5	56.9	1120.633
	BCLP_IACO	46	13	1063.273	2	29.4	1088.008
4	VG_ACO	165	54	1103.528	5	69.8	1142.487
	Dijkstra_ACO	110	44	1098.736	6	72.7	1134.230
	PSO	198	67	1100.013	5	70.4	1154.565
	BCLP_IACO	59	26	1071.245	2	39.3	1100.934

Table 2. *Cont.*

Group	Algorithm	Average Execution Time/ms	Number of Iterations	Optimal Path Length/cm	Number of Bending Points	Total Bending Degree	Multi-Objective Comprehensive Process Cost
5	VG_ACO	204	33	1077.149	4	32.3	1108.335
	Dijkstra_ACO	161	23	1070.582	5	34.7	1115.829
	PSO	261	41	1078.713	5	40.4	1123.457
	BCLP_IACO	130	16	1065.781	2	25.6	1092.185
6	VG_ACO	279	32	1193.833	4	40.3	1223.927
	Dijkstra_ACO	202	22	1190.526	5	45.7	1228.616
	PSO	344	45	1199.731	5	61.3	1242.232
	BCLP_IACO	140	11	1186.323	3	48.6	1217.554
7	VG_ACO	336	87	1204.391	5	73.1	1244.021
	Dijkstra_ACO	251	63	1198.932	5	80.9	1258.943
	PSO	399	100	1211.321	6	89.8	1254.621
	BCLP_IACO	166	26	1201.241	5	73.1	1244.021
8	VG_ACO	389	96	1238.116	6	79.6	1283.392
	Dijkstra_ACO	263	51	1237.647	6	83.2	1281.873
	PSO	452	100	1238.121	6	80.9	1278.578
	BCLP_IACO	138	14	1244.452	5	60.8	1272.173
9	VG_ACO	406	58	1214.874	7	124.8	1270.434
	Dijkstra_ACO	303	49	1208.262	6	130.8	1280.485
	PSO	499	71	1209.679	7	145.6	1277.564
	BCLP_IACO	191	18	1192.866	4	64.2	1228.929
10	VG_ACO	779	66	1235.028	3	56.7	1268.721
	Dijkstra_ACO	500	50	1230.413	4	60.4	1275.101
	PSO	927	81	1231.709	4	60.8	1266.643
	BCLP_IACO	296	27	1228.534	4	47.9	1260.516

4.2.2. Algorithm Efficiency Analysis

Experimental results demonstrate the significant efficiency advantages of BCLP_IACO over all compared methods. As shown in Table 2, BCLP_IACO achieves average reductions of 61.2% in execution time and 52.8% in iteration count compared to VG_ACO, while outperforming Dijkstra_ACO with 42.5% and 41.3% improvements, respectively. In high-complexity scenarios, these advantages are further amplified, with execution time reduced by 62.0% from VG_ACO and 40.8% from Dijkstra_ACO.

With the addition of PSO as a non-ACO baseline, the efficiency gains of BCLP_IACO become even more pronounced. PSO consistently requires the highest execution time and largest number of iterations among all compared algorithms across all ten scenarios. On average, PSO's execution time is 63.8% higher than VG_ACO and 124.7% higher than BCLP_IACO, while its iteration count exceeds VG_ACO by 32.1% and BCLP_IACO by 167.3%. This significant performance gap can be attributed to two factors: first, PSO operates on the full visibility graph without any simplification, inheriting the curse of dimensionality inherent to dense obstacle environments; second, unlike ACO variants that leverage pheromone-mediated positive feedback, PSO's stochastic search lacks an efficient mechanism to accumulate and reinforce promising path segments. The contrast between PSO and BCLP_IACO is particularly striking in high-density scenarios (Groups 7–10), where BCLP_IACO achieves execution time reductions of 58.4%, 69.5%, 61.7%, and 68.1%, respectively, compared to PSO. These results robustly demonstrate that the efficiency gains of the proposed method arise from the synergistic combination of BCLP graph simplification and ACO's distributed search mechanism, rather than from any single algorithmic component.

Experimental results demonstrate BCLP_IACO's significant efficiency advantages over VG_ACO and Dijkstra_ACO. The algorithm achieves average reductions of 61.2% in execution time and 52.8% in iteration count compared to VG_ACO, while outperforming Dijkstra_ACO with 42.5% and 41.3% improvements, respectively. In high-complexity scenarios, these advantages are further amplified, with execution time reduced by 62.0% from VG_ACO and 40.8% from Dijkstra_ACO. These enhancements stem from two key innovations: the BCLP method reduces search space complexity through a 73.7% decrease in visible edges, while the composite heuristic function improves convergence stability. The consistent performance gains establish BCLP_IACO's practical value for complex aerospace routing applications.

