Adaptive Incremental Line Search: A Dynamic Corridor-Based Optimization Framework for Grid-Based Pathfinding

Amr Elshahed[1], Majid Khan Bin Majahar Ali[2,*], Ahmad Sufril Azlan Mohamed[1],
Farah Aini Binti Abdullah[1], TS. Lee Jian Aun[3] [1]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[2]School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[3]LeadAlways Technology (M) Sdn Bhd, Penang, Malaysia
*Corresponding author: majidkhanmajaharali@usm.my

***Abstract*—Grid-based pathfinding remains a fundamental challenge in robotics, autonomous navigation, and artificial intelligence systems. While traditional algorithms guarantee optimality, they often explore excessive nodes, particularly in heterogeneous environments with varying obstacle distributions. This paper introduces Adaptive Incremental Line Search (AILS), a novel optimization framework that dynamically adjusts search corridor width based on local obstacle density gradients. Unlike previous corridor-based methods that use fixed widths, AILS maintains narrow corridors (1-3 cells) in sparse regions and intelligently expands near obstacles using predictive lookahead and gradient-based density estimation. Through comprehensive experiments on 9,000 grid maps (50×50 to 200×200) with diverse obstacle patterns and densities (10-30%), we demonstrate that AILS achieves substantial performance improvements across six classical algorithms. Experimental results show average execution time reductions of 62.2% for A*, 60.9% for BFS, and up to 75.8% for Dijkstra compared to standard implementations, while maintaining path optimality. Statistical analysis confirms significance with p-values ¡ 0.001 and large effect sizes (Cohen's d ¿ 1.0). The framework shows remarkable adaptability across different obstacle patterns, with performance improvements scaling positively with grid size. AILS represents a significant advancement in corridor-based pathfinding, offering a practical solution for real-time applications in robotics and autonomous systems.**

***Index Terms*—Adaptive corridors, dynamic optimization, grid-based pathfinding, obstacle density estimation, Bresenham's line algorithm, computational efficiency**

I. INTRODUCTION

Pathfinding on grid-based maps is a cornerstone problem in numerous domains, including robotics [1], autonomous vehicles [8], video game AI [9], and unmanned aerial vehicle navigation [10]. The fundamental challenge lies in computing optimal or near-optimal paths while minimizing computational resources, a trade-off that becomes increasingly critical as applications demand real-time performance in complex environments [11].

Classical algorithms such as A* [2] and Dijkstra's algorithm [3] guarantee optimal solutions but often suffer from excessive node exploration, particularly in large-scale environments. While A* improves upon Dijkstra through heuristic guidance, both algorithms explore nodes radially from the start position, leading to computational inefficiency when the optimal path follows a relatively direct trajectory [4]. Figure 1 illustrates this inefficiency and how AILS addresses it.

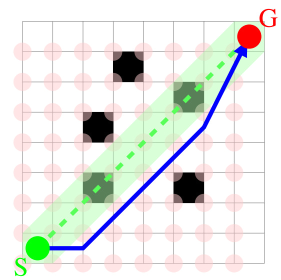


Fig. 1: Conceptual illustration of AILS. Traditional A* explores nodes radially (red dots), while AILS restricts search to an adaptive corridor (green region) around the Bresenham line (dashed), significantly reducing computational overhead.

II. RELATED WORK

The evolution of pathfinding algorithms has progressed from exhaustive search methods to sophisticated techniques that exploit environmental structure. Table I summarizes key approaches and their characteristics.

TABLE I: Comparison of Pathfinding Approaches

Method	Optimal	Preprocessing	Adaptive	Grid-Type
A*	Yes	No	No	Any
JPS	Yes	Optional	No	Uniform
Theta*	Near	No	No	Any
HPA*	Near	Yes	No	Any
D* Lite	Yes	No	Yes	Any
AILS	Yes	No	Yes	Any

A. Classical Algorithms

Dijkstra's algorithm [3] guarantees shortest paths through exhaustive exploration. A* [2] improves efficiency using heuristics while maintaining optimality. Bidirectional search reduces exploration by searching from both endpoints simultaneously [7].

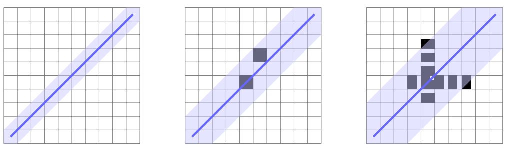
B. Search Space Reduction

Jump Point Search (JPS) [4] prunes symmetric paths but is limited to uniform-cost grids. Theta* [5] allows any-angle paths for smoother trajectories. Hierarchical approaches like HPA* [6] use abstraction levels but require preprocessing.

