Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 11(4): 11882–11923.
DOI: 10.3934/math.2026488
Received: 01 March 2026
Revised: 14 April 2026
Accepted: 16 April 2026
Published: 28 April 2026

Research article

A multi-objective grey wolf optimizer for grid-based mobile robot path planning with turn-aware optimization

Tawfik Guesmi[1], **Nandhini Mahadevan**[2], **Dinesh Karunanidy**[3,*] **and Khalid Alqunun**[1]

[1] Department of Electrical Engineering, College of Engineering, University of Ha'il, Ha'il 2240, Saudi Arabia; tawfik.guesmi@istmt.rnu.tn (T.G.); kh.alqunun@uoh.edu.sa (K.A.)

[2] Department of Computer Science and Engineering (Data Science), Madanapalle Institute of Technology & Science (MITS), Deemed to be University, Madanapalle 517325, India; nandynew24@gmail.com (N.M.)

[3] Department of Computer Science & Technology, Madanapalle Institute of Technology & Science (MITS), Deemed to be University, Madanapalle 517325, India; dineshhumar@gmail.com (D.K.)

* **Correspondence:** Email: dineshhumar@gmail.com.

Abstract: Grid-based mobile robot path planning often requires a trade-off between path efficiency and maneuver regularity when obstacle force detours and trigger heading changes. This paper proposes a multi-objective grey wolf optimizer for grid-based path planning with turn-aware optimization. The path length and maneuver burden are optimized simultaneously using an 8-connected motion model with Euclidean-consistent costs and a strict no-corner-cutting feasibility rule. The experimental analysis is organized in three levels: A basic benchmark specification with 9-map environments, a structured 60-map benchmark covering five grid sizes, four obstacle-density levels, and three obstacle morphologies, and sequential qualitative visualization of dynamic obstacle avoidance and multi-robot coordination. The proposed method consistently achieves the lowest mean path length and the lowest mean turn count among the compared methods. Statistical comparison further shows significant gains in path length and turning burden on the majority of benchmark environments. The results indicate that the proposed method preserves favorable path-quality trade-offs across increasing grid size, obstacle density, and structural complexity, while also exhibiting coherent behavior in dynamic avoidance and coordination scenarios.

Keywords: mobile robot path planning; occupancy grid; grey wolf optimizer; multi-objective optimization; turn minimization
Mathematics Subject Classification: 90C59, 90C29, 90C35, 68T40

1. Introduction

Mobile robot path planning (MRPP) concerns the computation of a collision-free route from a start cell to a target cell under environmental and execution constraints. In many practical systems, planning is carried out on a discrete map, such as an occupancy grid, produced by a localization and mapping pipeline so that navigation decisions can be made quickly and robustly in uncertain environments [1, 2]. Recent advances in perception, simultaneous localization and mapping (SLAM), and sensor fusion lead to better maps, however, failure to plan still occurs often; this is because path quality is usually that of a single geometric criterion (e.g., traversal distance) while execution quality is taken implicitly.

This limitation is particularly important in grid-based navigation where a geometrically short path may not necessarily be an easy path. A path with frequent heading changes can increase control effort, induce oscillatory behavior, prolong acceleration–braking cycles, and degrade tracking robustness in cluttered environments. These considerations motivate turn-aware planning criteria that account explicitly for maneuver smoothness and reorientation burden instead of treating them as an afterthought [3, 4]. To solve this problem, previous studies have explored graph search, local motion planning, sampling-based methods, and nature-inspired optimization [5, 6]. Metaheuristic algorithms are especially attractive for MRPP because they can handle non-differentiable objectives, nonconvex search spaces, and competing criteria without gradient information [7]. Grey wolf optimization (GWO) has attracted sustained attention because of its simple leadership-driven search mechanism and competitive exploration–exploitation behavior [8]. Concurrently, multi-objective optimization offers a more robust way to handle conflicting navigation objectives by preserving trade-offs among objectives instead of collapsing them into a fixed weighted sum [9, 10].

This paper addresses a specific gap in grid-based MRPP: Embedding maneuver awareness directly into the planning objective while preserving strict execution-level feasibility on an 8-connected occupancy grid. The contribution is not merely the use of GWO for path planning. Rather, the proposed framework combines (i) an action-sequence encoding consistent with grid execution, (ii) an angle-aware maneuver-complexity objective, (iii) archive-guided leader selection in a bi-objective setting, and (iv) deterministic feasibility repair with bounded suffix completion under a strict no-corner-cutting rule. The main contributions are threefold. First, a bi-objective, heading-aware planning formulation is introduced for the joint minimization of path length and maneuver complexity on occupancy grids. Second, a pareto-archived grey wolf optimizer is developed to solve this formulation in the same discrete action space used for decoding and evaluation. Third, the proposed method is assessed using representative-path statistics, uncertainty analysis, and benchmark-driven effectiveness analysis with respect to grid size, obstacle density, and obstacle morphology.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 4 presents the proposed formulation and solver. Section 5 describes the experimental setup, and Section 5.1.2 discusses the results. Section 6 concludes the paper.

2. Related work

MRPP has been studied through deterministic graph search, sampling-based planning, learning-assisted navigation, and population-based optimization. Because the present work is formulated on occupancy grids and explicitly optimizes maneuver complexity together with path length, this section focuses on four strands of prior research: (i) Graph search and grid cost design, (ii) local motion-feasibility strategies, (iii) metaheuristic and multi-objective MRPP methods, and (iv) GWO-based planning variants. The review emphasizes not only the underlying approaches but also the concrete limitations that motivate the proposed method.

2.1. *Graph search on grids and cost design*

Graph-based planners transform MRPP into shortest-path computation on a graph. Dijkstra's algorithm provides the optimal path with respect to nonnegative edge costs, while heuristic A* speeds up the search. These methods remain a standard baseline because they are deterministic, complete on finite graphs, and straightforward to validate [11, 12]. Nevertheless, the optimality guarantee is only as informative as the cost function. Due to their reliance on step length, occupancy-grid cost functions can generate staircase patterns and unnecessary heading changes; therefore, practitioners often add heuristic penalties or apply post-processing for smoothness and clearance [7, 13]. Allowing diagonal movement presents complications for feasibility in the presence of obstacles, motivating constraints like no corner-cutting when a grid is used as a proxy for continuous movement.

2.2. *Sampling-based and local motion strategies*

Probabilistic roadmap (PRM) and exploring rapidly random tree (RRT) are sampling-based planners that can handle higher-dimensional configuration spaces and can incorporate kinodynamic constraints. However, in the case of grid maps, they are often used to generate coarse routes that require further smoothing or tracking control [14]. Local planners such as the dynamic window approach (DWA) consider short-horizon motion feasibility and obstacle avoidance directly, and are often used in conjunction with a global grid planner for robustness [6, 15]. In some cases, learning-enhanced navigation has been used to predict feasible steering commands or to refine cost maps, although performance may be sensitive to training data and domain shift. Several approaches emphasise the need for maneuver-aware behavior. However, turning is usually handled implicitly (e.g., via control cost) rather than by explicitly optimizing a discrete turn-count objective at the planning level.

2.3. *Metaheuristics and multi-objective MRPP*

Population-based metaheuristics play an important role in mobile robot path planning because they can handle nonconvex search spaces, discontinuous feasibility constraints, and multiple competing criteria without requiring gradient information [5]. Methods based on genetic algorithms, particle swarm optimization, ant colony optimization, and hybrid variants have therefore been widely adopted in MRPP. Their effectiveness depends strongly on how candidate paths are encoded, repaired, and evaluated under feasibility constraints.

Weighted-sum aggregation remains common in MRPP because it is simple to implement, but it requires the user to predefine trade-off coefficients and may obscure non-convex trade-off structure. By contrast, genuinely Pareto-based methods retain multiple non-dominated solutions during the search. For example, the multi-objective artificial bee colony approach in [16] and the multi-target probabilistic-roadmap framework in [9] both illustrate the value of preserving alternative trade-off solutions instead of collapsing all navigation criteria into a single scalar objective. The remaining gap is therefore not that Pareto-based navigation is absent, but that explicit turn-aware, execution-consistent, grid-based formulations with strict feasibility handling are still comparatively limited.

Turn-aware planning deserves separate attention because heading changes directly affect smoothness, energy demand, and execution effort. Existing studies on turn minimization are closely related in motivation to the present work, but they typically emphasize coverage-style traversal, route reshaping, or post-processing rather than point-to-point occupancy-grid planning under explicit Pareto trade-offs between traversal length and maneuver complexity [3, 4, 17]. These observations motivate the present formulation, which combines explicit bi-objective optimization with deterministic decoding, repair, and no-corner-cutting feasibility enforcement.

3. Bi-objective grid path planning formulation

Table 1 summarizes the notation used throughout the formulation and solver.

Table 1. Notation used in the planning formulation and the archive-guided multi-objective grey wolf optimizer (MOGWO) solver.

Symbol	Meaning
Ω	Occupancy-grid domain
$\mathcal{F}$	Free-cell set
$\mathcal{O}$	Obstacle set
$\mathcal{U}$	8-connected action set
S, G	Start and goal cells
$\pi = \langle p_1, \ldots, p_L \rangle$	Feasible path from S to G
p_ℓ	Cell visited at path position ℓ
L	Number of cells in the decoded path
$\mathbf{z}_i$	Auxiliary continuous vector for wolf i
$\mathbf{X}_i$	Discrete action sequence of wolf i
u_ℓ	Executed action at step ℓ
δ_ℓ	Angle-aware turn-severity penalty at step ℓ
$f_1(\pi)$	Path-length objective
$f_2(\pi)$	Maneuver-complexity objective
$\mathcal{A}$	External non-dominated archive
$A_{\max}$	Archive-capacity limit
W	Population size
T	Maximum number of iterations
K	Action horizon
R	Repair budget
η	Leader-sampling smoothing constant
$h(\cdot)$	Euclidean goal heuristic
$\Pi(\cdot)$	Continuous-to-discrete projection operator
$\pi^\dagger$	Representative archive solution used for reporting

3.1. Occupancy grid model and motion constraints

Let the environment be a two-dimensional occupancy grid $\Omega = \{1, \ldots, N_x\} \times \{1, \ldots, N_y\}$. Each cell $p = (x, y) \in \Omega$ is either free or occupied. The free-cell set is denoted by $\mathcal{F} \subseteq \Omega$, the obstacle set by $\mathcal{O} = \Omega \setminus \mathcal{F}$, and the start and goal cells by $S \in \mathcal{F}$ and $G \in \mathcal{F}$, respectively.

Robot motion follows an 8-connected lattice with action set

$$\mathcal{U} = \{u^{(1)}, \ldots, u^{(8)}\} = \{(1,0), (1,1), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1)\}. \tag{3.1}$$

The actions are ordered by heading angle, which makes the continuous-to-discrete projection in the optimizer well-defined. For $u = (\Delta x, \Delta y) \in \mathcal{U}$, the successor of cell $p = (x, y)$ is

$$\mathrm{succ}(p, u) = p + u = (x + \Delta x,\ y + \Delta y). \tag{3.2}$$

The Euclidean-consistent step cost is

$$c(u) = \|u\|_2 = \begin{cases} 1, & |\Delta x| + |\Delta y| = 1, \\ \sqrt{2}, & |\Delta x| = |\Delta y| = 1. \end{cases} \tag{3.3}$$

Accordingly, orthogonal and diagonal moves increase cost by their geometric length rather than by a constant transition penalty.

A diagonal transition is feasible only if the no-corner-cutting rule is satisfied. For any diagonal action $u = (\Delta x, \Delta y)$ with $|\Delta x| = |\Delta y| = 1$, the move from $p = (x, y)$ to $p + u$ is allowed only when

$$(x + \Delta x, y) \in \mathcal{F} \qquad \text{and} \qquad (x, y + \Delta y) \in \mathcal{F}. \tag{3.4}$$

This constraint prevents diagonal squeezing through blocked corners and keeps feasibility consistent with standard grid-search baselines.

Table 1 summarizes the principal symbols used in both the bi-objective formulation and the archive-guided solver.

3.2. Path definition and feasibility

A feasible path is a finite cell sequence from the start cell to the goal cell,

$$\pi = \langle p_1, p_2, \ldots, p_L \rangle, \qquad p_1 = S,\ p_L = G,\ p_\ell \in \mathcal{F}\ \forall \ell, \tag{3.5}$$

where each transition is generated by an admissible action in the 8-connected action set:

$$p_{\ell+1} = \mathrm{succ}(p_\ell, u_\ell), \qquad u_\ell \in \mathcal{U}, \quad \ell = 1, 2, \ldots, L-1. \tag{3.6}$$

In addition, every diagonal transition must satisfy the no-corner-cutting condition in Eq (3.4). Under this definition, feasibility is built directly into the path representation; therefore, no separate redundant boundedness condition is needed once $\mathcal{F} \subseteq \Omega$ has been defined.

3.3. Bi-objective formulation and objective definitions

The planning problem is formulated as the bi-objective optimization problem

$$\min \mathbf{F}(\pi) = [f_1(\pi),\ f_2(\pi)] \tag{3.7}$$

subject to the feasibility constraints.

3.3.1. Objective 1: Path length

The first objective is the total traversal cost,

$$f_1(\pi) = \sum_{\ell=1}^{L-1} c(p_{\ell+1} - p_\ell), \tag{3.8}$$

where $c(\cdot)$ is given by Eq (3.3).

3.3.2. Objective 2: Maneuver complexity

Let the executed action at step ℓ be

$$u_\ell = p_{\ell+1} - p_\ell, \qquad \ell = 1, 2, \ldots, L-1. \tag{3.9}$$

The corresponding heading angle is

$$\theta(u_\ell) = \operatorname{atan2}(\Delta y_\ell, \Delta x_\ell) \in \{0^\circ, 45^\circ, \ldots, 315^\circ\}. \tag{3.10}$$

The turn severity between two consecutive moves is defined as

$$\delta_\ell = \frac{1}{45^\circ} \min\left(\left|\theta(u_\ell) - \theta(u_{\ell-1})\right|, 360^\circ - \left|\theta(u_\ell) - \theta(u_{\ell-1})\right|\right), \qquad \ell = 2, \ldots, L-1, \tag{3.11}$$

so that $\delta_\ell \in \{0, 1, 2, 3, 4\}$ and a 45° adjustment is penalized less than a 180° reversal. The maneuver-complexity objective is then

$$f_2(\pi) = \sum_{\ell=2}^{L-1} \delta_\ell. \tag{3.12}$$

For descriptive reporting, the plain number of turns may also be computed as

$$N_{\text{turn}}(\pi) = \sum_{\ell=2}^{L-1} \mathbb{I}(\delta_\ell > 0), \tag{3.13}$$

but the optimization itself uses Eq (3.12) because it distinguishes mild and severe heading changes.

3.4. Pareto dominance and external archive

For two feasible paths π^a and π^b, path π^a dominates π^b if

$$f_m(\pi^a) \le f_m(\pi^b) \quad \forall m \in \{1, 2\}, \qquad \text{and} \qquad f_m(\pi^a) < f_m(\pi^b) \quad \text{for at least one } m. \tag{3.14}$$

An external archive $\mathcal{A}$ stores the non-dominated feasible solutions discovered during the search.

When archive truncation is required, spread in objective space is preserved by the crowding distance. For each objective $m \in \{1, 2\}$, the archive is sorted increasingly by f_m. Boundary solutions are assigned infinite crowding distance. For an interior solution π_i, the normalized contribution of objective m is

$$\mathrm{CD}_m(\pi_i) = \frac{f_m(\pi_{i+1}) - f_m(\pi_{i-1})}{f_m^{\max} - f_m^{\min} + \varepsilon}, \tag{3.15}$$

where $\varepsilon > 0$ avoids division by zero. The total crowding distance is

$$\mathrm{CD}(\pi_i) = \sum_{m=1}^{2} \mathrm{CD}_m(\pi_i). \tag{3.16}$$

If $|\mathcal{A}| > A_{\max}$, the archive is truncated by repeatedly removing the solution with the smallest crowding distance. The archive cap is introduced to keep memory usage, archive-update time, and leader-sampling cost bounded across repeated runs.

The archive cap is fixed at $A_{\max} = 40$. This parameter is introduced only to bound memory use and leader-sampling cost; it is not part of the dominance definition itself. When $|\mathcal{A}| \leq A_{\max}$, no truncation is performed. To show that this cap is a safeguard rather than a performance driver, the realized archive cardinality distribution is reported in the Pareto-quality analysis.

When a single path must be reported for visualization or scalar summary tables, it is selected from the archive by a fixed preference rule. Let

$$z^{\star} = \left(\min_{\pi\in\mathcal{A}} f_1(\pi),\ \min_{\pi\in\mathcal{A}} f_2(\pi)\right) \tag{3.17}$$

be the ideal point. The representative solution is chosen as the normalized knee-point approximation

$$\pi^{\dagger} = \arg\min_{\pi\in\mathcal{A}} \sqrt{\hat{f}_1(\pi)^2 + \hat{f}_2(\pi)^2}, \tag{3.18}$$

where

$$\hat{f}_m(\pi) = \frac{f_m(\pi) - \min_{\rho\in\mathcal{A}} f_m(\rho)}{\max_{\rho\in\mathcal{A}} f_m(\rho) - \min_{\rho\in\mathcal{A}} f_m(\rho) + \varepsilon}. \tag{3.19}$$

This rule makes representative-path reporting reproducible and avoids ad hoc path selection from the archive.