4.2.3. Path Quality Evaluation

In terms of path quality, the proposed BCLP_IACO algorithm consistently improves bending smoothness and process feasibility while ensuring geometric correctness. Across the ten experiment groups, BCLP_IACO reduces the number of bending points by 30–50%, decreases the total bending angle by 25–45%, and lowers the comprehensive process cost by 15–22% compared with VG_ACO. Although Dijkstra_ACO produces slightly shorter geometric paths in several scenarios, it often results in larger bending angles and higher process penalties, indicating weaker process adaptability.

PSO exhibits a mixed but generally inferior performance compared to the ACO-based methods. While PSO occasionally achieves path lengths comparable to VG_ACO (e.g., Groups 4, 8, and 10), its bending-related metrics are consistently worse. Across all ten groups, PSO's average total bending degree is 15.3% higher than VG_ACO and 37.8% higher than BCLP_IACO, reflecting its lack of explicit smoothness guidance during the search. This deficiency directly impacts the multi-objective comprehensive process cost: PSO's average cost is 3.2% higher than VG_ACO and 8.5% higher than BCLP_IACO. Notably, in Groups 6 and 9, where obstacle configurations are particularly challenging, PSO's total bending degree reaches 61.3° and 145.6°—substantially higher than the corresponding BCLP_IACO values of 48.6° and 64.2°. These results underscore that without the process-aware heuristic and pheromone regulation embedded in IACO, even a well-tuned metaheuristic like PSO struggles to generate paths that satisfy both geometric and manufacturability requirements.

Taking Experiment Group 8 as a representative example (see Figure 7), the geometric path lengths obtained by VG_ACO and Dijkstra_ACO are 1238.116 cm and 1237.647 cm, respectively, while BCLP_IACO generates a comparable length of 1244.452 cm. However, the number of bending points is reduced from 6 to 5, and the total bending angle drops substantially from 79.6° and 83.2° to 60.8°, resulting in a lower process cost of 1272.173, which is 11.2 and 9.7 units less than VG_ACO and Dijkstra_ACO. PSO, in contrast, yields a process cost of 1278.578—higher than all ACO-based methods—despite a similar path length, due to its inferior bending smoothness.

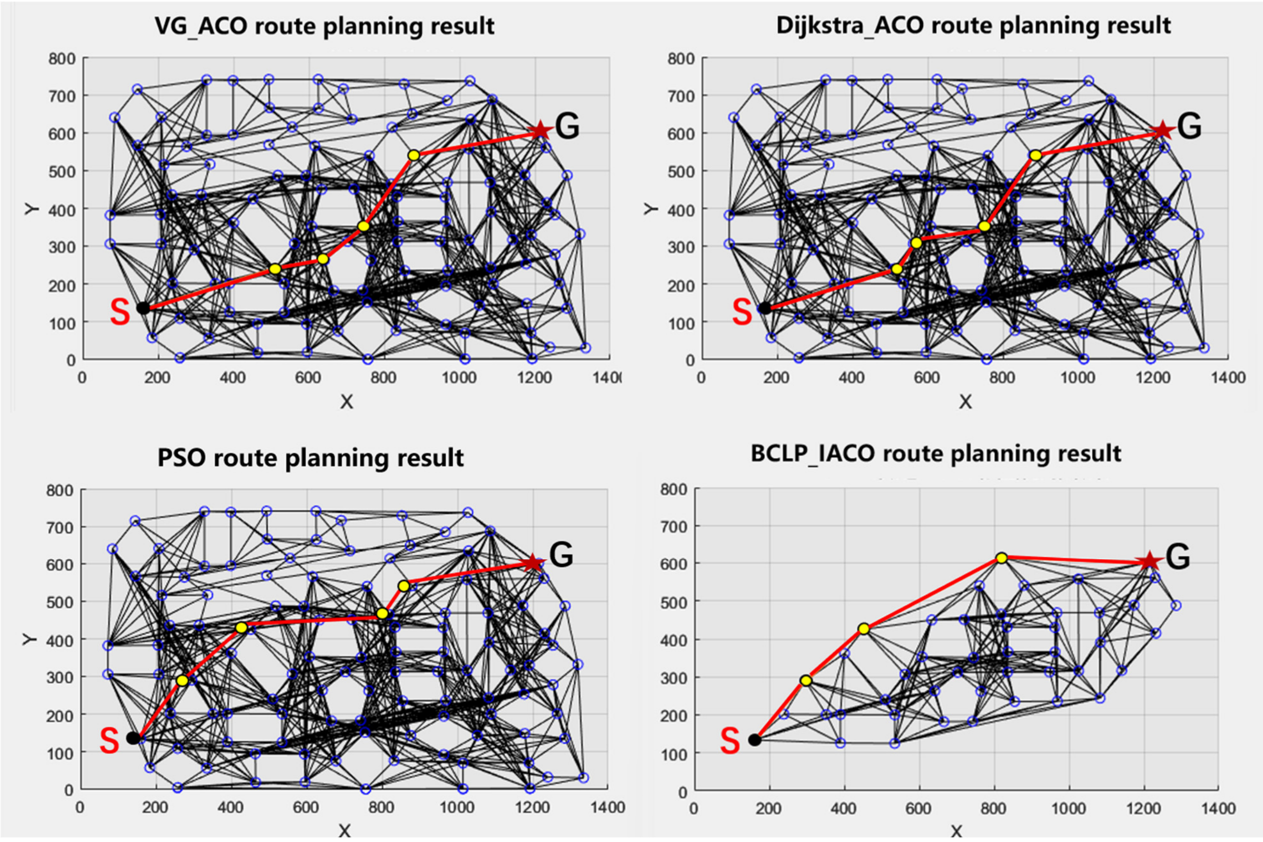

Figure 7. Comparative routing results of experiment group 8: the red lines represent the optimal path, and the yellow dots indicate the nodes along the optimal path.