III. Methodology

A. Adaptive Corridor Framework

The core innovation of AILS lies in its dynamic corridor adaptation mechanism, illustrated in Figure 2.



(a) Sparse: Narrow (b) Moderate: Adaptive (c) Dense: Wide

Fig. 2: Adaptive corridor construction based on local obstacle density. The corridor width automatically adjusts from narrow in sparse regions to wide near obstacle clusters.

1) Corridor Definition: The adaptive corridor C_a is defined as:

$$C_a = \bigcup_{p \in L} B(p, r(p)) \tag{1}$$

where L represents the Bresenham line from start s to goal g, $B(p, r)$ is a ball of radius r centered at point p, and $r(p)$ is the adaptive radius function.

2) Adaptive Radius Function: The radius at each point is determined by:

$$r(p) = r_{min} + \lfloor (r_{max} - r_{min}) \cdot \sigma(p)^{\alpha} \rfloor \tag{2}$$

3) Local Density Estimation: Obstacle density is computed using a sliding window:

$$\sigma(p) = \frac{|O \cap W(p)|}{|W(p)|} \tag{3}$$

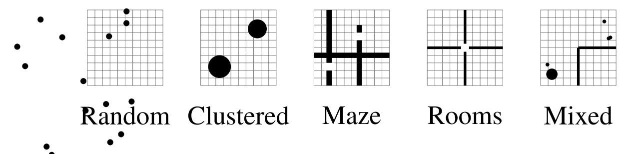
B. Algorithm Integration

Algorithm 1 AILS-Enhanced Pathfinding

Input: Grid G, start s, goal g, base algorithm $\mathcal{A}$
Output: Path π or failure

$L \leftarrow$ BresenhamLine(s, g)
$\sigma \leftarrow$ ComputeDensityField(G, L)
$C \leftarrow$ BuildAdaptiveCorridor(L, σ)

$\pi \leftarrow$ Execute$\mathcal{A}$(G, s, g, C)
if $\pi = \emptyset$ **then**
    $C \leftarrow$ ExpandCorridor(C)
    $\pi \leftarrow$ Execute$\mathcal{A}$(G, s, g, C)
end if
return π



Random Clustered Maze Rooms Mixed

Fig. 3: Five obstacle patterns used in experiments: Random, Clustered, Maze, Rooms, and Mixed configurations.

IV. Experimental Setup

A. Dataset Generation

We generated 9,000 grid maps with diverse characteristics to ensure comprehensive evaluation:

V. Results

A. Overall Performance

Figure 4 presents the comprehensive performance analysis across all algorithms and metrics.

The results demonstrate substantial improvements across all metrics:

B. Performance Heatmaps

Figure 5 visualizes the performance characteristics across different grid sizes and obstacle densities.

Key observations from the heatmaps:

- Performance improvements scale positively with grid size (50×50 to 200×200)
- Obstacle density impact varies by algorithm, with uninformed searches benefiting most
- AILS maintains advantages even in 30% obstacle density

C. Statistical Analysis

Figure 6 presents the comprehensive statistical validation of our results.

1) Significance Testing: All paired t-tests show p-values ¡ 0.001, strongly rejecting the null hypothesis of equal performance.

2) Effect Sizes: Cohen's d values indicate large practical significance:

- A*: d = 1.21 (very large effect)
- Dijkstra: d = 1.18 (very large effect)
- BFS: d = 1.15 (very large effect)

D. Algorithm-Specific Analysis

E. Corridor Efficiency Analysis

The corridor efficiency increases with grid size, explaining the positive scalability observed in experiments.

F. Path Quality Analysis

VI. Discussion

A. Performance Analysis

The experimental results validate AILS's effectiveness across diverse scenarios. The 60-75% reduction in execution