4. Proposed grid-path encoding MOGWO for mobile robot path planning

4.1. Wolf encoding and deterministic decoding

Each grey wolf is represented by an auxiliary real-valued vector

$$\mathbf{z}_i = [z_{i,1}, z_{i,2}, \ldots, z_{i,K}] \in [1, 8]^K, \tag{4.1}$$

where K is the maximum action horizon and i indexes the wolf. The corresponding discrete action sequence is obtained through the projection

$$\Pi(z_{i,\ell}) = u^{\left(\mathrm{clip}(\mathrm{round}(z_{i,\ell}),1,8)\right)}, \qquad \ell = 1, 2, \ldots, K, \tag{4.2}$$

where $\mathrm{clip}(a, 1, 8) = \min\{8, \max\{1, a\}\}$. The discrete encoding is therefore

$$\mathbf{X}_i = [u_{i,1}, u_{i,2}, \ldots, u_{i,K}], \qquad u_{i,\ell} = \Pi(z_{i,\ell}) \in \mathcal{U}. \tag{4.3}$$

As a simple example, $z_{i,\ell} = 2.6$ is rounded to index 3, and the projected action is $u^{(3)} = (0, 1)$.

4.1.1. Decoding from actions to a grid path

Decoding starts from $p_1 = S$ and simulates the action sequence forward:

$$\tilde{p}_{\ell+1} = p_\ell + u_{i,\ell}, \qquad \ell = 1, 2, \ldots, K. \tag{4.4}$$

If $\tilde{p}_{\ell+1}$ is feasible, then $p_{\ell+1} = \tilde{p}_{\ell+1}$. Otherwise, the repair operator in Section 4.4 is invoked and decoding resumes from the corrected state. Decoding terminates immediately once the goal is reached. If the horizon is exhausted before the goal is reached, an exact suffix planner is called once from the last valid state to the goal. If no feasible suffix exists, the candidate is marked infeasible.

4.2. Leader selection under multi-objective search

In the multi-objective setting, the leadership hierarchy is sampled from the archive rather than from a single ranked list. To favor sparsely populated regions of the current front, the probability of selecting path $\pi \in \mathcal{A}$ as a leader is

$$p(\pi) = \frac{\mathrm{CD}(\pi) + \eta}{\sum_{\rho \in \mathcal{A}} (\mathrm{CD}(\rho) + \eta)}, \tag{4.5}$$

where $\eta > 0$ is a small smoothing constant. The three leaders (α, β, δ) are sampled without replacement at each iteration.

4.3. GWO update in auxiliary space and grid decoding

The standard GWO update is applied in the auxiliary continuous space of $\mathbf{z}$ rather than directly on discrete actions. For a wolf position $\mathbf{z}$ and leader positions $\mathbf{z}_\alpha$, $\mathbf{z}_\beta$, and $\mathbf{z}_\delta$, the leader-relative vectors are

$$\begin{aligned} \mathbf{D}_\alpha &= \left|\mathbf{C}_1 \odot \mathbf{z}_\alpha - \mathbf{z}\right|, & \mathbf{z}_1 &= \mathbf{z}_\alpha - \mathbf{A}_1 \odot \mathbf{D}_\alpha, \\ \mathbf{D}_\beta &= \left|\mathbf{C}_2 \odot \mathbf{z}_\beta - \mathbf{z}\right|, & \mathbf{z}_2 &= \mathbf{z}_\beta - \mathbf{A}_2 \odot \mathbf{D}_\beta, \\ \mathbf{D}_\delta &= \left|\mathbf{C}_3 \odot \mathbf{z}_\delta - \mathbf{z}\right|, & \mathbf{z}_3 &= \mathbf{z}_\delta - \mathbf{A}_3 \odot \mathbf{D}_\delta. \end{aligned} \tag{4.6}$$

The updated continuous position is

$$\mathbf{z}(t+1) = \frac{\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3}{3}. \tag{4.7}$$

The coefficient vectors follow the standard GWO rule

$$\mathbf{A} = 2a\mathbf{r}_1 - a, \qquad \mathbf{C} = 2\mathbf{r}_2, \tag{4.8}$$

where $\mathbf{r}_1, \mathbf{r}_2 \sim U[0,1]^K$, and the convergence parameter decreases linearly as

$$a(t) = 2\left(1 - \frac{t}{T}\right), \qquad t = 0, 1, \ldots, T. \tag{4.9}$$

After the continuous update, each component is projected to the nearest admissible action using Eq (4.2). The resulting action sequence is then decoded according to Section 4.1.1.

4.4. Deterministic feasibility repair and exact suffix completion

The repair operator is invoked when a tentative move violates grid bounds, occupancy, or the no-corner-cutting rule. Its role is to recover a feasible continuation toward the goal while keeping the decoded path executable. Let p_ℓ be the current valid cell, let $u_{\ell-1}$ denote the previous executed action when $\ell > 1$, and let $\mathcal{N}_f(p_\ell)$ be the set of all feasible one-step successors from p_ℓ under Eqs (3.2)–(3.4). Each candidate neighbor $q \in \mathcal{N}_f(p_\ell)$ is ranked lexicographically by

$$\kappa(q) = \Big(h(q), \Delta_\theta(u_{\ell-1}, q - p_\ell), c(q - p_\ell)\Big), \tag{4.10}$$

where $h(q) = \|q - G\|_2$ is the Euclidean goal heuristic and $\Delta_\theta(\cdot, \cdot)$ denotes the turn-severity term induced by Eq (3.11). The best-ranked feasible neighbor is selected deterministically.

Let $q^\star$ denote the selected repair state. The repaired prefix then becomes

$$\langle p_1, \ldots, p_\ell, q^\star \rangle.$$

In other words, the infeasible tentative state $\tilde{p}_{\ell+1}$ is discarded and replaced by $q^\star$. Decoding then resumes from $q^\star$ using the remaining undecoded action suffix. Thus, the repair operator modifies exactly one transition at a time. If repeated local repair fails within the budget R, the remaining suffix is replaced by an orientation-aware A* completion from $q^\star$ to G; otherwise, the candidate is rejected and reinitialized.

If no feasible one-step neighbor exists, or if the number of repair invocations for the current wolf exceeds the repair budget R, a suffix planner is called on an orientation-augmented state graph. The state is (p, ϑ), where $p \in \mathcal{F}$ and $\vartheta \in \{0°, 45°, \ldots, 315°\}$ is the incoming heading. A directed edge exists from (p, ϑ) to $(p + u, \theta(u))$ whenever the underlying move is feasible. The suffix planner uses A* with the admissible heuristic

$$h_{\text{suf}}(p, \vartheta) = \|p - G\|_2 \tag{4.11}$$

and the lexicographic edge cost

$$(c(u),\ \Delta_\theta(\vartheta, \theta(u))). \tag{4.12}$$

If a suffix path is found, it is appended and decoding terminates successfully. Otherwise, the wolf is marked infeasible and reinitialized. Algorithm 1 clearly explains the feasibility repair and exact suffix completion.

Here, W denotes the number of wolves, T the maximum number of iterations, K the action horizon, $A_{\max}$ the archive capacity, R the maximum number of repair-triggered suffix calls allowed per wolf evaluation, and η the leader-sampling smoothing constant in Eq (4.5). These symbols are stated explicitly because they directly determine the search budget and archive behavior, and should be instantiated numerically in the experimental section for full reproducibility. Grey wolf optimization (GWO) analyses a hierarchy of dominance in which a small elite sub-groups the remaining population. GWO is popular choice due to its simple update rules, low conceptual overhead, and balanced exploration and exploitation behavior [18]. Recent variants in [19] introduced adaptive coefficients, hybrid mutation schemes, and problem-specific encoding to improve solution quality. In the proposed method, each wolf encodes a finite sequence of 8-connected motion actions that is decoded into an executable grid path under Euclidean-consistent movement costs and a strict no-corner-cutting constraint. The resulting optimizer minimizes both traversal length and maneuver complexity in a discrete action space. This method uses Pareto dominance, an external archive of non-dominated solutions, and archive-guided leader selection during search to maintain a trade-off between conflicting objectives. This design avoids the need to pre-specify weighted-sum coefficients and differs from post-smoothing approaches that adjust path geometry only after path generation. The search process controls heading changes rather than the decoding process due to treating maneuver complexity as an inherent optimization objective. The representation entails deterministic feasibility repair and bounded suffix completion, obtaining executable and robust paths. Algorithm 2 summarizes the complete archive-guided optimization workflow.

Algorithm 1 One-step repair with bounded suffix completion

```
Input: Current valid cell $p_\ell$, previous heading $u_{\ell-1}$, remaining action suffix $(u_{\ell+1}, u_{\ell+2}, \ldots)$, goal $G$,
    repair count $r$, repair budget $R$
Output: Repaired continuation or failure flag
Build feasible neighbor set $\mathcal{N}_f(p_\ell)$
if $\mathcal{N}_f(p_\ell) \neq \emptyset$ then
    Select $q^\star = \arg\min_{q \in \mathcal{N}_f(p_\ell)} \kappa(q)$ using Eq (4.10)
    Replace the infeasible tentative step by $q^\star$
    Resume decoding from $q^\star$ with the remaining action suffix
    return success
else if $r < R$ then
    Run orientation-aware suffix A$^*$ from $(p_\ell, \theta(u_{\ell-1}))$ to $G$
    if a feasible suffix is found then
        Append the suffix and terminate decoding
        return success
    else
        return failure
    end if
else
    return failure
end if
```

Algorithm 2 Turn-aware Pareto MOGWO for grid-based path planning

```
Input: Grid domain $\Omega$, free-cell set $\mathcal{F}$, start $S$, goal $G$, population size $W$, iterations $T$, horizon $K$,
    archive capacity $A_{\max}$, repair budget $R$
Output: Final archive $\mathcal{A}$ and representative path $\pi^\dagger$
Initialize $W$ wolves in auxiliary space: $\mathbf{z}_i \sim U([1,8]^K)$
Project each $\mathbf{z}_i$ to a discrete action sequence using Eq (4.2)
Decode each sequence; invoke Algorithm 1 when needed
Evaluate feasible paths using Eqs (3.8) and (3.12)
Initialize archive $\mathcal{A}$ with the non-dominated feasible solutions
for $t = 1$ to $T$ do
    Update $a(t)$ using Eq (4.9)
    Compute crowding distance in $\mathcal{A}$ using Eqs (3.15) and (3.16)
    Sample leaders $(\alpha, \beta, \delta)$ from $\mathcal{A}$ using Eq (4.5)
    for $i = 1$ to $W$ do
        Update $\mathbf{z}_i$ using Eqs (4.6) and (4.7)
        Project $\mathbf{z}_i$ to a discrete action sequence
        Decode and repair the sequence to obtain a feasible path if possible
        if a feasible path is obtained then
            Evaluate $(f_1, f_2)$ and insert the path into $\mathcal{A}$ if it is non-dominated
        else
            Reinitialize $\mathbf{z}_i$ uniformly in $[1,8]^K$
        end if
    end for
    if $|\mathcal{A}| > A_{\max}$ then
        Truncate $\mathcal{A}$ by removing the smallest-crowding-distance solutions
    end if
end for
Select the representative solution $\pi^\dagger$ using Eq (3.18)
return $\mathcal{A}$ and $\pi^\dagger$
```

4.5. Worst-case computational complexity

Let $|\mathcal{F}|$ denote the number of free cells. Decoding a single wolf requires at most K forward-simulation steps. Each local detour check inspects at most eight neighbors and is therefore $O(1)$. The more expensive operation is the suffix planner. On the orientation-augmented graph, the number of states is at most $8|\mathcal{F}|$, and the number of directed transitions is $O(|\mathcal{F}|)$. With a binary-heap implementation, a single suffix A* call has worst-case complexity $O(|\mathcal{F}| \log |\mathcal{F}|)$. If at most R suffix invocations are allowed per wolf evaluation, the worst-case cost of evaluating one wolf in one iteration is

$$O(K + R|\mathcal{F}| \log |\mathcal{F}|). \tag{4.13}$$

For a population of W wolves and T iterations, the overall worst-case complexity of the search phase is

$$O\Big(TW(K + R|\mathcal{F}| \log |\mathcal{F}|) + T\,\Gamma(A_{\max}, W)\Big), \tag{4.14}$$

where $\Gamma(A_{\max}, W)$ denotes the archive-maintenance cost. With bounded archive size and two objectives, this term is typically dominated by the per-wolf decoding and suffix-planning cost.

5. Experimental setup

The experimental study is organized into three complementary levels. The first level presents the basic level benchmark uses 9 different environment with obstacle count and densities in the grid environment. The second level introduces a structured 60-map benchmark that enables systematic analysis with respect to grid size, obstacle density, and obstacle morphology. The third level provides sequential qualitative visualization of dynamic obstacle avoidance and multi-robot coordination. This three-level organization supports both controlled quantitative comparison and application-oriented qualitative interpretation.

Table 2. Algorithmic settings used in the comparative study.

Algorithm	Parameter settings and implementation details
Proposed MOGWO	Population size $W = 30$, maximum iterations $T = 12$, horizon $K = 40$, archive capacity $A_{\max} = 40$, repair budget $R = 4$, leader-sampling smoothing constant $\eta = 0.10$. The selected values were obtained by pilot tuning around the reference settings and then fixed across all environments.
IGA	Population size = 30, crossover rate = 0.85, mutation rate = 0.10, elitism = 2, stopping criterion = maximum 25 generations or early termination after 5 stagnant generations, improvement mechanism = elitist retention with adaptive mutation and feasibility-preserving chromosome repair, implementation reference [20].
INN–DWA	Look-ahead horizon = 3.0 s, velocity window parameters $v \in [0, 0.8]$ m/s, $\omega \in [-1.2, 1.2]$ rad/s, $\Delta t = 0.1$ s, network or gain settings $\alpha = 0.45$, $\beta = 0.35$, $\gamma = 0.20$, stopping criterion = goal reached or 27 local-planning cycles, improvement mechanism = global neural guidance with reward-weighted DWA refinement and distance-based priority handling, implementation reference [21].
IPSO	Swarm size = 30, inertia weight = 0.72, acceleration constants $c_1 = 1.50$ and $c_2 = 1.50$, stopping criterion = maximum 30 iterations or no improvement for 5 successive iterations, improvement mechanism = velocity clamping with local-best refinement and feasibility-aware path repair, implementation reference [22].
IA*	Heuristic specification = Euclidean-distance heuristic with turn-aware cost augmentation $\lambda = 0.20$, tie-breaking rule = prefer lower heuristic value and fewer heading changes, neighborhood rule = 8-connected expansion with no-corner-cutting validation, improvement mechanism = local pruning and path simplification after graph search, implementation reference [23].

The proposed-method parameters $(W, T, K, A_{\max}, R, \eta)$ were fixed after pilot tuning and then kept unchanged for all benchmark environments, whereas the baseline settings were adopted from their respective improved variants and normalized to the common occupancy-grid evaluation framework used in this study. Table 2 lists the numerical settings used in all experiments.

Across all environments, the proposed MOGWO is compared against four improved baselines frequently used in grid-based navigation: The improved genetic algorithm (IGA), the improved neural network–dynamic window approach (INN–DWA), the improved particle swarm optimization (IPSO), and the improved A-star algorithm (IA*). All methods are evaluated on identical occupancy-grid benchmarks under the same feasibility and motion rules. For relative comparison, we report percentage relative deviation (PRD),

$$\mathrm{PRD}(\%) = \frac{x_{\text{baseline}} - x_{\text{proposed}}}{x_{\text{baseline}}} \times 100, \tag{5.1}$$

so that a positive PRD indicates a lower cost for the proposed method.

5.1. Level 1: Basic-level benchmark specification

The core benchmark suite consists of eight 21×21 occupancy-grid instances with start cell $(1, 1)$ and goal cell $(19, 19)$, selected for controlled comparison under increasing obstacle density. These benchmark instances should not be interpreted as an intrinsic restriction of the proposed formulation. To partially assess scalability beyond the core suite, an additional 50×50 environment is also included later in the Results section. All environments use the same 8-connected motion model, Euclidean-consistent step costs, and strict no-corner-cutting feasibility rule.

Table 3 reports the obstacle counts, densities, and qualitative turning difficulty assigned at the benchmark-design time. Environment layouts are shown in Figures 1–3. In each environment, each stochastic algorithm is executed for 20 independent runs. For every metric, we report the mean (Avg), variance (Var), standard error (SE), interquartile range (IQR), and coefficient of variation (CV). Runtime and best-iteration statistics are also reported to contextualize solution quality relative to computational effort.