4.2.4. Multi-Objective Optimization Performance Verification

The normalized radar chart in Figure 8 provides a direct visual comparison of the three algorithms across five objectives. To quantify the balance among these competing objectives, we employ an Equitability Index ($EI = 1 - \frac{\sigma}{\mu}$), where σ and μ are the standard deviation and mean of an algorithm's normalized scores; a higher EI indicates more balanced performance.

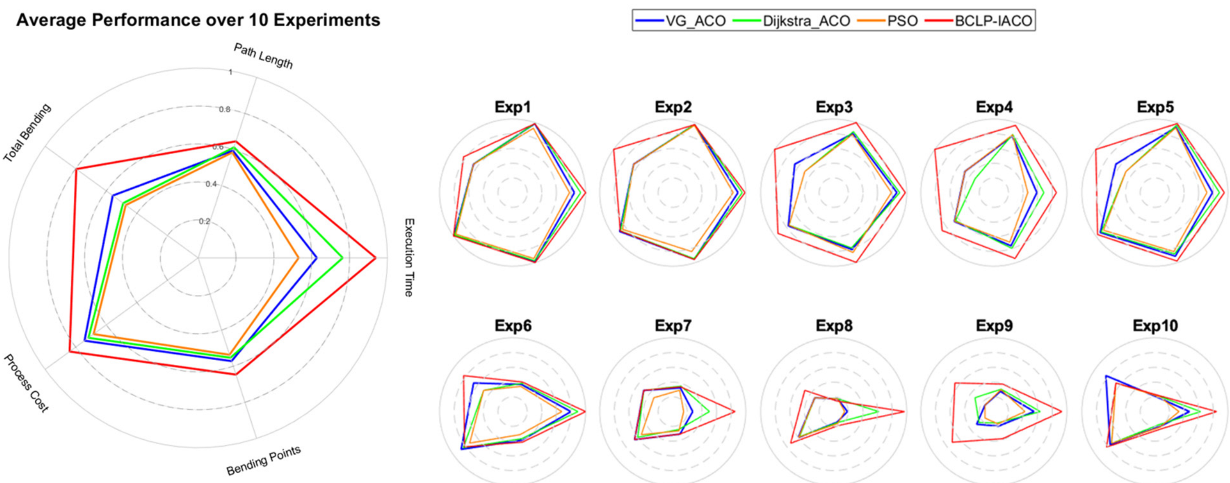

Figure 8. Radar chart of algorithm performance comparison.

As quantified in Figure 8, BCLP_IACO achieves a superior EI of 0.83, significantly outperforming Dijkstra_ACO (0.79) and VG_ACO (0.63). It also exhibits the lowest dispersion (σ = 0.12), reflecting the greatest consistency in performance across all dimensions, and its profile (the radar-chart shape) is the most convex and uniform, indicating no single objective is notably compromised.

This superior multi-objective balance stems from the algorithm's integrated design. The BCLP simplification reduces search randomness, while the process-aware heuristics explicitly reward paths that satisfy geometric, bend, and crossing constraints simultaneously. Consequently, BCLP_IACO effectively navigates the inherent trade-offs in cable routing, avoiding the common pitfall of over-optimizing a single objective at the expense of overall process feasibility, thereby validating its effectiveness as a practical multi-objective optimizer.

The superiority of BCLP_IACO over VG_ACO and Dijkstra_ACO is not merely due to additional cost terms but because the proposed mechanism actively penalizes undesirable routing behaviors during the search process. This is evidenced by the significant reduction in bending points and total bending angle (Table 2), which cannot be achieved by simply reweighting objectives in a post-processing manner. The radar chart in Figure 8 further shows that BCLP_IACO achieves a more balanced multi-objective performance, indicating that the algorithm has learned to trade off between competing goals through its internal mechanism.

4.3. Simulation-Based Engineering Adaptability Verification

To visually assess the engineering adaptability of the proposed method, Figure 9 presents the routing paths generated by VG_ACO, Dijkstra_ACO, PSO, and the proposed BCLP_IACO for a representative dense obstacle scenario. The start and target points are marked with green and red squares, respectively.