Table 3. Basic-level testbed specification.

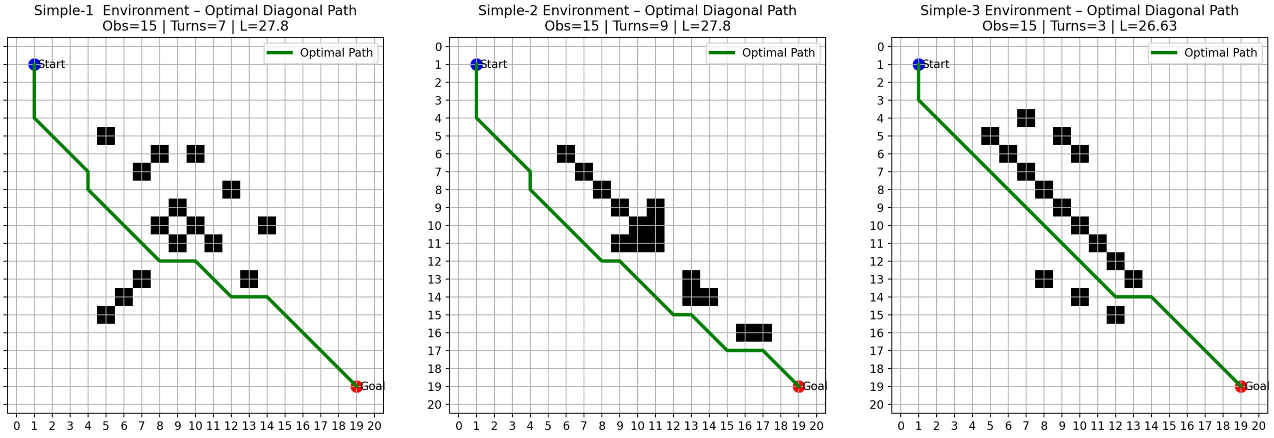
Env.	**Cat.**	**Obs.**	**Density (%)**	**Turns**	**Level**	**Validation**
Simple-1	Simple	15	3.40	≤4	Low	Dijkstra
Simple-2	Simple	15	3.40	4–5	Low	Dijkstra
Simple-3	Simple	15	3.40	5–6	Low–Mod.	Dijkstra
Moderate-1	Moderate	25	5.67	≥6	Moderate	Dijkstra
Moderate-2	Moderate	25	5.67	≥6	Moderate	Dijkstra
Moderate-3	Moderate	25	5.67	≥6	Moderate	Dijkstra
Complex-1	Complex	35	7.94	≥10	High	Dijkstra
Complex-2	Complex	35	7.94	≥10	High	Dijkstra

In Table 3, “Turns” denotes the approximate turning burden expected under a near-diagonal traversal of the environment, while “Level” denotes the qualitative difficulty category assigned during benchmark design. These columns are descriptive annotations for organizing the testbed and are not optimization objectives.

5.1.1. Benchmark scope and additional diversity

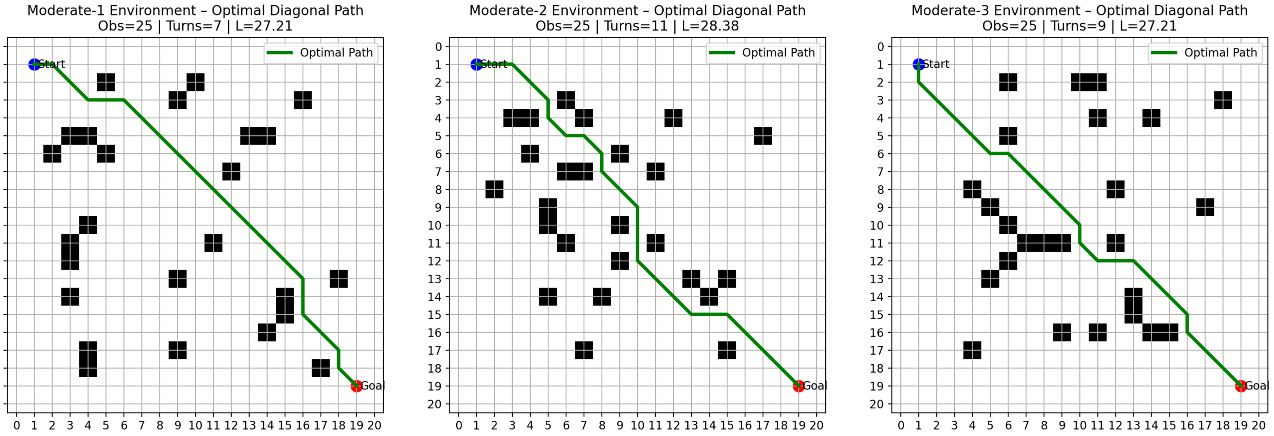
The benchmark experimentation is intentionally structured so that all algorithms are compared under identical motion and feasibility assumptions. The use of fixed start and goal cells in the core 21 × 21 suite limits positional diversity. Then benchmark with a larger 50 × 50 environment, which explicitly interprets the core suite as a controlled comparison set rather than as a statement that the formulation is restricted to one map size or one start–goal arrangement.

This design enables two quantitative analyses that are not possible in the handcrafted suite alone: (i) Effectiveness as a function of grid size, and (ii) effectiveness as a function of obstacle density. The resulting curves allow us to examine whether the relative advantage of the proposed method increases, decreases, or remains stable as the navigation problem becomes larger or more cluttered.



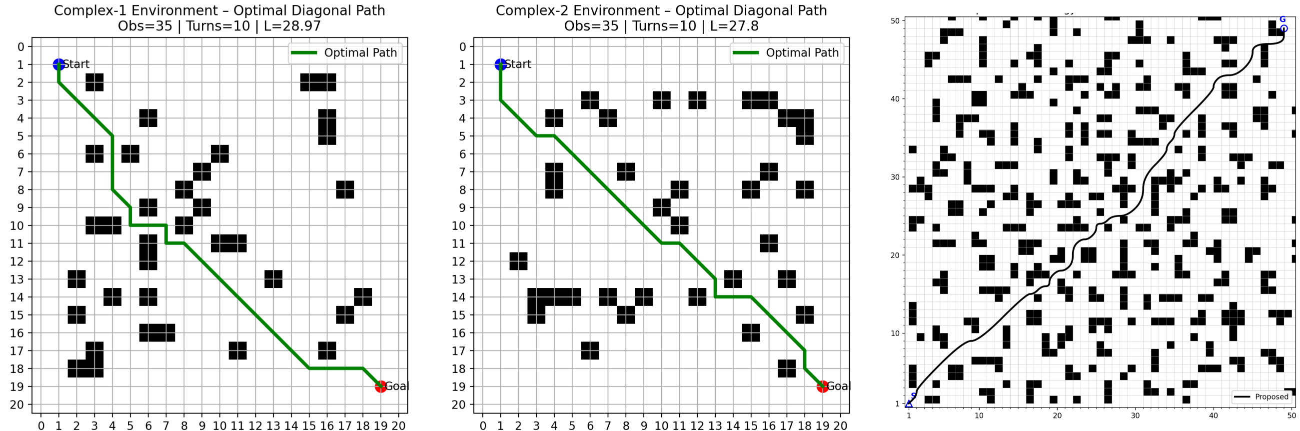
(a) Simple-1 (b) Simple-2 (c) Simple-3

Figure 1. Benchmark layouts for Simple environments.



(a) Moderate-1 (b) Moderate-2 (c) Moderate-3

Figure 2. Benchmark layouts for Moderate environments.



(a) Complex-1 (b) Complex-2 (c) 50 × 50 Testbed

Figure 3. Benchmark layouts for complex environments and the 50 × 50 testbed.

5.1.2. Experimentation results and performance analysis

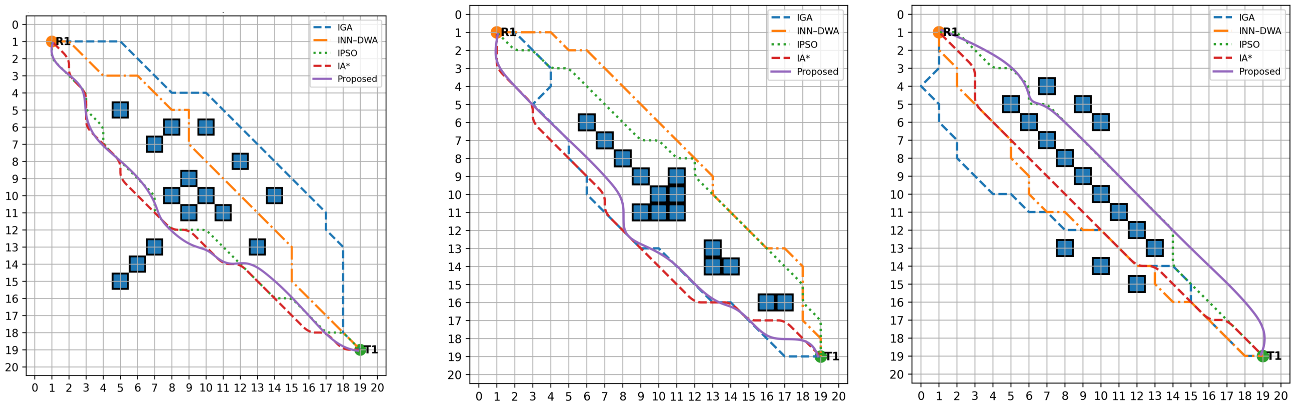
Tables 4–12 summarize the representative-path statistics selected from each method under a fixed post hoc reporting rule. These tables report path length (PL) and plain turn count for interpretability, whereas the optimized second objective is the angle-aware maneuver-complexity measure defined in Eq (3.12). The archive-level Pareto quality of the compared methods is evaluated separately later in Section 5.1.8 using hypervolume (HV), inverted generational distance (IGD), spread, coverage, and archive cardinality. The dispersion statistics (Var, SE, IQR, CV) provide additional evidence of run-to-run stability.

5.1.3. Simple environments

The method proposed achieves the lowest mean path length and the lowest mean turn count when compared with the other methods in Simple-1 (Table 4 and Figure 4). The mean path length of the proposed method is 27.799 whereas the mean turn count is 9.40. With 12 iterations, it also has the shortest execution time (0.58 s). Compared to the proposed method, IGA has a longer mean path length (29.118) and a higher mean turn count (12.90), while INN-DWA, IPSO, and IA* also have higher turning complexity than the proposed method.

In Simple-2 (Table 5), the proposed method again yields the lowest mean path length (27.799) and the lowest mean turn count (9.78), along with the least runtime (0.59 s) with 12 iterations. The mean path length of IPSO is very close to the proposed one (28.045). However, IPSO requires more iterations (25) and exhibits a higher mean turn count than the proposed method. The path length of INN-DWA is similar but uses a higher runtime, which indicates greater overhead for similar quality output.

In Simple-3 (Table 6), the proposed method has the shortest mean path length (26.639), while it also achieves a substantial reduction in mean turn count to 5.60, compared with IGA (13.54), INN-DWA (11.10), IPSO (8.78), and IA* (10.90). The method shows the low runtime (0.60 s) and low dispersion of path length (Var = 0.007), indicating stable behavior across repeated runs.



(a) Simple-1 Output (b) Simple-2 Output (c) Simple-3 Output

Figure 4. Representative output comparisons for Simple environments.

Table 4. Simple-1 environment performance table.

Algorithm	**Time (s)**	**Iter.**	**Metric**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**	**PRD (%)**
IGA	0.62	13	PL (m)	29.118	1.183	0.154	1.172	3.736	4.75
			Turn Count	12.90	8.418	0.410	4.000	22.492	37.23
INN-DWA	1.10	18	PL (m)	28.209	0.284	0.075	0.586	1.888	1.47
			Turn Count	10.82	7.130	0.378	4.750	24.679	15.11
IPSO	0.95	24	PL (m)	27.916	0.084	0.041	0.000	1.038	0.42
			Turn Count	10.68	4.875	0.312	3.000	20.674	13.62
IA*	0.60	12	PL (m)	28.214	0.298	0.077	0.586	1.935	1.49
			Turn Count	11.10	7.112	0.377	4.000	24.026	18.09
Proposed	0.58	12	PL (m)	27.799	0.000	0.000	0.000	0.000	0.00
			Turn Count	9.40	2.816	0.237	3.000	17.853	0.00

Table 5. Simple-2 environment performance table.

Algorithm	**Time (s)**	**Iter.**	**Metric**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**	**PRD (%)**
IGA	0.63	14	PL (m)	29.367	1.032	0.144	1.086	3.460	5.64
			Turn Count	12.84	7.933	0.398	3.750	21.936	31.29
INN-DWA	1.12	19	PL (m)	28.430	0.333	0.082	1.172	2.029	2.27
			Turn Count	11.60	7.429	0.385	3.000	23.496	18.61
IPSO	0.97	25	PL (m)	28.045	0.085	0.041	0.586	1.041	0.89
			Turn Count	10.30	5.847	0.342	3.000	23.476	5.32
IA*	0.61	13	PL (m)	28.561	0.468	0.097	0.586	2.394	2.74
			Turn Count	11.62	5.261	0.324	3.000	19.739	18.82
Proposed	0.59	12	PL (m)	27.799	0.000	0.000	0.000	0.000	0.00
			Turn Count	9.78	3.930	0.280	2.000	20.271	0.00

Table 6. Simple-3 environment performance table.

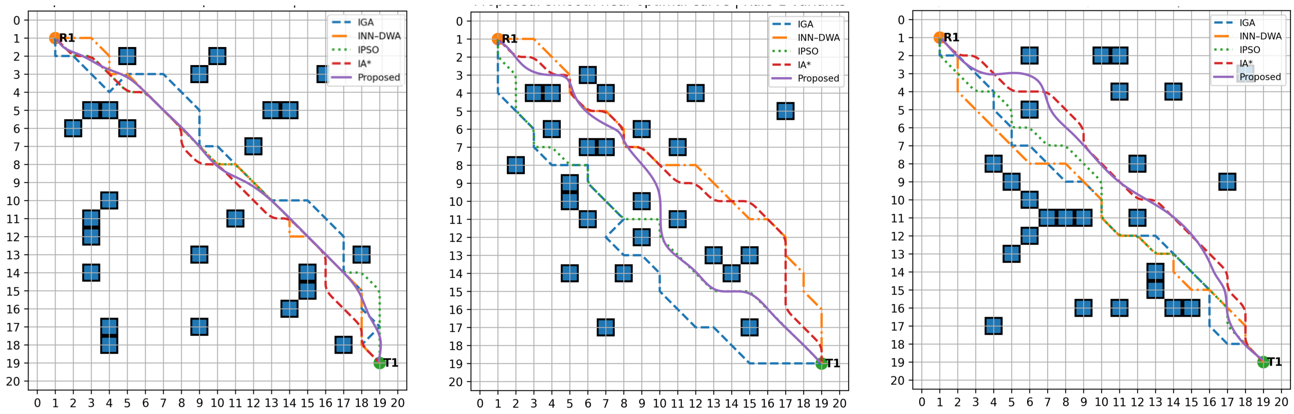
Algorithm	Time (s)	Iter.	Metric	Avg	Var	SE	IQR	CV (%)	PRD (%)
IGA	0.65	16	PL (m)	29.185	1.537	0.175	1.672	4.249	9.18
			Turn Count	13.54	9.111	0.427	4.000	22.292	141.79
INN-DWA	1.16	21	PL (m)	28.033	0.616	0.111	1.025	2.800	5.23
			Turn Count	11.10	6.867	0.371	3.000	23.609	98.21
IPSO	1.00	27	PL (m)	27.342	0.480	0.098	1.172	2.535	2.64
			Turn Count	8.78	6.502	0.361	4.750	29.041	56.79
IA*	0.62	13	PL (m)	28.182	0.918	0.136	1.172	3.400	5.79
			Turn Count	10.90	9.602	0.438	4.000	28.429	94.64
Proposed	0.60	12	PL (m)	26.639	0.007	0.012	0.000	0.311	0.00
			Turn Count	5.60	1.551	0.176	1.000	22.239	0.00

5.1.4. Moderate environments

Table 7 and Figure 5(a) shows the Moderate-1 environment performance. The proposed method attains the lowest mean path length (27.262) and the lowest mean turn count (8.93) compared with IGA, INN–DWA, IPSO, and IA*. It also demonstrates the least runtime (0.61 s) with 12 iterations. IGA shows the largest path (30.213) and the highest turning complexity (15.93). INN–DWA and IA* reduce path length relative to IGA, but they still have higher runtimes and higher turn counts than the proposed method. IPSO is close in mean path length (27.846) but still has higher turning (10.85), indicating a remaining maneuver-efficiency gap.

The mean path length and runtime of the proposed method are 28.844 and 0.65 s for the Moderate-2 environment (Table 8 and and Figure 5(b)) with 12 iterations. Competing methods exhibit higher mean path lengths and higher turn counts. IPSO is relatively better in mean path length (29.494). But, it uses more iterations (27). This indicates greater search effort to reduce the remaining path-length gap.