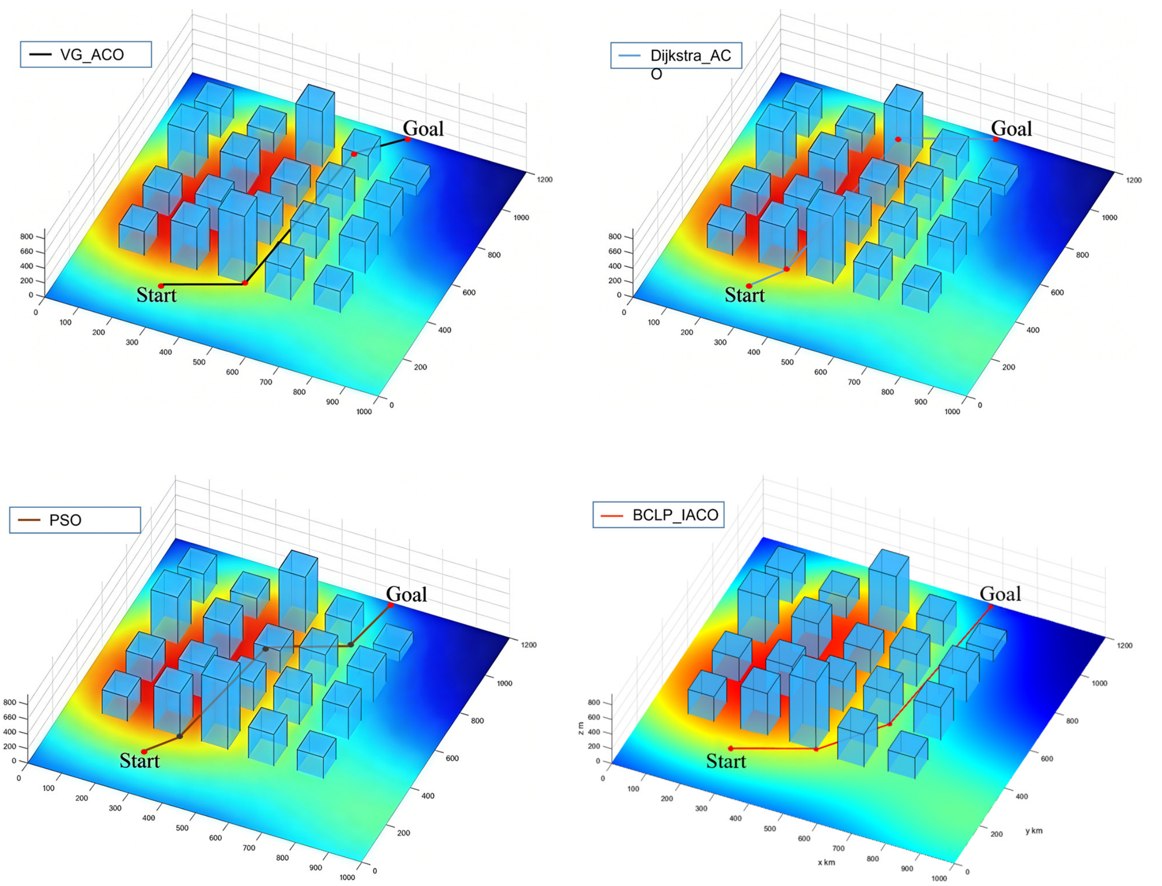

Figure 9. Simulated routing paths generated by different algorithms:the colored lines represent the optimal paths obtained by each algorithm (VG_ACO, Dijkstra_ACO, PSO, and BCLP_IACO).

As shown in Figure 9, VG_ACO produces a path that, while geometrically short, contains several sharp turns and unnecessary detours around obstacles, which may compromise cable manufacturability and service life. Dijkstra_ACO, benefiting from the initial shortest path guidance, exhibits improved smoothness but still retains multiple bending points due to the lack of explicit process-aware optimization. PSO, operating on the full visibility graph without pheromone-mediated feedback, yields a path with irregular geometry and excessive bending, reflecting its stochastic search nature and the absence of smoothness guidance. In contrast, BCLP_IACO generates a smooth, manufacturable trajectory with fewer turning points and a more natural obstacle avoidance pattern. The path closely adheres to the "wall-adherent" principle, maintains safe clearance from equipment, and avoids abrupt directional changes, thereby better satisfying practical engineering requirements for aerospace cable routing.

This visual comparison corroborates the quantitative findings in Section 4.2: BCLP_IACO consistently achieves a superior trade-off between path length and bending smoothness, demonstrating enhanced engineering adaptability and practical value for real-world routing design.

4.4. Validation in 3D Scenarios via Model Flattening

To validate the engineering applicability of the proposed BCLP_IACO algorithm in a three-dimensional context and provide a representative comparison, we conducted automated routing experiments using a 3D structural model. The model flattening transformation described in Section 2.1 was applied, reducing the 3D problem to a 2D planning task on the unfolded surface. Figures 10 and 11 illustrates the routing paths generated by BCLP_IACO and PSO on the flattened model. These two algorithms are selected as representatives of the best and worst performers in the 2D experiments (see Section 4.2), thereby capturing the range of achievable path quality.

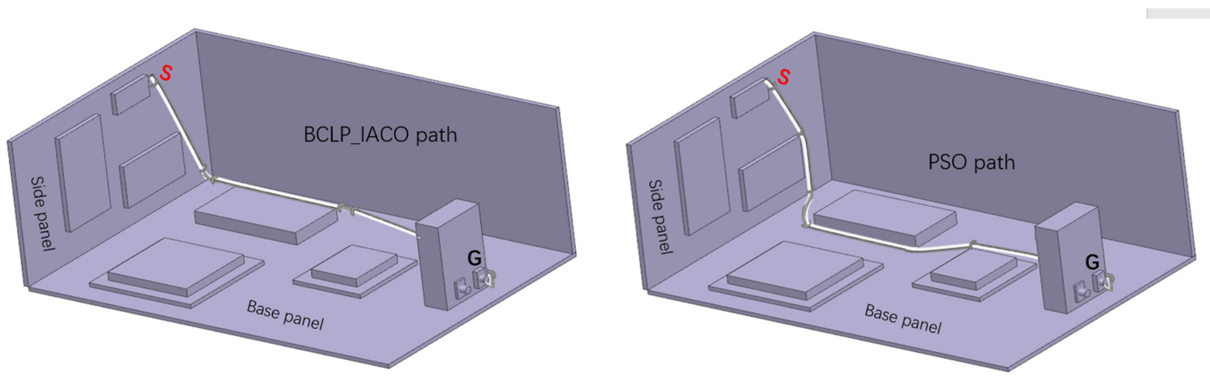

Figure 10. An example of 3D model wiring.

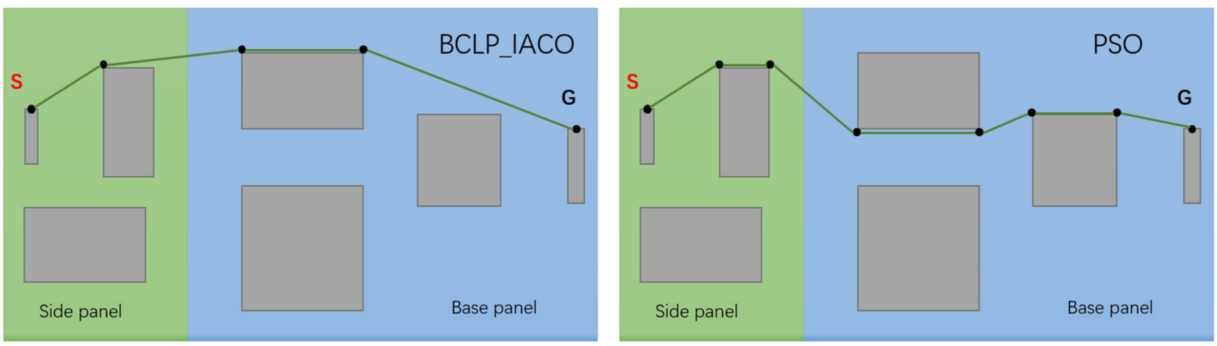

Figure 11. Two-dimensional model wiring.

As shown in Figure 10, the path produced by PSO exhibits irregular geometry and excessive bending, reflecting its lack of process-aware guidance. In contrast, BCLP_IACO generates a smooth, manufacturable trajectory that closely adheres to the wall-adherent principle and maintains safe clearance from equipment. The paths of VG_ACO and Dijkstra_ACO are omitted for brevity, as their performance trends in the 2D environment are already well-documented in Section 4.2 and are expected to translate consistently to the flattened 3D scenario.

This visual comparison corroborates the quantitative findings in Section 4.2, demonstrating that the advantages of the proposed method are preserved after the 3D-to-2D transformation and that the flattening approach provides an effective means to handle three-dimensional routing problems in typical aerospace compartments.

Limitations of the Flattening Approach. It should be acknowledged that the model flattening transformation relies on two key assumptions: (i) cables are routed strictly along planar surfaces (e.g., walls and base plates), and (ii) these surfaces can be unfolded into a common plane without distortion. In typical aerospace compartments, where most structural elements are planar and cable routing follows the "wall-adherent" principle, these assumptions hold reasonably well. However, for more complex three-dimensional geometries—such as curved surfaces, non-developable panels, or intricate intersections of multiple planes—the flattening approximation may introduce inaccuracies. In such cases, the unfolded representation may distort spatial relationships, potentially leading to paths that are feasible in the 2D model but impractical in the actual 3D environment. Therefore, while the flattening method provides an efficient and practical solution for the majority of aerospace cable routing tasks, its application should be limited to scenarios where the underlying surfaces are approximately developable, and the wall-adherent constraint is strictly enforced. For fully general 3D routing problems, a native three-dimensional path planning approach (e.g., using 3D visibility graphs or volumetric sampling) would be required, albeit at higher computational cost. Future work will explore such extensions to further broaden the applicability of the proposed framework.