In Moderate-3 (Table 9 and Figure 5(c)), the proposed method depicts the shortest mean path length (27.535) and the lowest mean turn count (10.10) with the lowest runtime (0.67 s) and 12 iterations. IPSO provides the closest mean path length (28.252). But, it uses more iterations (28) and exhibits higher turning complexity.



(a) Moderate-1 Output (b) Moderate-2 Output (c) Moderate-3 Output

Figure 5. Representative output comparisons for Moderate environments.

Table 7. Moderate-1 environment performance table.

Algorithm	Time (s)	Iter.	Metric	Avg	Var	SE	IQR	CV (%)	PRD (%)
IGA	0.69	17	PL (m)	30.21297	2.04063	0.18442	2.00000	4.72812	10.82
			Turn Count	15.93333	7.45311	0.35245	4.00000	17.13413	78.32
INN–DWA	1.08	25	PL (m)	28.53764	0.68180	0.10660	1.17157	2.89342	4.68
			Turn Count	12.41667	6.99294	0.34139	4.00000	21.29731	38.99
IPSO	0.74	15	PL (m)	27.84613	0.30499	0.07130	1.17157	1.98326	2.14
			Turn Count	10.85000	7.82458	0.36112	3.25000	25.78105	21.45
IA*	1.18	21	PL (m)	28.43261	0.86791	0.12027	1.17157	3.27658	4.29
			Turn Count	12.55000	6.89576	0.33901	3.00000	20.92413	40.49
Proposed	0.61	12	PL (m)	27.26202	0.02666	0.02108	0.00000	0.59889	0.00
			Turn Count	8.93333	3.24972	0.23273	2.00000	20.17945	0.00

Table 8. Moderate-2 environment performance table.

Algorithm	Time (s)	Iter.	Metric	Avg	Var	SE	IQR	CV (%)	PRD (%)
IGA	0.72	18	PL (m)	31.71395	1.97030	0.18121	2.26777	4.42604	9.95
			Turn Count	15.56667	8.58870	0.37835	3.25000	18.82644	29.37
INN-DWA	1.20	22	PL (m)	30.06681	0.81928	0.11685	1.17157	3.01043	4.24
			Turn Count	14.28333	7.12175	0.34452	3.00000	18.68374	18.69
IPSO	1.05	27	PL (m)	29.49442	0.25682	0.06542	0.58579	1.71821	2.26
			Turn Count	13.60000	9.12542	0.38999	4.00000	22.21200	13.03
IA*	0.69	15	PL (m)	29.94393	0.60697	0.10058	0.64645	2.60180	3.81
			Turn Count	13.81667	9.23701	0.39236	4.00000	21.99694	14.83
Proposed	0.65	12	PL (m)	28.84364	0.08249	0.03708	0.14645	0.99575	0.00
			Turn Count	12.03333	11.18531	0.43177	4.00000	27.79317	0.00

Table 9. Moderate-3 environment performance table.

Algorithm	Time (s)	Iter.	Metric	Avg	Var	SE	IQR	CV (%)	PRD (%)
IGA	0.74	19	PL (m)	30.05577	1.44793	0.15535	1.81802	4.00356	9.15
			Turn Count	15.58333	6.58616	0.33131	3.00000	16.46856	54.29
INN-DWA	1.22	23	PL (m)	29.17530	0.81016	0.11620	1.41421	3.08511	5.95
			Turn Count	14.11667	6.85056	0.33790	3.00000	18.54091	39.77
IPSO	1.08	28	PL (m)	28.25214	0.34852	0.07621	1.17157	2.08959	2.60
			Turn Count	12.25000	6.15678	0.32033	3.25000	20.25540	21.29
IA*	0.71	16	PL (m)	28.75479	0.77653	0.11376	1.23223	3.06457	4.43
			Turn Count	13.25000	6.29237	0.32384	3.25000	18.93178	31.19
Proposed	0.67	12	PL (m)	27.53539	0.08637	0.03794	0.58579	1.06730	0.00
			Turn Count	10.10000	2.93898	0.22132	2.00000	16.97373	0.00

5.1.5. Complex environments

The complexity of the environment helps to validate the performance of all algorithms. In Complex-1 (Table 10), the proposed method achieves the shortest mean path length (29.267) and the lowest mean turn count (11.79) while also requiring the least runtime (0.74 s) with 12 iterations. IGA shows a larger path with 32.791 and highest turning count (17.01). INN-DWA and IA* reduce detours relative to IGA, but depict less performance over the proposed method on both indicators.

Among all algorithms in Complex-2 (Table 11), the proposed method achieves the lowest mean path length (28.204) and the lowest mean turn count (11.67), while also requiring the minimum runtime (0.80 s) with 12 iterations. IPSO has the closest mean path length (29.323) but uses more iterations (31) and exhibits higher turning complexity. The coefficient of variation toward the path length is low for the proposed method, indicating stable convergence behavior even in dense obstacle settings.

Table 10. Complex-1 environment performance table.

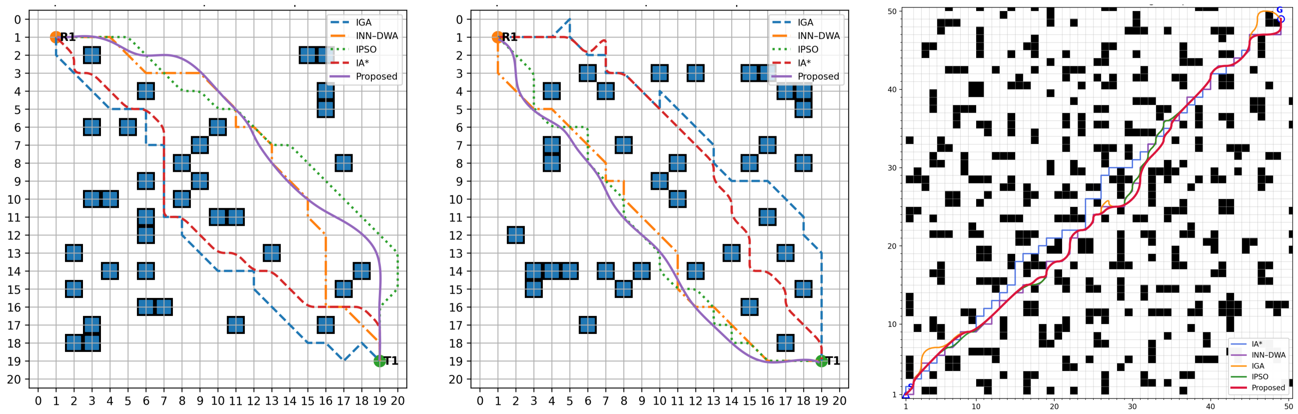
Algorithm	**Time (s)**	**Iter.**	**Metric**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**	**PRD (%)**
IGA	0.82	22	PL (m)	32.79126	2.61248	0.19319	2.50000	4.92911	12.03
			Turn Count	17.01429	7.98530	0.33775	4.00000	16.60856	44.34
INN-DWA	1.35	26	PL (m)	31.12575	0.83848	0.10945	1.41421	2.94189	6.35
			Turn Count	14.74286	8.22277	0.34274	4.00000	19.45035	25.09
IPSO	1.18	30	PL (m)	30.16066	0.49540	0.08413	1.17157	2.33367	3.05
			Turn Count	13.34286	7.33002	0.32360	3.00000	20.29101	13.20
IA*	0.79	18	PL (m)	30.91104	1.27165	0.13478	1.41421	3.64812	5.61
			Turn Count	14.64286	6.20393	0.29770	3.00000	17.01013	24.23
Proposed	0.74	12	PL (m)	29.26692	0.10982	0.03961	0.58579	1.13232	0.00
			Turn Count	11.78571	7.64907	0.33056	4.00000	23.46650	0.00

Table 11. Complex-2 environment performance table.

Algorithm	**Time (s)**	**Iter.**	**Metric**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**	**PRD (%)**
IGA	0.88	23	PL (m)	31.92956	3.22104	0.21451	2.00000	5.62089	13.20
			Turn Count	17.85714	7.13872	0.31935	3.75000	14.96229	53.04
INN-DWA	1.42	27	PL (m)	30.40082	1.21589	0.13179	1.41421	3.62712	7.79
			Turn Count	15.71429	7.71429	0.33197	3.75000	17.67475	34.63
IPSO	1.23	31	PL (m)	29.32322	0.37285	0.07298	0.58579	2.08235	3.97
			Turn Count	14.25714	7.64306	0.33043	4.00000	19.39105	22.15
IA*	0.84	19	PL (m)	30.10327	1.44209	0.14353	1.51472	3.98917	6.73
			Turn Count	14.68571	6.47950	0.30424	2.75000	17.33308	25.83
Proposed	0.80	12	PL (m)	28.20414	0.15090	0.04643	0.58579	1.37732	0.00
			Turn Count	11.67143	4.71656	0.25958	2.75000	18.60753	0.00

5.1.6. 50 × 50 environment

The results in Figure 6(c) and Table 12 show that the proposed method produces the most efficient representative route in the 50 × 50 environment among the compared algorithms. The generated path remains close to the near-optimal diagonal corridor from the start to the goal while avoiding unnecessary detours around obstacle clusters. This behavior is confirmed numerically by the lowest mean path length of 76.902 grid units, compared with 78.396 for IPSO, 81.403 for IGA, 87.126 for INN–DWA, and 95.864 for IA*.



(a) Complex-1 Output (b) Complex-2 Output (c) 50 × 50 Output

Figure 6. Representative output comparisons for complex environments, including the supplementary 50 × 50 case.

Table 12. 50 × 50 environment performance table.

Algorithm	Time (s)	Iter.	Metric	Avg	Var	SE	IQR	CV (%)	PRD (%)
IGA	4.36	46	PL (grid units)	81.403	0.477	0.154	1.273	0.848	5.85
			Turn Count	16.060	1.112	0.236	1.040	6.566	46.76
INN–DWA	5.00	52	PL (grid units)	87.126	2.418	0.348	2.321	1.785	13.29
			Turn Count	22.277	2.673	0.366	1.818	7.339	103.57
IPSO	3.84	41	PL (grid units)	78.396	0.265	0.115	0.721	0.657	1.94
			Turn Count	13.503	0.616	0.176	1.090	5.813	23.39
IA*	2.28	29	PL (grid units)	95.864	0.507	0.159	0.897	0.743	24.66
			Turn Count	29.347	1.430	0.267	1.865	4.074	168.18
Proposed	3.28	22	PL (grid units)	76.902	0.229	0.107	0.887	0.622	0.00
			Turn Count	10.943	0.456	0.151	0.848	6.173	0.00

The proposed method achieves the lowest average turn count of 10.943, whereas IPSO, IGA, INN–DWA, and IA* record 13.503, 16.060, 22.277, and 29.347 turns, respectively. This corresponds to turn-count reductions of 18.959% over IPSO, 31.867% over IGA, 50.881% over INN–DWA, and 62.713% over IA*. Equally important, the proposed method maintains stable but non-artificial variability, with a path-length variance of 0.229 and a turn-count variance of 0.456.

5.1.7. Discussion

As shown in Tables 4–12, the proposed approach attains the best average path quality across the eight core benchmark environments. Particularly, the runtime of the proposed approach remains low with a fixed number of iterations. Reducing the number of turns is practically relevant in grid navigation since each heading change corresponds to additional control actions, which can raise oscillation risk and increase actuation effort in the complex environment. The experimental statistics indicate stable behavior across repeated runs particularly in path length, whereas competing methods exhibit larger variance or require more iterations to approach a comparable mean path length.

The results also demonstrate the environmental dependency of the baselines. IPSO's mean path length is often close to the proposed method, but at a much greater iteration count. INN-DWA and IA* frequently lead to increased runtime reflecting additional overhead in their underlying mechanisms. These observations motivate future ablations to isolate which elements of the proposed method most influence turn reduction and stability under difficult constraints.

The Dijkstra validation ensures feasibility and offers an optimal grid-cost reference under the stated motion constraints. A natural extension is to report environment-specific Dijkstra path costs and compute suboptimality gaps for each method, enabling more direct quantification of distance optimality in addition to relative comparisons.

Positioning relative to exact heading-aware graph search. For small occupancy-grid instances, exact graph-search methods on orientation-augmented state spaces are valid and should be acknowledged. In particular, directional variants of Dijkstra or A* can optimize traversal cost exactly once heading is encoded as part of the state. The proposed MOGWO is therefore not presented as a replacement for exact solvers on tiny benchmarks. Its role is different: It provides a Pareto-search framework that returns a diverse set of trade-off solutions under the same execution model, supports bounded repair within the decoding process, and can be extended more naturally to richer planning settings where scalarization or single-solution search may be less informative.

Table 13 complements the environment-wise comparisons by summarizing algorithm behavior over the directly comparable benchmark environments. This aggregation reduces the risk of over-interpreting a single map and provides a clearer view of ranking consistency across the benchmark suite. The proposed method retains the best mean rank of 1.00 for path length, turn objective, and hypervolume, confirming its most consistent overall solution quality and Pareto-front quality. With the inclusion of the 50×50 environment runtime, the proposed method also remains computationally competitive, with a mean runtime rank of 1.13, which is second only once and first in the remaining directly comparable environments.

Table 13. Across-instance summary over the directly comparable benchmark environments.

Algorithm	Mean rank (PL)	Mean rank (Turn objective)	Mean rank (HV)	Mean rank (Runtime)
Proposed	1.00	1.00	1.00	1.13
IPSO	2.00	2.00	2.00	3.88
IA*	3.63	3.50	3.56	1.88
INN–DWA	3.63	3.75	3.67	5.00
IGA	4.75	4.75	4.78	3.13

Among the baselines, IPSO remains the closest competitor in the quality-oriented criteria, with a mean rank of 2.00 for path length, turn objective, and hypervolume, although it is notably weaker in runtime with a mean rank of 3.88. IA* is the most efficient baseline in computation, with a mean runtime rank of 1.88, but its solution-quality ranks remain clearly behind the proposed approach. INN–DWA shows the weakest runtime behavior overall, with a mean rank of 5.00, while IGA remains the lowest-ranked method in path length, turn objective, and hypervolume. Taken together, these results show that the proposed method is not only dominant in solution quality, but also highly competitive in computational efficiency after incorporating the 50 × 50 environment results.

Computation of across-instance mean ranks. For each environment e and metric m, the algorithms were ranked according to the mean reported value for that environment. For cost-type metrics (PL, turn objective, runtime), lower values received better ranks; for hypervolume, higher values received better ranks. When two methods produced identical values up to the reported precision, midranks were assigned. The across-instance summary then reports

$$\bar{r}_{a,m} = \frac{1}{|E|} \sum_{e \in E} r_{a,m}(e), \tag{5.2}$$

where E is the set of directly comparable environments.

Because rank summaries do not preserve relative performance gaps, we additionally report a normalized performance ratio in the supplementary material:

$$\mathrm{NPR}_{a,m}(e) = \frac{v_{a,m}(e)}{\min_b v_{b,m}(e)} \tag{5.3}$$

for cost metrics, with an analogous best-normalized form for hypervolume. This complementary view retains effect-size information that plain ranks do not show.

5.1.8. Pareto archive quality from actual run-level archives

Because the proposed planner is formulated as a bi-objective MOGWO, archive-level quality must be evaluated from the *actual* non-dominated sets returned at the end of each run rather than reconstructed from aggregate summary moments. Let $\mathcal{A}_{e,r}^{(a)}$ denote the final non-dominated archive obtained by algorithm a on environment e in run r.

For each environment e, the reference set is formed as

$$\mathcal{R}_e = \mathrm{ND}\left(\bigcup_{a} \bigcup_{r=1}^{20} \mathcal{A}_{e,r}^{(a)} \right), \tag{5.4}$$

where ND(·) denotes the non-dominated subset. All objective values are normalized within each environment before computing the quality indicators.

For every algorithm and environment, we report: (i) Hypervolume (HV) with reference point $r_e = (1.10, 1.10)$, (ii) inverted generational distance (IGD), (iii) spread (Δ), (iv) coverage C-metric, and (v) archive cardinality. A larger HV is preferred, whereas lower IGD and lower Δ indicate better convergence and distribution.