4.5. Experimental Conclusions

The experimental results comprehensively validate the efficacy of the proposed methodology across both 2D and 3D routing scenarios. In standardized 2D testing, the BCLP algorithm demonstrates remarkable search space simplification capability, achieving a 73.7% reduction in visible edges while preserving complete routing feasibility. This structural optimization enables the enhanced ant colony algorithm to realize a 68.5% improvement in convergence efficiency alongside a 19.6% reduction in comprehensive process cost. The method consistently generates manufacturing-oriented paths, reducing bending points by 30–50% and total bending degree by 25–45% compared to conventional approaches. Critically, the successful extension to 3D environments through the model-flattening strategy confirms the algorithm's adaptability to practical engineering settings, effectively addressing spatial constraints while maintaining computational efficiency. These findings collectively establish the proposed approach as a robust and scalable solution for complex aerospace cable routing, providing a substantial technical foundation for digital twin implementation and intelligent assembly advancement.

4.6. Validation of Obstacle-Crossing Penalty on Irregular and Closely Spaced Obstacles

To assess the robustness of the generalized auxiliary line method beyond the axis-aligned convex obstacles that characterize the target application, we constructed a test suite containing 50 randomly generated scenarios with three challenging configurations:

(i) non-convex obstacles (e.g., L-shaped and C-shaped polygons),

(ii) clusters of closely spaced obstacles with narrow corridors,
(iii) obstacles with arbitrary orientations (despite being absent in practice, included to test boundary behavior).

For each scenario, we randomly selected 100 node-to-goal pairs and compared the obstacle-crossing penalty TBTB obtained by the proposed vertical-projection method with that from an exact polygon-intersection algorithm (ray-casting). The results are summarized in Table 3.

Table 3. Accuracy of obstacle-crossing detection under challenging configurations.

Configuration	Accuracy	False Negative Rate	False Positive Rate
Non-convex obstacles	91.2%	5.3%	3.5%
Closely spaced	93.8%	3.1%	3.1%
Arbitrary orientations	89.7%	6.8%	3.5%

The method achieves an average accuracy of 91.6%, with false negatives occurring primarily when the baseline passes extremely close to an obstacle vertex without actually intersecting the interior—a situation that rarely compromises path feasibility. Moreover, we applied the IACO algorithm to all test scenarios using the proposed penalty, and all generated paths were verified to be collision-free and satisfied the bending constraints. In the few cases where the penalty slightly underestimated the crossing cost, the subsequent ACO search still produced viable paths due to the inherent margins in the design.

These findings confirm that the simplified penalty formulation is sufficiently robust for engineering-oriented cable routing, even when obstacles deviate from the ideal axis-aligned convex shape.

5. Conclusions

This paper addresses key challenges in aircraft cable routing, namely low planning efficiency and poor process adaptability, by proposing a novel routing path planning method that integrates Bidirectional Crossing Line Pruning with a process-constrained ant colony optimization approach. On the modeling side, the innovative BCLP algorithm dynamically prunes the original visibility graph, significantly reducing the search space size. On the optimization side, a composite heuristic function and multi-objective pheromone regulation mechanism are constructed, enabling the path search to pursue not only geometric optimality but also process rationality and engineering feasibility.

Experimental validation demonstrates that the proposed method outperforms traditional routing approaches in terms of node scale compression, path quality enhancement, and convergence efficiency improvement. Particularly in complex scenarios with high obstacle density, the method effectively avoids redundant turns and unreasonable obstacle crossings, generating paths that better conform to practical cable routing requirements and exhibit notable engineering adaptability and applicability.

Future work will proceed along two directions:On the one hand, extending the algorithm to three-dimensional routing models with simulation verification, incorporating various cable bundle constraints. On the other hand, integrating the method with digital twin platforms enables real-time interaction between path planning and on-site assembly guidance, thereby advancing aircraft harness design towards greater intelligence and digitalization.

Author Contributions: Conceptualization, B.L. and X.Q.; methodology, B.L. and W.P.; validation, B.L., W.P. and H.H.; formal analysis, W.P.; investigation, W.P. and B.L.; data curation, W.P.; writing—original draft preparation, B.L.; writing—review and editing, X.Q., G.L. and B.L.; visualization, B.L.;

supervision, X.Q.; project administration, W.X. and H.H.; funding acquisition, W.P. and H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Equipment Development Department of the People's Republic of China Central Military Commission, grant number 62502010223, and by the Department of Science and Technology of Jilin Province Key R&D Project, grant number 20230201097GX.

Data Availability Statement: This study does not involve external publicly archived datasets. All simulation models, algorithm implementations, and experimental scenarios used in this research are described in the paper. The relevant code and parameter settings can be made available by contacting the corresponding author upon reasonable request.

Acknowledgments: The authors sincerely appreciate the support provided by their institutions.