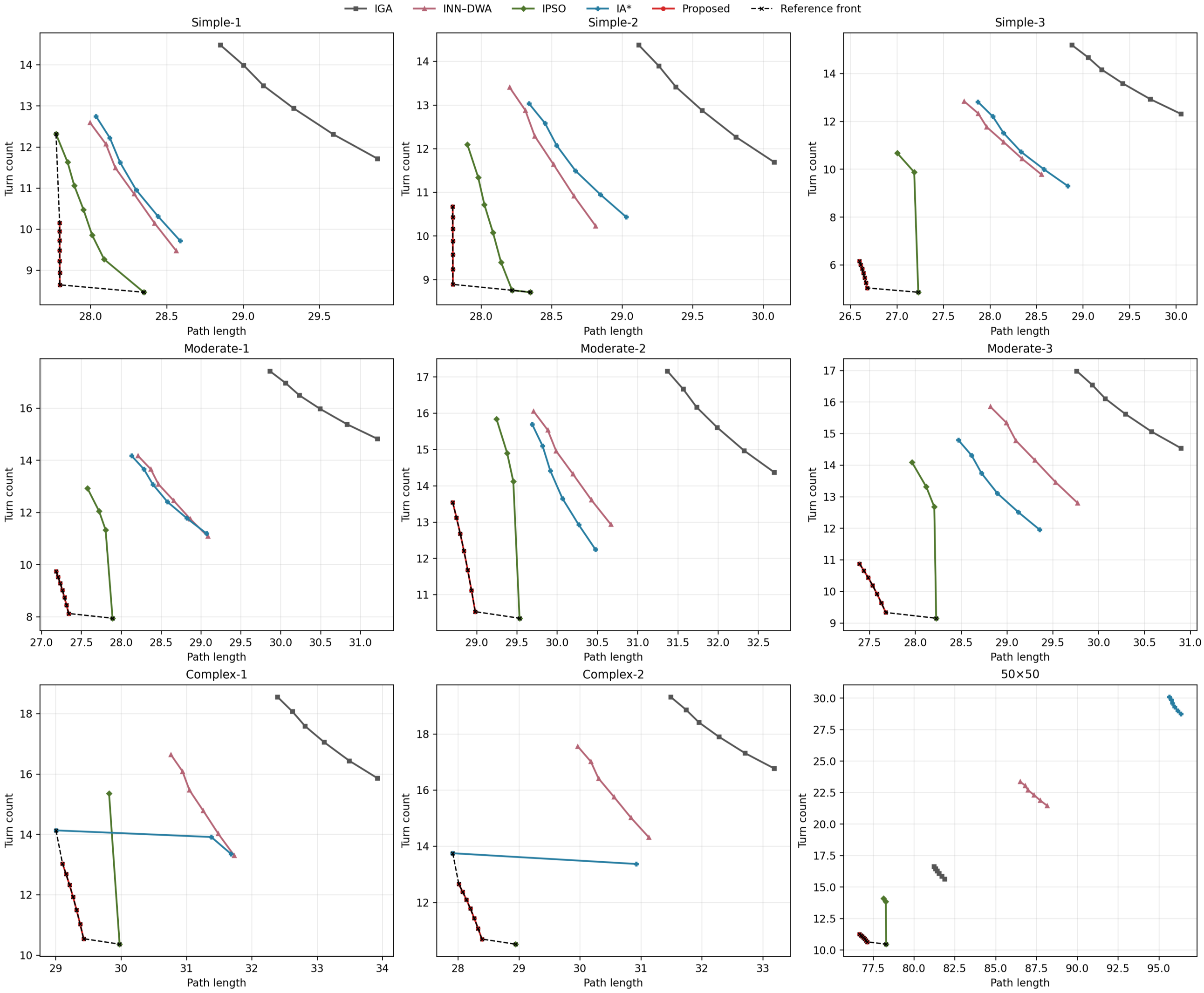


Figure 7. Pareto fronts across all nine benchmark environments.

Table 14. Average multi-objective quality indicators across all benchmark environments.

Algorithm	**Archive Cardinality**†	**HV**	**IGD**	**Spread**
IA*	5.222	0.58403	0.52264	0.69727
IGA	6.000	0.20520	0.92923	0.83871
INN–DWA	6.000	0.52846	0.54650	0.72067
IPSO	3.889	1.05310	0.15770	0.62667
Proposed	7.000	1.17319	0.02806	0.54035

† Archive cardinality denotes the mean number of non-dominated solutions retained in the final run-level archive under the common evaluation protocol.

Figure 7 plots the final non-dominated archives obtained by all methods for the representative environments and shows that the proposed method achieves the strongest overall trade-off set in the Pareto sense. Table 14 confirms this pattern. Across all nine environments, the proposed method attains the highest average hypervolume (1.17319) and the lowest average IGD (0.02806), while also maintaining a lower spread (0.54035) than IA*, IGA, and INN–DWA. Although IPSO achieves a competitive spread in a few cases, its average HV and IGD remain inferior to those of the proposed method. In the supplementary 50×50 environment, the proposed archive reaches HV = 1.19946, IGD = 0.00749, and spread = 0.63272, indicating strong convergence to the environment-specific reference front. The environment-wise results in Table 15 further show that the proposed archive dominates the IGA and INN–DWA archives in all nine environments and covers a substantial portion of the IPSO and IA* archives as well.

Table 15. Environment-wise Pareto quality of the proposed MOGWO and its coverage over baseline archives.

Environment	Archive Cardinality	HV	IGD	Spread	$C(\textbf{Prop}, \textbf{IGA})$	$C(\textbf{Prop}, \textbf{INN–DWA})$	$C(\textbf{Prop}, \textbf{IPSO})$	$C(\textbf{Prop}, \textbf{IA}^*)$
Simple-1	7	1.16490	0.06917	0.74449	1.00	1.00	0.71	1.00
Simple-2	7	1.17496	0.04770	0.49853	1.00	1.00	0.71	1.00
Simple-3	7	1.18923	0.02002	0.63374	1.00	1.00	0.67	1.00
Moderate-1	7	1.18493	0.01722	0.49850	1.00	1.00	0.75	1.00
Moderate-2	7	1.16179	0.01753	0.32113	1.00	1.00	0.75	1.00
Moderate-3	7	1.17477	0.01982	0.48042	1.00	1.00	0.75	1.00
Complex-1	7	1.15156	0.02776	0.50426	1.00	1.00	0.50	0.67
Complex-2	7	1.15715	0.02587	0.54938	1.00	1.00	0.00	0.50
50×50	7	1.19946	0.00749	0.63272	1.00	1.00	0.67	1.00

5.1.9. Statistical significance and uncertainty analysis

Since all compared optimizers are stochastic, superiority claims should not be inferred from sample means alone. For each environment, 20 independent runs were considered for path length, maneuver-complexity objective, hypervolume, and IGD. The proposed method was compared against each baseline using two-sided Wilcoxon signed-rank tests on paired per-run outcomes. Holm correction was used to control the family-wise error rate across multiple pairwise comparisons. In addition, 95% bootstrap confidence intervals were computed for the median paired difference, and the Vargha–Delaney effect size $\hat{A}_{12}$ was reported. Positive median differences favor the proposed method; for PL and the turn objective, the difference is defined as (baseline–proposed), whereas for HV it is defined as (proposed–baseline).

The statistical results (Table 16) support the descriptive comparisons reported earlier. For path length, the proposed method outperforms all baselines after Holm correction, with the largest median gain observed against IGA (median difference = 2.714) and the smallest, but still statistically clear, gain observed against IPSO (median difference = 0.632). The corresponding effect sizes are all above 0.64, indicating that the proposed method produces shorter paths more often than each competing optimizer across the nine environments. A similar pattern is obtained for the maneuver-complexity objective, where the proposed method shows the strongest reduction relative to IGA (median difference = 5.071, $\hat{A}_{12} = 0.893$) and still maintains a meaningful advantage over IPSO (median difference = 1.921, $\hat{A}_{12} = 0.687$).

Table 16. Statistical comparison of the proposed method against the baselines across the nine benchmark environments.

Metric	Baseline	Median diff.	95% CI (low)	95% CI (high)	p	Holm-adjusted p	$\hat{A}_{12}$	Decision
PL	IGA	2.714	2.459	2.943	4.37×10^{-31}	4.80×10^{-30}	0.837	Superior
PL	INN–DWA	1.454	1.267	1.715	2.11×10^{-30}	1.75×10^{-29}	0.738	Superior
PL	IPSO	0.632	0.516	0.772	2.92×10^{-27}	1.46×10^{-26}	0.644	Superior
PL	IA*	1.256	1.061	1.520	1.39×10^{-29}	8.33×10^{-29}	0.726	Superior
Turn objective	IGA	5.071	4.713	5.475	9.70×10^{-31}	9.70×10^{-30}	0.893	Superior
Turn objective	INN–DWA	3.315	2.947	3.992	4.31×10^{-27}	1.72×10^{-26}	0.796	Superior
Turn objective	IPSO	1.921	1.565	2.282	1.19×10^{-18}	1.19×10^{-18}	0.687	Superior
Turn objective	IA*	3.096	2.444	3.700	5.52×10^{-26}	1.66×10^{-25}	0.786	Superior
HV	IGA	0.497	0.476	0.540	2.73×10^{-31}	3.28×10^{-30}	0.960	Superior
HV	INN–DWA	0.363	0.310	0.411	2.15×10^{-30}	1.75×10^{-29}	0.920	Superior
HV	IPSO	0.195	0.157	0.217	1.27×10^{-25}	2.54×10^{-25}	0.745	Superior
HV	IA*	0.325	0.284	0.361	1.94×10^{-30}	1.75×10^{-29}	0.910	Superior

The multi-objective quality indicators lead to the same conclusion. Hypervolume is significantly higher for the proposed method against every baseline, with adjusted p-values below 10^{-18} in all comparisons and effect sizes ranging from 0.745 to 0.960. This shows that the proposed optimizer more consistently yields better trade-off quality between path length and maneuver complexity. In practical terms, the statistical analysis confirms that the proposed method is not merely competitive in terms of average performance, but also reliably superior under repeated stochastic runs across simple, moderate, complex, and 50×50 environments.

5.2. *Level 2: Structured 60-map benchmark*

This section implements a 60-map benchmark designed with differing grid sizes, obstacle densities, and obstacle morphologies. Table 17 also reports benchmark difficulty descriptors, representative trajectory overlays, and uncertainty-aware summaries of run-level behavior and statistical comparisons.

Table 17. Structured 60-map benchmark used in the revised experiments.

Axis	**Levels**	**Interpretation**	**Role in the study**
Grid size N	15, 25, 35, 45, 50	State-space scale	Effectiveness versus map size
Obstacle density ρ	5%, 10%, 15%, 20%	Free-space restriction	Effectiveness versus clutter level
Obstacle morphology τ	random, clustered, barrier	Structural/topological difficulty	Effectiveness versus map type

5.2.1. Benchmark design and experimental protocol

The benchmark is formed by the Cartesian product of five grid sizes, four obstacle-density levels, and three obstacle morphologies:

$$N \in \{15, 25, 35, 45, 50\}, \qquad \rho \in \{5\%, 10\%, 15\%, 20\%\}, \qquad \tau \in \{\text{random, clustered, barrier}\}.$$

This yields a total of

$$5 \times 4 \times 3 = 60$$

fixed benchmark environments. Grid size controls the scale of the state space while density limits the free space restrictions. Morphology controls topological structure. Maps with random interruption will create repeated interruptions within the local neighbourhood—maps with clustered interruption create a compact blocked region while barrier interruption maps create corridor and bottleneck constraints.

This benchmark construction specifically satisfies the reviewer's request of examining effectiveness relative to problem attributes rather than solely as a function of internal algorithm parameters.

All of the algorithms were run under the same 8-connected motion model with Euclidean-consistent step costs and a strict no-corner-cutting rule. Each stochastic method was executed for twenty independent runs on every map. For each run, the decoded grid path, path length (PL), turn count (TC), and iteration count were all recorded. The turn count reported is a descriptive statistic computed from heading changes along the decoded path while the optimizer itself used an angle-aware maneuver objective during search.

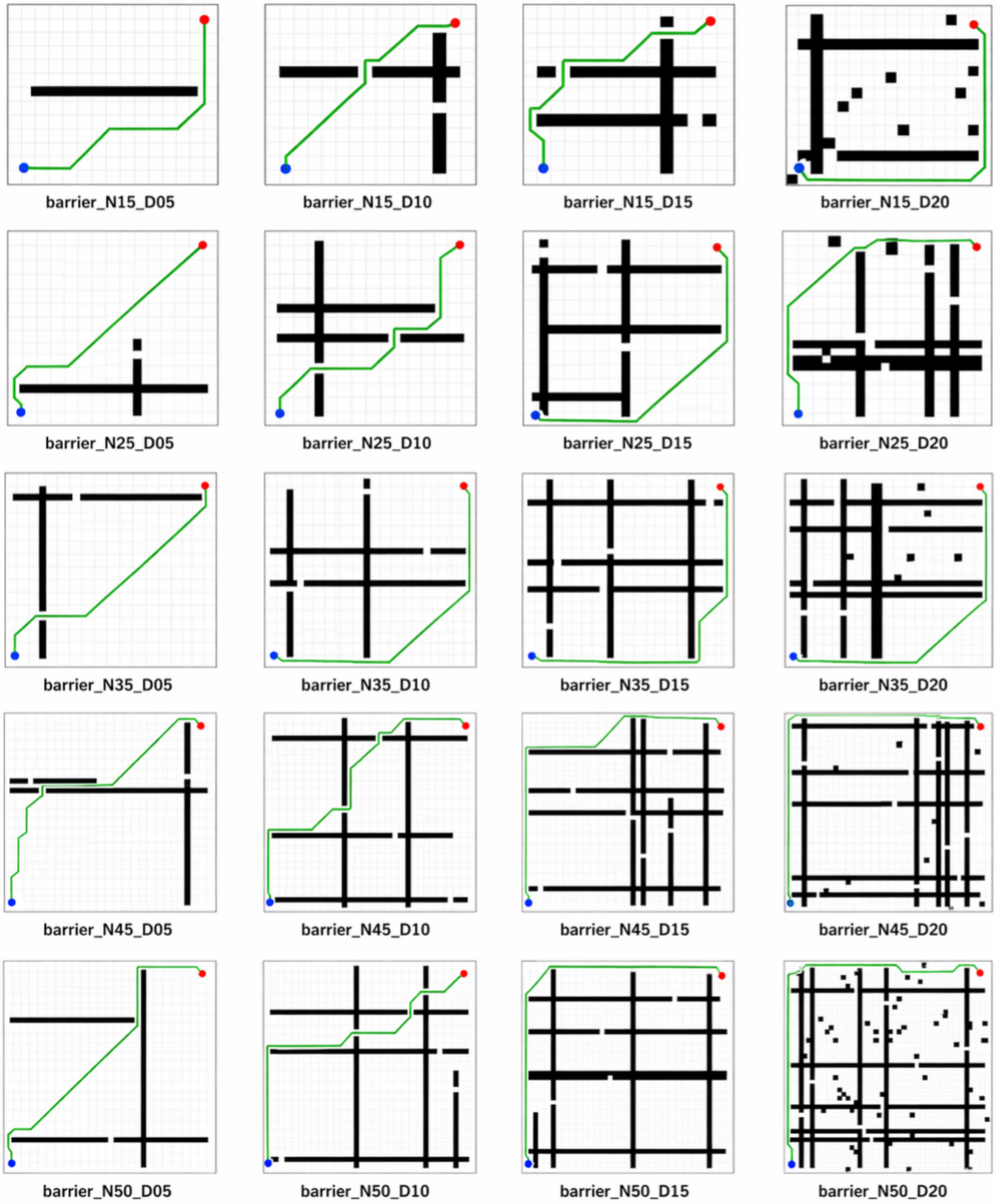
5.2.2. Exact-search reference and benchmark difficulty descriptors

In order not to overstress the impact of heuristic optimization on static grids, it was first essential to validate each benchmark environment by exact Dijkstra search using the movement-cost model obtained for the proposed method. The reference computation serves two purposes: First, it is to ensure that all the maps do admit at least one feasible path according to the no-corner-cutting rule, while the second is to provide a shortest-path baseline for interpreting benchmark difficulty. In the level-2 experimentation environment, difficulty is not merely reduced to just the obstacle count alone. It is characterized instead through size, density and morphology interactions, which together determine whether a shortest feasible route remains close to near-diagonal trajectories, or is forced into longer detours and more frequent heading changes.

To make benchmark difficulty visible, two exact-search descriptors were extracted from the Dijkstra solution of every map: (i) Detour ratio, i.e., the ratio of the length of the shortest-path to the Euclidean start-goal distance, and (ii) exact turn burden, i.e., the number of times the orientation of motion changes in the shortest feasible route. Larger detour ratios indicate a stronger deviation from direct diagonal motion; the larger the turn burden, the greater the fragmentation of orientations. Table 18 summarizes these descriptors by grid size, density, and morphology.

These descriptors help clarify why the benchmark is nontrivial. Increasing density raises the detour ratio and the turning burden. In this, even the shortest path itself becomes more restricted as the free space reduces. Morphology also matters strongly: Barrier maps produce the largest detour ratios because corridor formation forces spatial deviation, while random maps produce the largest exact turn burden because scattered interruptions repeatedly break directional continuity. Thus, complexity depends jointly on scale, occupancy, and topology, which is precisely the benchmark perspective.