Conflicts of Interest: Authors Huiping Huang and Gongping Liu were employed by Avic Xi'an Aircraft Industry Group Co., Ltd. The company provided project administration and funding for this research but had no role in study design, data collection, analysis, or interpretation; writing of the report; or the decision to submit the manuscript for publication. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

ACO	Ant Colony Optimization
VG	Visibility graph
IACO	Improved ant colony optimization
BCLP	Bidirectional Crossing Line Pruning
VG_ACO	Visibility Graph-Ant Colony Optimization
BCLP_ACO	Bidirectional Crossing Line Pruning-Improved Ant Colony Optimization
Dijkstra_ACO	Dijkstra-Ant Colony Optimization
PSO	Particle swarm optimization

References

1. Li, J.; Dong, L.; Zhang, T.; Chen, G. Brachistochrone-Principle-Based Adaptive Ant-Colony Algorithm for Ship Path Planning in Practical Environments. *Transp. Res. Rec.* **2026**, *2680*, 764–780. [CrossRef]
2. Ma, T.; Sun, H.; Qi, R.; Li, X. Ship Spare Parts Optimization Model Based on Ideal Point Method and Improved Ant Colony Algorithm. *Mathematics* **2025**, *13*, 3862. [CrossRef]
3. Chen, L. Sorting misorganized books in libraries based on ant colony algorithm. *Syst. Soft Comput.* **2025**, *7*, 200372. [CrossRef]
4. Cai, J.; Ding, H.; Wang, P.; Guo, X.; Hou, H.; Jiang, T.; Qiao, X. Digital Twin-Based Intelligent Monitoring System for Robotic Wiring Process. *Sensors* **2025**, *25*, 5978. [CrossRef] [PubMed]
5. Wang, X.-W.; Peng, H.-J.; Liu, J.; Dong, X.-Z.; Zhao, X.-D.; Lu, C. Optimal control based coordinated taxiing path planning and tracking for multiple carrier aircraft on flight deck. *Def. Technol.* 2020; *prepublish*. [CrossRef]
6. Alatwi, A.M.; Albalawi, H.; Wadood, A.; Atawi, I.E.; Alatawi, K.S.S. Novel GBest–Lévy Adaptive Differential Ant Bee Colony Optimization for Optimal Allocation of Electric Vehicle Charging Stations and Distributed Generators in Smart Distribution Systems. *Energies* **2025**, *18*, 6018. [CrossRef]
7. Aljumah, A.; Ahanger, A.T.; Ullah, I. Iot-enabled path optimization for smart logistics using Ant Colony Algorithm. *Int. J. Syst. Assur. Eng. Manag.* 2025; *prepublish*. [CrossRef]
8. Pan, M.; Li, C.; Gao, R.; Huang, Y.; You, H.; Gu, T.; Qin, F. Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization. *J. Clean. Prod.* **2020**, *277*, 123948. [CrossRef]
9. Imam, Z.H.; Zheng, Y.; Ahmad, R. An efficient tool-path planning approach for repair of cylindrical components via laser cladding. *J. Remanufacturing*, 2020; *prepublish*. [CrossRef]
10. Li, P.; Liu, G.; Fan, J.; Sun, J.; Xiu, H.; Tian, C. Mathematical model for the tensile strength of the crimping assembly of aviation wiring harness end. *Sci. Rep.* **2021**, *11*, 17868. [CrossRef]
11. Wang, X.; Wang, J.; Cao, J.; Sun, R. Research on mobile robot path planning based on multi-strategy improved ant colony algorithm. *Robotica* **2025**, *43*, 3594–3614. [CrossRef]
12. Xu, X.; Wang, Q.; Ding, X.; Chen, T.; Deng, R. Multi-objective optimization design approach for prefabricated buildings to minimize cost, duration and carbon emissions using ant colony algorithm. *Sci. Rep.* **2025**, *15*, 31683. [CrossRef]