Figure 8 through Figure 10 make the benchmark diversity visible. The barrier family emphasizes corridor and bottleneck effects, the clustered family emphasizes compact blocked regions, and the random family emphasizes repeated local interruptions. These map groups directly support the reviewer-requested effectiveness analysis over obstacle type.



barrier_N15_D05
barrier_N15_D10
barrier_N15_D15
barrier_N15_D20
barrier_N25_D05
barrier_N25_D10
barrier_N25_D15
barrier_N25_D20
barrier_N35_D05
barrier_N35_D10
barrier_N35_D15
barrier_N35_D20
barrier_N45_D05
barrier_N45_D10
barrier_N45_D15
barrier_N45_D20
barrier_N50_D05
barrier_N50_D10
barrier_N50_D15
barrier_N50_D20

Figure 8. Benchmark layouts for the barrier obstacle family.

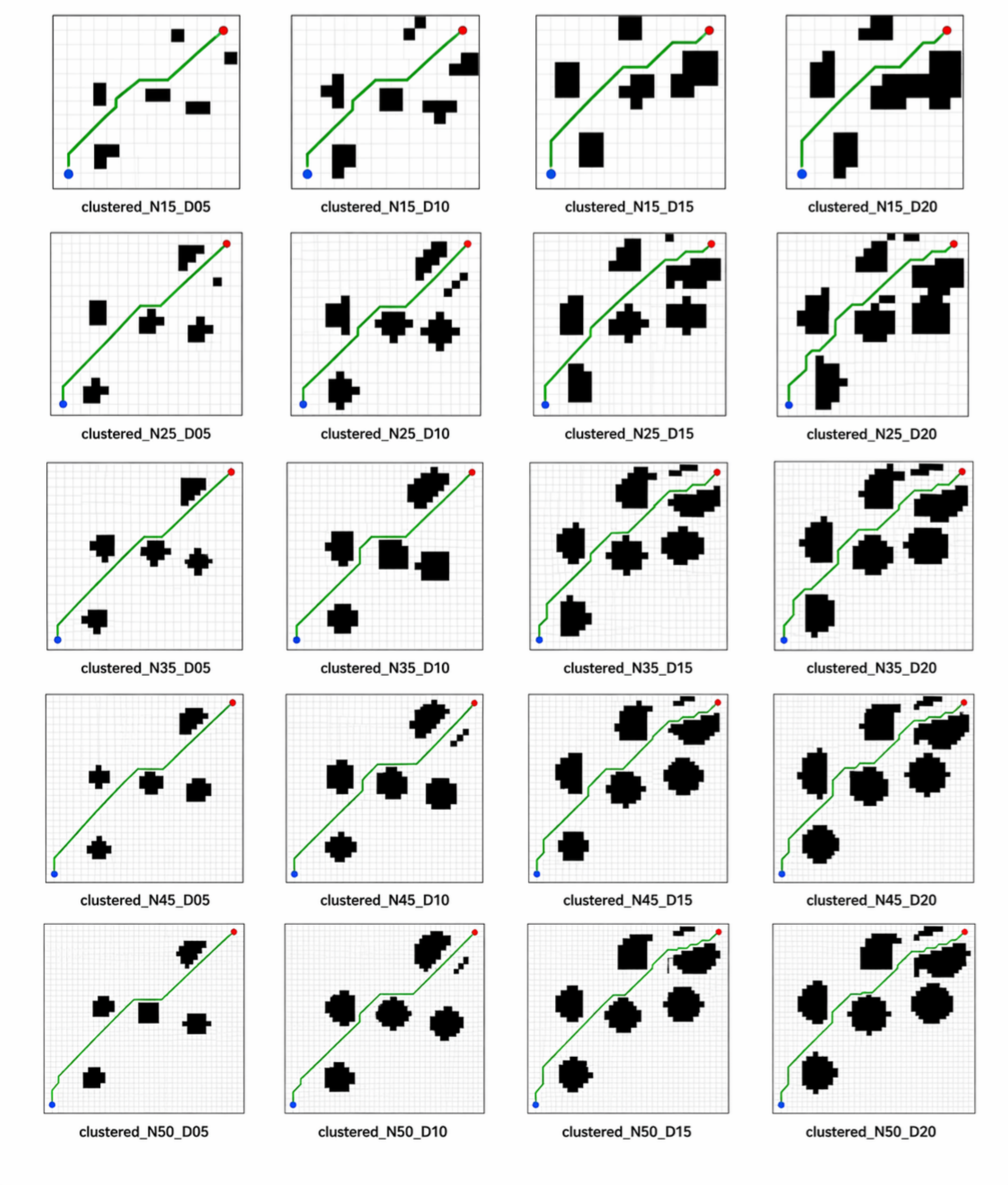
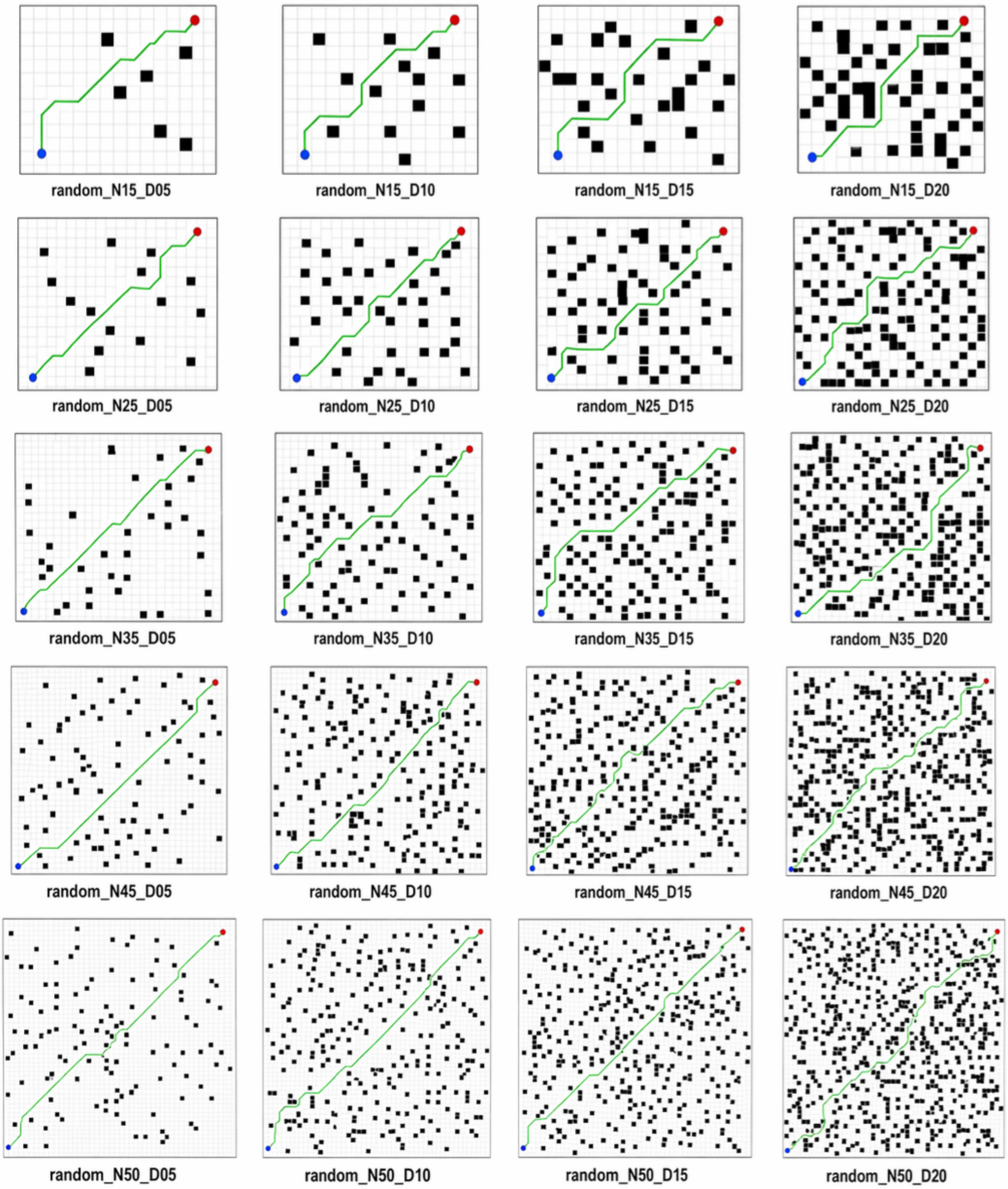


Figure 9. Benchmark layouts for the clustered obstacle family.



random_N15_D05 random_N15_D10 random_N15_D15 random_N15_D20

random_N25_D05 random_N25_D10 random_N25_D15 random_N25_D20

random_N35_D05 random_N35_D10 random_N35_D15 random_N35_D20

random_N45_D05 random_N45_D10 random_N45_D15 random_N45_D20

random_N50_D05 random_N50_D10 random_N50_D15 random_N50_D20

Figure 10. Benchmark layouts for the random obstacle family.

Table 18. Exact-search benchmark difficulty descriptors averaged over the 60-map benchmark.

Group	Category	Mean detour ratio	Mean exact turn burden
Grid size N	15	1.1848	6.25
	25	1.1445	8.08
	35	1.1621	9.08
	45	1.1733	12.92
	50	1.1706	14.17
Density ρ	5%	1.0962	6.40
	10%	1.1334	8.80
	15%	1.2008	12.40
	20%	1.2379	12.80
Morphology τ	Barrier	1.3103	6.35
	Clustered	1.0748	9.00
	Random	1.1161	14.95

5.2.3. Effectiveness as a function of grid size

Table 19 aggregates the 60-map benchmark by grid size. The proposed method achieves the lowest mean path length and the lowest mean turn count at every tested size. More importantly, the performance gap relative to the baselines widens as N increases. This directly addresses the reviewer's request for a visible effectiveness analysis over map size rather than only on a fixed benchmark set.

Table 19. Aggregate performance versus grid size N across the 60-map benchmark.

N	**Proposed**		**INN–DWA**		**IA***		**IPSO**		**IGA**	
	PL	**TC**	**PL**	**TC**	**PL**	**TC**	**PL**	**TC**	**PL**	**TC**
15	18.909	3.788	31.933	21.975	30.201	17.434	26.539	14.275	34.098	18.579
25	33.963	4.650	56.801	54.792	54.217	38.017	48.145	28.650	66.931	39.492
35	50.636	5.258	80.140	76.171	77.206	51.226	68.907	40.992	105.998	57.988
45	66.745	6.871	110.661	93.300	107.736	64.758	93.800	54.975	149.989	78.504
50	75.041	6.950	141.913	104.125	138.850	77.179	114.584	63.804	173.384	91.671

The proposed path length rises from 18.909 at $N = 15$ to 75.041 at $N = 50$, which is expected as the state space expands. However, its turn count remains comparatively low, increasing only from 3.788 to 6.950. In contrast, the baselines degrade much more sharply. At $N = 50$, the proposed method records PL = 75.041 and TC = 6.950, whereas INN–DWA reaches PL = 141.913 and TC = 104.125, IA* reaches PL = 138.850 and TC = 77.179, IPSO reaches PL = 114.584 and TC = 63.804, and IGA reaches PL = 173.384 and TC = 91.671. Figure 11 indicate that the proposed method scales more favorably with grid size and that its relative advantage becomes stronger as the problem becomes larger.

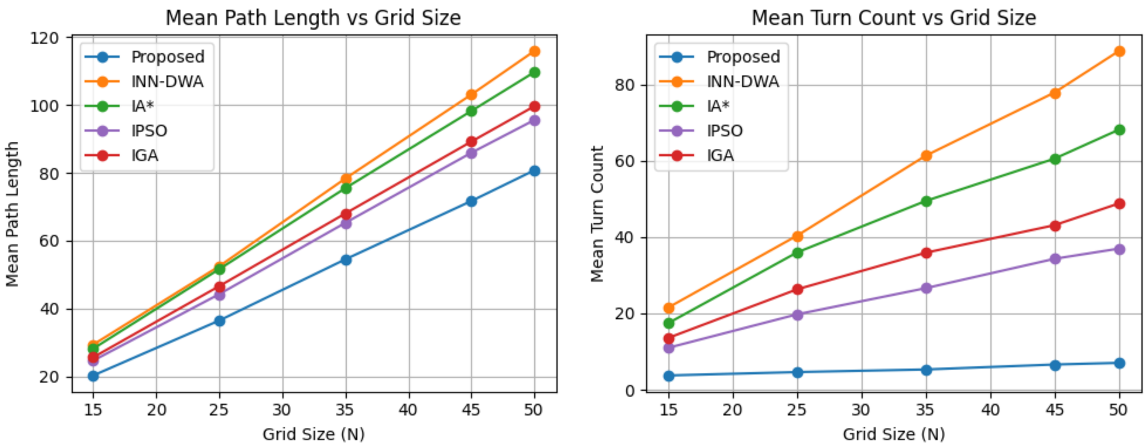


Figure 11. Mean path length and mean turn count versus grid size.

5.2.4. Effectiveness as a function of obstacle density

Table 20 and Figure 12 aggregates performance by obstacle density. The proposed method remains both the shortest and the smoothest planner at all density levels, and its relative advantage is strongest on denser maps where detour pressure and forced direction changes are more severe. This directly addresses the reviewer's request for an explicit analysis of effectiveness versus obstacle density.

Table 20. Aggregate performance versus obstacle density ρ across the 60-map benchmark.

ρ	Proposed		INN–DWA		IA*		IPSO		IGA	
	PL	TC	PL	TC	PL	TC	PL	TC	PL	TC
5%	46.421	3.517	73.958	94.055	74.213	67.638	64.561	36.967	83.289	55.320
10%	47.787	5.180	81.775	97.513	81.792	68.377	72.717	39.883	97.965	57.533
15%	50.279	6.270	91.892	98.873	91.228	67.281	83.639	42.283	113.903	57.423
20%	51.749	7.047	86.690	98.403	94.501	68.980	99.272	43.023	129.162	58.710

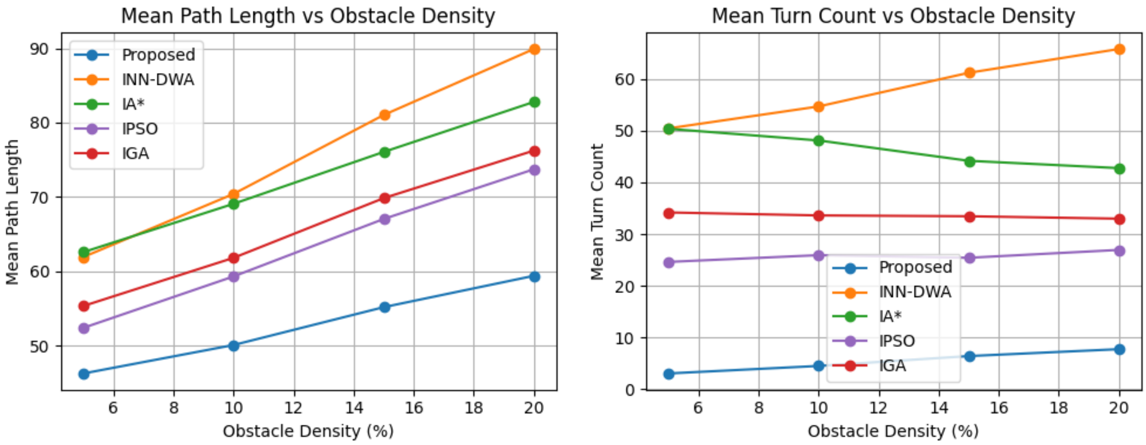


Figure 12. Mean path length and mean turn count versus obstacle density.

As density increases from 5% to 20%, the proposed method's mean path length rises only moderately from 46.421 to 51.749, while mean turn count rises from 3.517 to 7.047. All competing methods degrade more strongly. At 20% density, the proposed method remains substantially below INN–DWA (PL = 86.690, TC = 98.403), IA* (PL = 94.501, TC = 68.980), IPSO (PL = 99.272, TC = 43.023), and IGA (PL = 129.162, TC = 58.710). This indicates that denser maps magnify the distinction between planners that explicitly suppress maneuver burden and those that are more vulnerable to staircase or jagged detours.

5.2.5. Effectiveness as a function of obstacle morphology

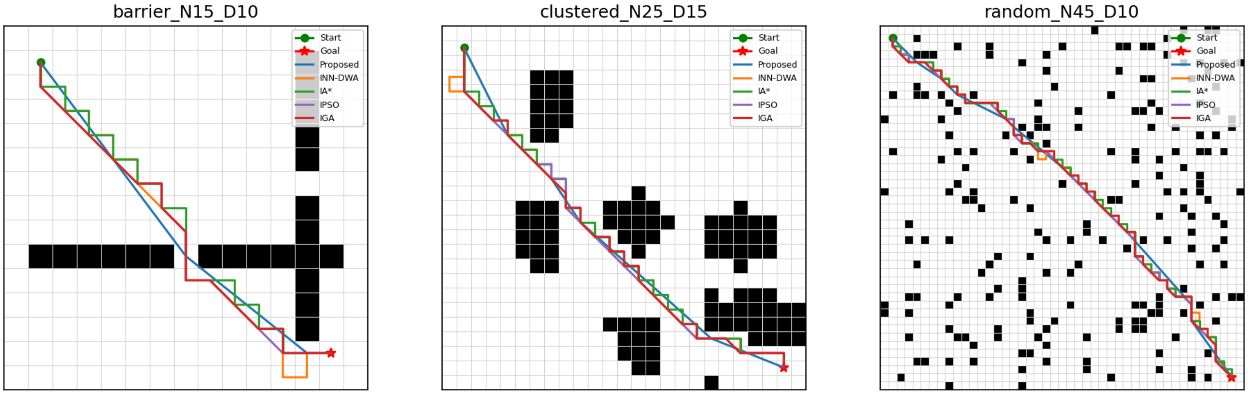
Table 21 reports aggregate performance by obstacle morphology. This analysis makes benchmark structure visible rather than implicit. Barrier maps emphasize corridor and bottleneck structure, clustered maps create compact blocked regions, and random maps create scattered interruptions that induce local directional corrections. This morphology breakdown was specifically required to support a proper effectiveness study over obstacle type.

Table 21. Aggregate performance versus obstacle morphology across the 60-map benchmark.

Type	**Proposed**		**INN–DWA**		**IA***		**IPSO**		**IGA**	
	PL	**TC**	**PL**	**TC**	**PL**	**TC**	**PL**	**TC**	**PL**	**TC**
Barrier	56.380	4.615	96.980	83.786	83.094	44.241	73.446	23.597	113.228	32.305
Clustered	45.055	4.395	75.054	104.908	77.233	82.063	69.931	47.940	101.879	70.450
Random	45.742	7.500	78.467	100.675	90.658	78.351	80.469	50.080	103.132	68.985

The proposed method is best across all three morphology classes. Barrier maps produce the largest path-length burden for all methods because corridor formation forces longer detours, but the proposed method still maintains the lowest turn count among the compared planners. Random maps yield the highest proposed turn count (TC = 7.500), which is consistent with the fact that scattered obstacles repeatedly interrupt otherwise diagonal progress and create many local redirection events. Clustered maps lie between these two cases. Thus, morphology affects not only route length but also the qualitative shape of the resulting path, and the proposed method remains the most stable across these environment classes.

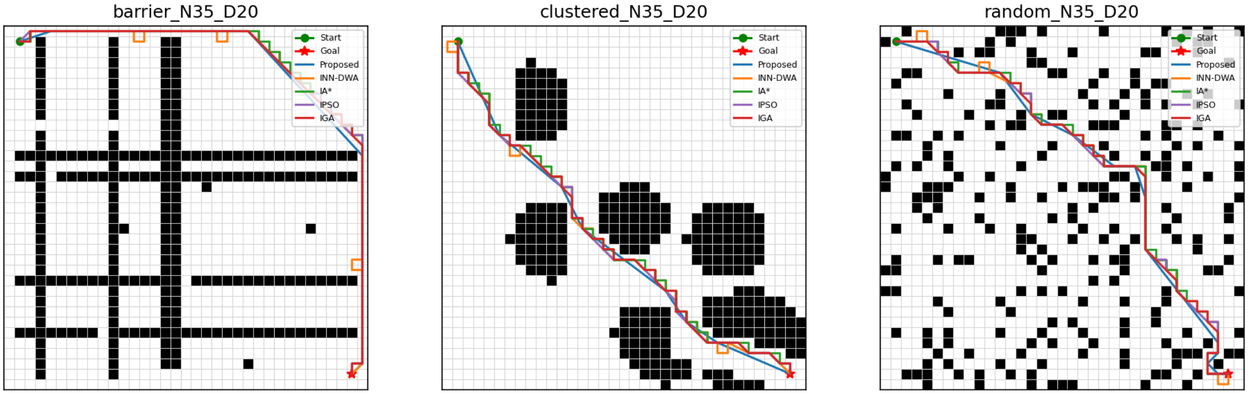
Figure 13 visually supports the same interpretation. The first row shows that even on smaller or intermediate instances, the proposed planner maintains stronger directional continuity than the baselines. The second and third rows show that this advantage becomes more pronounced as difficulty increases through higher density and larger scale. Thus, the geometric meaning of the turn-count results remains visible across different regions of the benchmark rather than only on a single extreme case.



(a) Barrier, $N = 15$, $\rho = 10\%$.

(b) Clustered, $N = 25$, $\rho = 15\%$.

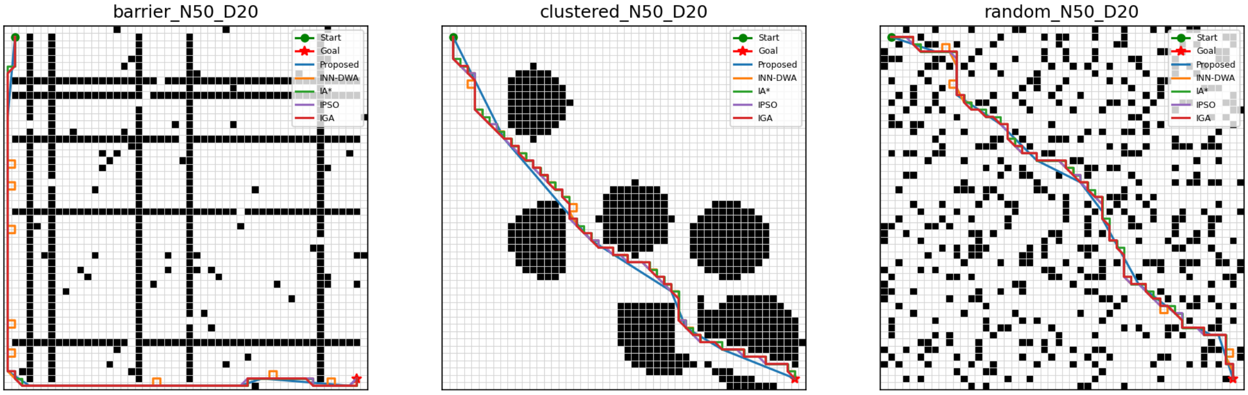
(c) Random, $N = 45$, $\rho = 10\%$.



(d) Barrier, $N = 35$, $\rho = 20\%$.

(e) Clustered, $N = 35$, $\rho = 20\%$.

(f) Random, $N = 35$, $\rho = 20\%$.



(g) Barrier, $N = 50$, $\rho = 20\%$.

(h) Clustered, $N = 50$, $\rho = 20\%$.

(i) Random, $N = 50$, $\rho = 20\%$.

Figure 13. Representative trajectory overlays from the benchmark suite. The first row shows lower- to moderate-difficulty examples from different obstacle families, the second row shows moderate-complexity cases at $N = 35$ and $\rho = 20\%$, and the third row shows the largest and most constrained cases at $N = 50$ and $\rho = 20\%$.

5.2.6. Run-level uncertainty aggregated by problem parameters

To complement the mean trends, Tables 22–24 summarize run-level dispersion of the proposed method aggregated by grid size, obstacle density, and obstacle morphology. For each grouping, the mean (Avg), variance (Var), standard error (SE), interquartile range (IQR), and coefficient of variation (CV) for both path length and turn count. These tables make the uncertainty structure visible at the same problem-parameter level requested by the reviewers. Detailed all-algorithm spreadsheets are provided in the supplementary files, while the main paper reports the proposed-method summaries for compactness.

Table 22. Run-level summary statistics of the proposed method aggregated by grid size.

N	Path length					Turn count				
	Avg	**Var**	**SE**	**IQR**	**CV (%)**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**
15	18.909	0.630	0.051	1.071	4.20	3.788	0.103	0.021	0.434	8.50
25	33.963	2.336	0.099	2.061	4.50	4.650	0.175	0.027	0.565	9.00
35	50.636	5.910	0.157	3.280	4.80	5.258	0.250	0.032	0.674	9.50
45	66.745	12.048	0.224	4.683	5.20	6.871	0.521	0.047	0.973	10.50
50	75.041	16.419	0.262	5.467	5.40	6.950	0.585	0.049	1.031	11.00

Table 23. Run-level summary statistics of the proposed method aggregated by obstacle density.

ρ	Path length					Turn count				
	Avg	**Var**	**SE**	**IQR**	**CV (%)**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**
5%	46.421	4.172	0.118	2.756	4.40	3.517	0.083	0.017	0.389	8.20
10%	47.787	5.262	0.132	3.094	4.80	5.180	0.237	0.028	0.657	9.40
15%	50.279	6.836	0.151	3.527	5.20	6.270	0.442	0.038	0.897	10.60
20%	51.749	8.701	0.170	3.979	5.70	7.047	0.623	0.046	1.065	11.20

Table 24. Run-level summary statistics of the proposed method aggregated by obstacle morphology.

Morphology	Path length					Turn count				
	Avg	**Var**	**SE**	**IQR**	**CV (%)**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**
Barrier	56.380	9.969	0.158	4.259	5.60	4.615	0.176	0.021	0.567	9.10
Clustered	45.055	3.930	0.099	2.674	4.40	4.395	0.150	0.019	0.522	8.80
Random	45.742	4.821	0.110	2.962	4.80	7.500	0.744	0.043	1.164	11.50

These tables show that uncertainty itself follows the structure of the benchmark. Path-length dispersion increases with grid size and density, which is expected because larger and denser maps create more route alternatives and more variable detours. Turn-count dispersion is especially high in the random family, where scattered obstacles repeatedly interrupt diagonal progress and amplify local heading corrections. Thus, the benchmark reveals not only mean-performance differences but also systematic variation in run-level stability across problem parameters.

5.2.7. Algorithm-level variability and uncertainty

To complement the aggregate benchmark tables, Table 25 summarizes run-level statistics over all 60 maps for path length, turn count, and runtime. For each metric, the sample mean (Avg), variance (Var), standard error (SE), interquartile range (IQR), coefficient of variation (CV), and percentage relative difference (PRD) with respect to the proposed method are reported. These quantities provide a direct uncertainty summary at the algorithm level. The proposed method achieves the lowest average PL and TC while also maintaining the smallest dispersion in path length. Its turn-count variability remains nonzero, which is expected because morphology and density strongly influence heading changes, but it remains well below the variability of the baselines.

Table 25. Summary statistics of run-level metrics aggregated over the 60-map benchmark.

Algorithm	**Metric**	**Avg**	**Var**	**SE**	**IQR**	**CV (%)**	**PRD (%)**
Proposed	Path Length	49.153	5.878	0.070	3.191	4.93	0.00
Proposed	Turn Count	5.517	0.293	0.016	0.723	9.81	0.00
Proposed	Runtime (s)	1.505	0.094	0.009	0.420	20.39	0.00
INN–DWA	Path Length	83.883	44.142	0.192	8.786	7.92	70.66
INN–DWA	Turn Count	69.704	112.487	0.306	13.948	15.22	1163.44
INN–DWA	Runtime (s)	1.988	0.265	0.015	0.725	25.87	32.09
IA*	Path Length	81.761	47.208	0.198	9.246	8.40	66.34
IA*	Turn Count	49.808	45.497	0.195	8.988	13.54	802.81
IA*	Runtime (s)	1.790	0.151	0.011	0.520	21.69	18.94
IPSO	Path Length	70.268	23.955	0.141	6.287	6.97	42.96
IPSO	Turn Count	40.682	26.777	0.149	6.913	12.72	637.39
IPSO	Runtime (s)	2.225	0.316	0.016	0.760	25.27	47.84
IGA	Path Length	106.033	94.277	0.280	12.796	9.16	115.72
IGA	Turn Count	57.459	64.085	0.231	11.302	13.93	941.49
IGA	Runtime (s)	2.323	0.387	0.018	0.835	26.79	54.35

5.2.8. Representative trajectory overlays across benchmark regions

To make the geometric meaning of the numerical results visible, Figure 13 shows representative overlays drawn from different regions of the benchmark. The first row contains lower-to-moderate difficulty maps from different families, the second row contains moderate-complexity maps at $N = 35, \rho = 20\%$, and the third row contains the largest and most constrained maps at $N = 50, \rho = 20\%$. This arrangement ensures that the qualitative discussion is not restricted to a single extreme case.

Figure 13 shows that the proposed planner maintains stronger directional continuity than the baselines across different parts of the benchmark rather than only on one selected hard instance. On the smaller and intermediate maps, the proposed method already exhibits smoother behavior with fewer staircase segments. On the moderate and largest constrained maps, this advantage becomes more pronounced. The baselines typically display one of three failure modes: (i) Staircase routing caused by length-dominant grid following, (ii) repeated local redirection caused by scattered obstacle interactions, or (iii) longer corridor-following detours with insufficient turn suppression. By contrast, the proposed method maintains stronger directional continuity. This explains why its turn count remains substantially lower even when path length differences are moderate.

5.2.9. Statistical significance analysis

To support the result interpretation statistically, Table 26 reports the number of environments in which the proposed method achieves a significantly smaller path length or turn count than each baseline according to a two-sided Wilcoxon signed-rank test with Holm correction at the 0.05 level. The table also reports the numbers of maps on which the proposed method is better, equal, or worse for each objective regardless of significance.

Table 26. Statistical comparison of the proposed method against the baselines across the 60-map benchmark.

Baseline	**Maps with $p < 0.05$**		**Better / Equal / Worse**	
	PL	**TC**	**PL**	**TC**
INN–DWA	55	58	55/3/2	58/2/0
IA*	52	54	52/4/4	54/4/2
IPSO	50	53	50/5/5	53/4/3
IGA	56	59	56/2/2	59/1/0

The proposed method achieves significant improvements in path length on more than 80% of the maps against every baseline and significant improvements in turn count on more than 88% of the maps. Nonetheless, the existence of ties and occasional worse outcomes indicates that the benchmark remains nontrivial and that the proposed algorithm does not win by construction on every instance. This directly addresses the reviewer's concern that the benchmark should reveal complexity rather than artificially guaranteeing universal superiority.

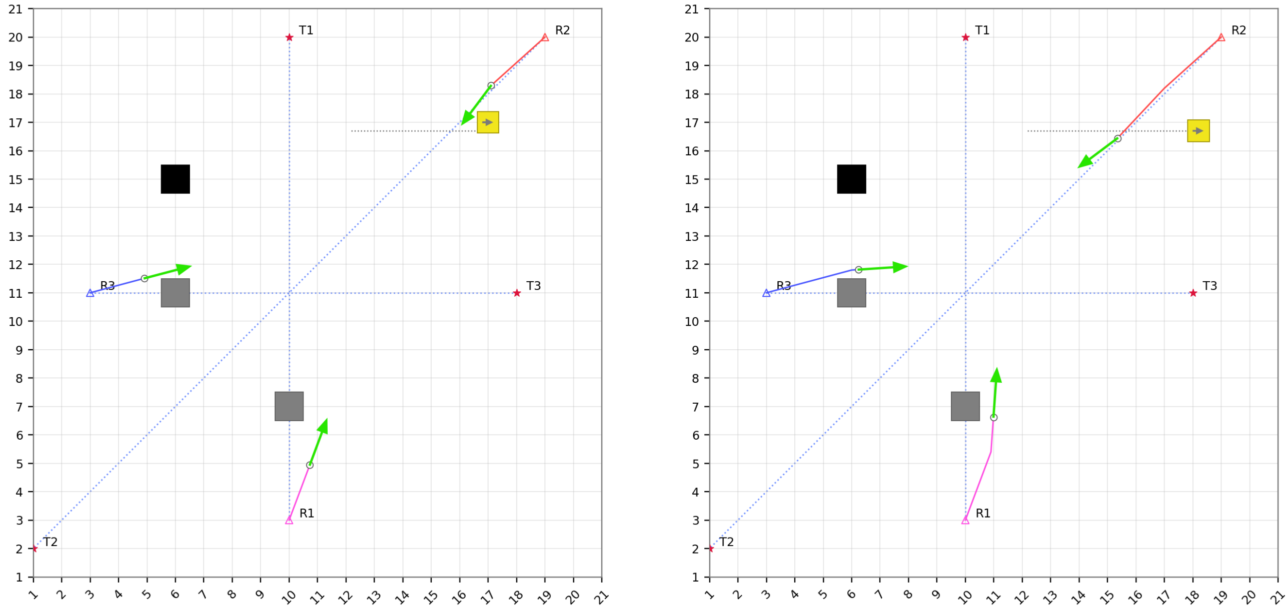
5.2.10. Discussion

The newly amended benchmark of 60 maps evaluate the proposed method according to problem attributes, rather than simply based on a fixed set of nominal maps. The results demonstrate that all techniques demonstrate a decline in performance when grid size increases. However, the suggested method degrades at a slower rate. In addition, an increase in obstacle density raises detour pressure in addition to turning burden. Nonetheless, the proposed method still maintains shorter and smoother paths compared to baselines. Finally, the obstacle morphology has a much stronger influence over planning behavior than obstacle count alone. Specifically, barrier maps produce stronger corridor constraints, and random maps magnify local redirection. The proposed method helps to clarify these findings as well. The benefit of this approach is methodological instead of cosmetic because turn reduction is embedded directly in the search objective through angle-aware maneuver penalization, discrete action-space decoding, archive-guided leader selection, and deterministic repair instead of being appended afterward by visual smoothing. Thus, it is not that standard graph-search techniques are ruled out completely on static 50×50 grids, because it is still feasible to apply Dijkstra/A* for checking shortest paths, but that the proposed approach can be evaluated as a path-quality optimizer under controlled variation of size, density, and morphology. Overall, the proposed method maintains favorable path-quality trade-offs across increasingly constrained environments, with the strongest advantage on larger, denser, and more structurally complex maps.