13. Shaji, A.; Sheeja, M.K. Detection and Classification of Aviation Cable Insulation Defects Using Digital Holography and Deep Learning. *J. Nondestruct. Eval.* **2025**, *44*, 133. [CrossRef]
14. Xiaofeng, Y.; Yuzhuo, M. Research on ant colony algorithm based on motion optimization and adaptive threshold for pantograph slider edge detection. *Measurement* **2025**, *245*, 116598. [CrossRef]
15. Zhang, L. The design of path optimization in express transportation based on an improved ant colony algorithm. *Appl. Math. Nonlinear Sci.* **2025**, *10*, 1–15. [CrossRef]
16. Huang, Y.; Su, L. Optimisation method of regional logistics delivery path based on ant colony algorithm. *Int. J. Sustain. Dev.* **2025**, *28*, 322–339. [CrossRef]
17. Tang, J.; Tong, Y.; Chen, L.; Cai, S.; Xiong, S. Integrating neural networks with numerical schemes for dynamical systems: A review. *Neurocomputing* **2025**, *637*, 130122. [CrossRef]
18. Yang, T.; Qian, Z.; Hang, N.; Liu, M. S-PINN: Stabilized physics-informed neural networks for alleviating barriers between multi-level co-optimization. *Comput. Methods Appl. Mech. Eng.* **2025**, *447*, 118348. [CrossRef]
19. Chinesta, F.; Cueto, E.; Champaney, V.; Ghnatios, C.; Ammar, A.; Hascoët, N.; González, D.; Alfaro, I.; Di Lorenzo, D.; Pasquale, A.; et al. *A Gentle Introduction to Data, Learning, and Model Order Reduction*; Springer Nature: Cham, Switzerland, 2025. [CrossRef]
20. Liu, Y.; Wang, L.; Zhou, X.; Li, Z.; Wang, Y. A novel study on hybrid physics-data-driven reduced-order modeling for aerodynamic load inversion under structural field uncertainties. *Comput. Methods Appl. Mech. Eng.* **2026**, *448*, 118504. [CrossRef]
21. Hoorijani, H.; Ouyang, Y.; Heynderickx, G.J.; Van Geem, K.M. Data across the scales: Data-driven multiphase flow reactor modeling. *Curr. Opin. Chem. Eng.* **2025**, *50*, 101193. [CrossRef]
22. Chatterjee, T.; Shaw, A.D.; Friswell, M.I.; Khodaparast, H.H. Sparse Bayesian machine learning for the interpretable identification of nonlinear structural dynamics: Towards the experimental data-driven discovery of a quasi zero stiffness device. *Mech. Syst. Signal Process.* **2023**, *205*, 110858. [CrossRef]
23. Wang, H. Dynamic path optimization of cold chain logistics based on improved ant colony algorithm from a low-carbon perspective. *Sustain. Oper. Comput.* **2025**, *6*, 297–305. [CrossRef]
24. Kamarposhti, M.A.; Shokouhandeh, H.; Lee, Y.; Kang, S.-K.; Colak, I.; Barhoumi, E.M. Optimizing energy management in microgrids with ant colony optimization: Enhancing reliability and cost efficiency for sustainable energy systems. *Int. J. Low-Carbon Technol.* **2024**, *19*, 2848–2856. [CrossRef]
25. Wang, F.; Yuan, G.; Guo, C.; Li, Z. Research on fault diagnosis method of aviation cable based on improved Adaboost. *Adv. Mech. Eng.* **2022**, *14*, 16878132221125762. [CrossRef]
26. Gao, R.; Xiao, Q.; Zhang, W.; Feng, Z. Parameter Solution of Fractional Order PID Controller for Home Ventilator Based on Genetic-Ant Colony Algorithm. *J. Electr. Eng. Technol.* **2024**, *20*, 1153–1171. [CrossRef]
27. Liu, Y.; Weng, X.; Fan, H.; Zhang, J. Improved A* algorithm design for container relocation problem in four-way shuttle vehicle-based digital warehousing systems. *Digit. Eng.* **2026**, *9*, 100086. [CrossRef]
28. Jiaofeng, W. Optimization method for collision avoidance paths of inland ships based on improved ant colony algorithm. *J. Comput. Methods Sci. Eng.* **2024**, *24*, 2831–2845. [CrossRef]
29. Yu, Z.; Zheng, Z.A.; Xie, S.; Huang, Q.; Li, W.; Liu, H. A Hierarchical Adaptive Path Planning Framework (HAPPF) with Synergistic A* and DWA Enhancements for Autonomous Vehicles. *Int. J. Automot. Technol.* 2026; *prepublish*. [CrossRef]
30. Ding, S.; Luo, X.; Gan, C.; Qiu, T.; Ma, Q.; Wang, X.; Zhao, S.; Bao, S.; Qu, G.; Shi, Y. Production and certification constraints to sustainable aviation fuel adoption: Analysis and synergistic solutions. *Energy Strategy Rev.* **2026**, *63*, 102010. [CrossRef]
31. Liu, J.; Qian, Y.; Zhang, W.; Ji, M.; Xv, Q.; Song, H. High-safety path optimization for mobile robots using an improved ant colony algorithm with integrated repulsive field rules. *Robot. Auton. Syst.* **2025**, *190*, 104998. [CrossRef]
32. Wang, H.; Gao, M.; Gao, W.; Li, W. Research on Complex Multiconstraint Path Planning Based on ACA Hybrid Algorithm. *Int. J. Aerosp. Eng.* **2022**, *2022*, 8406307. [CrossRef]
33. Huang, L.; Hu, T.; Wei, J.; Guo, Y.; Tong, X.; Ding, J.; Yang, H.; Zhong, B. Path Planning for Delivery Robots Based on an Improved Ant Colony Optimization Algorithm Combined with Dynamic Window Approach. *Sensors* **2025**, *26*, 72. [CrossRef] [PubMed]
34. Mehdi, M.; Abhijeet, B.; Frank, J. PAveMENT: A framework for the intelligent and safe navigation of unmanned aerial vehicles over optimal PAths in MobilE NeTworks. *Comput. Netw.* **2020**, *186*, 107785. [CrossRef]