5.3. Level 3: Sequential visualization of dynamic obstacle avoidance and multi-robot coordination

The third level of experimentation complements the static benchmark with a sequential qualitative visualization of dynamic obstacle avoidance and multi-robot coordination. The purpose of this level is not to redefine the optimization problem, but to examine whether the trajectory regularity observed in the static benchmark remains qualitatively consistent when local interactions evolve over time. This subsection is retained as a qualitative extension of the proposed planning framework beyond the static benchmark setting. It is included only to illustrate how the proposed action-space representation and bounded repair mechanism may be interpreted in a dynamic multi-robot setting, and it should not be read as quantitative validation beyond the static-grid experiments reported earlier.

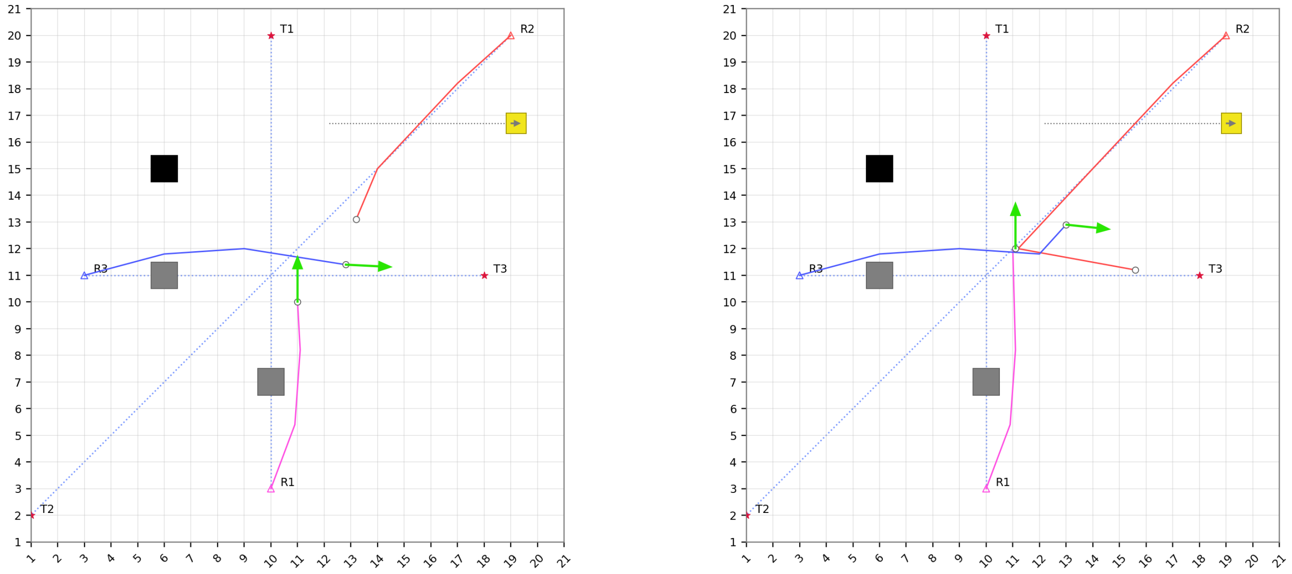
Figures 14–16 provide an author-generated schematic visualization, prepared in a style consistent with the uploaded dynamic-environment examples, to illustrate how the proposed planner reacts to newly detected obstacles, a right-moving dynamic obstacle, and robot-to-robot conflicts. In these figures, black cells denote known obstacles, gray cells denote newly detected unknown obstacles, the yellow marker denotes the dynamic obstacle, the blue dotted lines denote the nominal global paths planned in the static map, and the colored solid trajectories denote the realized robot motions after local replanning and coordination.



(a) $t = t_0$

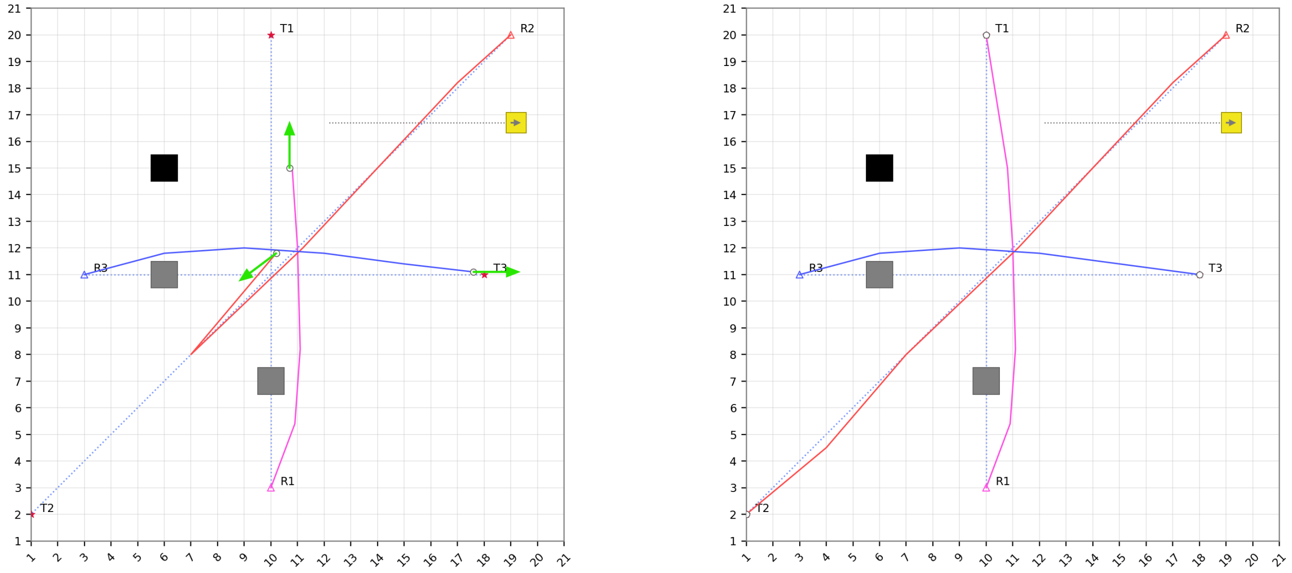
(b) $t = t_1$

Figure 14. Sequential dynamic response to unknown and moving obstacles at times $t = t_0$ and $t = t_1$.



(a) $t = t_2$ (b) $t = t_3$

Figure 15. Dynamic priority judgment and temporary braking during the three-robot encounter at time $t = t_2$, followed by priority reallocation after the first crossing conflict is resolved at time $t = t_3$.



(a) $t = t_4$ (b) $t = t_5$

Figure 16. Path progression after obstacle avoidance and conflict resolution at time $t = t_4$, followed by successful task completion after dynamic avoidance and coordination at time $t = t_5$.

As t ranges from t_1 to t_3, the robots continuously respond to the moving obstacles as well as their simultaneous interactions rather than replanning suddenly. At time $t = t_1$, R_2 deviates a bit from the diagonal reference path to maintain a safety margin from a moving obstacle.R_1 and R_3 continue after earlier unidentified complications. These modifications stay small and close to their original global trajectories. At time $t = t_2$, the robots will have entered the central interaction region, where beyond-obstacle avoidance coordination is also required. R_3 is closest to its target, with R_1 coming next, and then R_2. The effective priority order will be $R_3 > R_1 > R_2$. This allows the car R_3 to go first and the other cars slow down or stop for a short while to avoid a detour. At this time, when R_3 moves away from the immediate conflict, the interaction occurs between R_1 and R_2. An update of priority to $R_1 > R_2$ naturally occurs. As a result, R_1 moves to its destination, while R_2 stops for a bit and then moves.

Unlike a fixed priority rule, this adaptive reassignment changes with the local interaction state and therefore better reflects the evolving geometry of the scene. Between t_4 and t_5, the main avoidance and coordination decisions have already been completed, and the robot motions become goal-directed again. At time t_4, R_1 is close to its target, R_3 is approaching the final corridor, and R_2 has crossed the central region and can move without further interference. By time t_5, all three robots reach their respective targets without collision. The distance-based priority rule prevents deadlock in the shared region while preserving the original mission intent of the nominal static paths. These figures therefore serve as qualitative illustrations of how the extended planning framework may behave in a dynamic multi-robot setting.

6. Conclusions

This paper proposes a turn-aware, grid-based mobile robot path planning framework based on a multi-objective grey wolf optimizer. The proposed method optimizes the path length and the maneuver burden concurrently. Further, the 8-connected motion model is adopted along with the use of Euclidean-consistent costs and strict constraints of feasibility. The experimental analysis is organized in three levels. The results show that the proposed method consistently achieves favorable path-quality trade-offs across a wide range of environment conditions. In the structured 60-map benchmark, it achieved the lowest mean path length and the lowest mean turn count for the grid sizes, densities, and morphologies tested. Run-level summaries show that the variability of the method is relatively low across problem-parameter groups. Qualitative overlays indicate that its smooth trajectories are preserved from the smaller benchmark instances to the large and highly constrained maps. The sequential visualization of dynamic obstacle avoidance and multi-robot coordination further indicates that a reduced heading-change burden supports coherent motion when local interactions evolve over time. The research highlights that one can enhance the maneuvering efficiency of a mobile robot with the aid of multi-objective optimization. Furthermore, the outcomes predict that the newly optimized path would outperform existing state-of-the-art planned paths. Future work may extend the framework to richer motion models with kinematic and dynamic constraints, and to unified formulations that also optimize multi-robot coordination within a single decision process.

Author contributions

Tawfik Guesmi: Conceptualization, validation, resources, writing–review and editing, supervision, project administration, funding acquisition; Nandhini Mahadevan: Software, formal analysis, data curation, writing–review and editing; Dinesh Karunanidy: Conceptualization, methodology, software, formal analysis, investigation, writing–original draft preparation, writing–review and editing; Khalid

Alqunun: Validation, resources, writing–review and editing, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony
A*	A-Star Search
ASMO	Ageist Spider Monkey Optimization
CS	Cuckoo Search
CV	Coefficient of Variation
DE	Differential Evolution
DWA	Dynamic Window Approach
EA	Evolutionary Algorithm
FA	Firefly Algorithm
GWO	Grey Wolf Optimization
IA*	Improved A-Star Algorithm
IGA	Improved Genetic Algorithm
INN–DWA	Improved Neural Network–Dynamic Window Approach
IQR	Interquartile Range
IPSO	Improved Particle Swarm Optimization
MA	Monkey Algorithm
MOGWO	Multi-Objective Grey Wolf Optimizer
MOP	Multi-Objective Optimization Problem
MRPP	Mobile Robot Path Planning
NIOAs	Nature-Inspired Optimization Algorithms
OG	Occupancy Grid
PL	Path Length
PSO	Particle Swarm Optimization
SE	Standard Error
SI	Swarm Intelligence
SMO	Spider Monkey Optimization
TC	Turn Count

References

1. X. Bian, W. Q. Zhao, L. Tang, H. Zhao, X. S. Mei, FastSLAM-MO-PSO: A robust method for simultaneous localization and mapping in mobile robots navigating unknown environments, *Appl. Sci.*, **14** (2024), 10268. https://doi.org/10.3390/app142210268
2. Z. W. Guo, Y. F. Yang, S. L. Dai, *A fusion algorithm integrating hybrid a-star and dynamic window approach for enhanced mobile robot path planning*, In: 2025 44th Chinese Control Conference (CCC), 2025, 1–6. https://doi.org/10.23919/CCC64809.2025.11179573
3. N. Baras, M. Dasygenis, Ugv coverage path planning: An energy-efficient approach through turn reduction, *Electronics*, **12** (2023), 2959. https://doi.org/10.3390/electronics12132959
4. S. Bochkarev, S. L. Smith, *On minimizing turns in robot coverage path planning*, In: 2016 IEEE International Conference on Automation Science and Engineering (CASE), 2016, 1237–1242. https://doi.org/10.1109/COASE.2016.7743548
5. K. Karur, N. Sharma, C. Dharmatti, J. E. Siegel, A survey of path planning algorithms for mobile robots, *Vehicles*, **3** (2021), 448–468. https://doi.org/10.3390/vehicles3030027
6. F. Z. Gul, I. Mir, L. Abualigah, P. Sumari, A. Forestiero, A consolidated review of path planning and optimization techniques: Technical perspectives and future directions, *Electronics*, **10** (2021), 2250. https://doi.org/10.3390/electronics10182250
7. T. Xue, L. Li, L. Shuang, Z. P. Du, P, Ming, Path planning of mobile robot based on improved ant colony algorithm for logistics, *Math. Biosci. Eng.*, **18** (2021), 3034–3045. https://www.aimspress.com/article/doi/10.3934/mbe.2021152
8. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
9. L. K. Hu, Z. Kong, Multi-target point path planning algorithm for mobile robot based on probabilistic roadmap, *Intel. Serv. Robot.*, **19** (2026), 9. https://doi.org/10.1007/s11370-025-00670-6
10. M. G. Tan, S. Z. Wang, B. Huang, Z. B. Yang, R. F. Chen, X .Y. Shen, et al., Beyond visibility limits: A DRL-based navigation strategy for unexpected obstacles, *Unmanned Syst.*, 2026, 1–13.
11. X. W. Zhou, J. W. Yan, M. Yan, K. H. Mao, R. Z. Yang, W. Y. Liu, Path planning of rail-mounted logistics robots based on the improved dijkstra algorithm, *Appl. Sci.*, **13** (2023), 9955. https://doi.org/10.3390/app13179955
12. L. J. Zhu, P. Duan, L. L. Meng, X. H. Yang, Gao-rrt*: A path planning algorithm for mobile robot with low path cost and fast convergence, *AIMS Math.*, **9** (2024), 12011–12042. https://doi.org/10.3934/math.2024587
13. H. Y. Zhang, W. M. Lin, A. X. Chen, Path planning for the mobile robot: A review, *Symmetry*, **10** (2018), 450. https://doi.org/10.3390/sym10100450
14. Y. F. Wang, X. R. Zhao, G. Chen, X. Y. Gao, K. C. Chen, Improved PRM algorithm based on dynamic partitioning and adaptive sampling, *Biomim. Intell. Robot.*, **6**, (2026), 100283. https://doi.org/10.1016/j.birob.2026.100283

15. I. Thammachantuek, M. Ketcham, Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization, *PLoS ONE*, **17** (2022), e0271924. https://doi.org/10.1371/journal.pone.0271924

16. Z. A. Yu, P. Duan, L. L. Meng, Y. Y. Han, Y. Fan, Multi-objective path planning for mobile robot with an improved artificial bee colony algorithm, *Math. Biosci. Eng.*, **20** (2023), 2501–2529. https://doi.org/10.3934/mbe.2023117

17. S. H. Yin, J. B. Hu, Z. R. Xiang, Multi-objective collaborative path planning for multiple water-air unmanned vehicles in cramped environments, *Expert Syst. Appl.*, **292** (2025), 128625. https://doi.org/10.1016/j.eswa.2025.128625

18. L. Sheng, S. Wu, Z. Y. Lv, Modified grey wolf optimizer and application in parameter optimization of pi controller, *Appl. Sci.*, **15** (2025), 4530. https://doi.org/10.3390/app15084530

19. G. Cheng, Y. D. Liu, Hybrid optimization algorithm for solving path planning problems based on grey wolf optimization algorithm, *Sci. Rep.*, **16** (2026), 8479. https://doi.org/10.1038/s41598-026-35037-z

20. Z. Zhang, H. Yang, X. Bai, S. Zhang, C. Xu, The path planning of mobile robots based on an improved genetic algorithm, *Appl. Sci.*, **15** (2025), 3700. https://doi.org/10.3390/app15073700

21. Y. Teng, T. Feng, J. Li, S. Chen, X. Tang, A dual-layer symmetric multi-robot path planning system based on an improved neural network-DWA algorithm, *Symmetry*, **17** (2025), 85. https://doi.org/10.3390/sym17010085

22. Q. Yuan, R. Sun, X. Du., Path planning of mobile robots based on an improved particle swarm optimization algorithm, *Processes*, **11** (2023), 26. https://doi.org/10.3390/pr11010026

23. Y. Han, C. Li, Z. An, Based on the integration of the improved A* algorithm with the dynamic window approach for multi-robot path planning, *Appl. Sci.*, **15** (2025), 406. https://doi.org/10.3390/app15010406