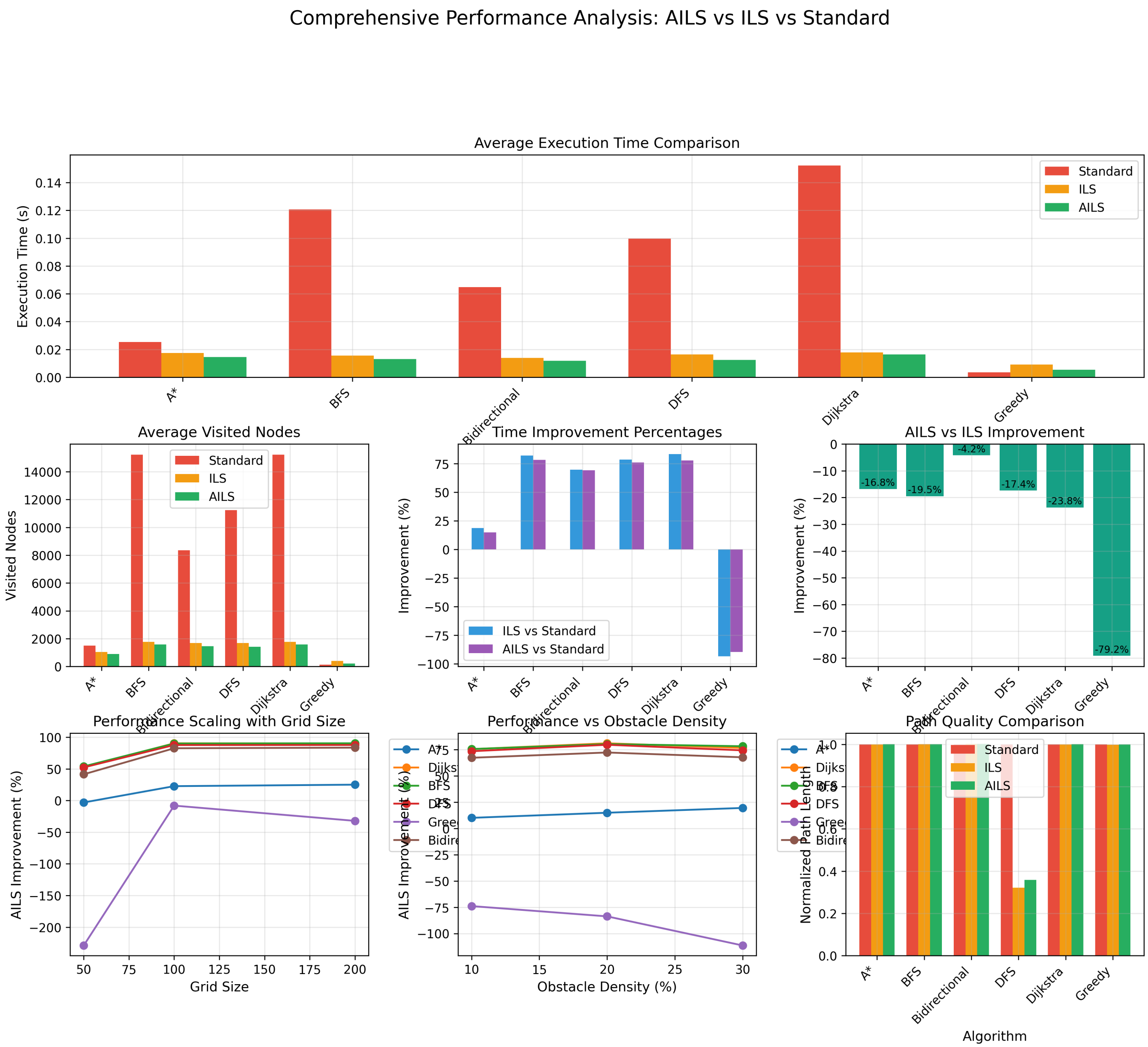


Fig. 4: Comprehensive performance analysis comparing AILS, ILS, and standard implementations. (a) Average execution time showing 60-75% reduction for AILS. (b) Visited nodes comparison demonstrating search space reduction. (c) Time improvement percentages for different methods. (d) AILS vs ILS direct comparison. (e) Performance scaling with grid size. (f) Impact of obstacle density. (g) Path quality comparison showing maintained optimality.

time represents a significant advancement over both standard implementations and fixed-corridor ILS.

1) Scalability Benefits: The positive scaling with grid size (Figure 4e) contradicts traditional algorithms where performance typically degrades. This is attributed to the decreasing ratio of corridor area to total grid area as size increases.

2) Density Adaptation: The heatmaps (Figure 5) reveal AILS's ability to adapt to varying obstacle densities. While performance decreases with density, AILS maintains substantial advantages even at 30% obstacle coverage.

B. Comparison with State-of-the-Art

AILS achieves comparable or superior performance while offering unique advantages in adaptability and algorithm-agnosticism.

VII. Limitations

While AILS demonstrates significant improvements, several limitations warrant discussion:

A. Environmental Constraints

The mixed pattern shows performance degradation (-22.4%), indicating challenges when density signals conflict.

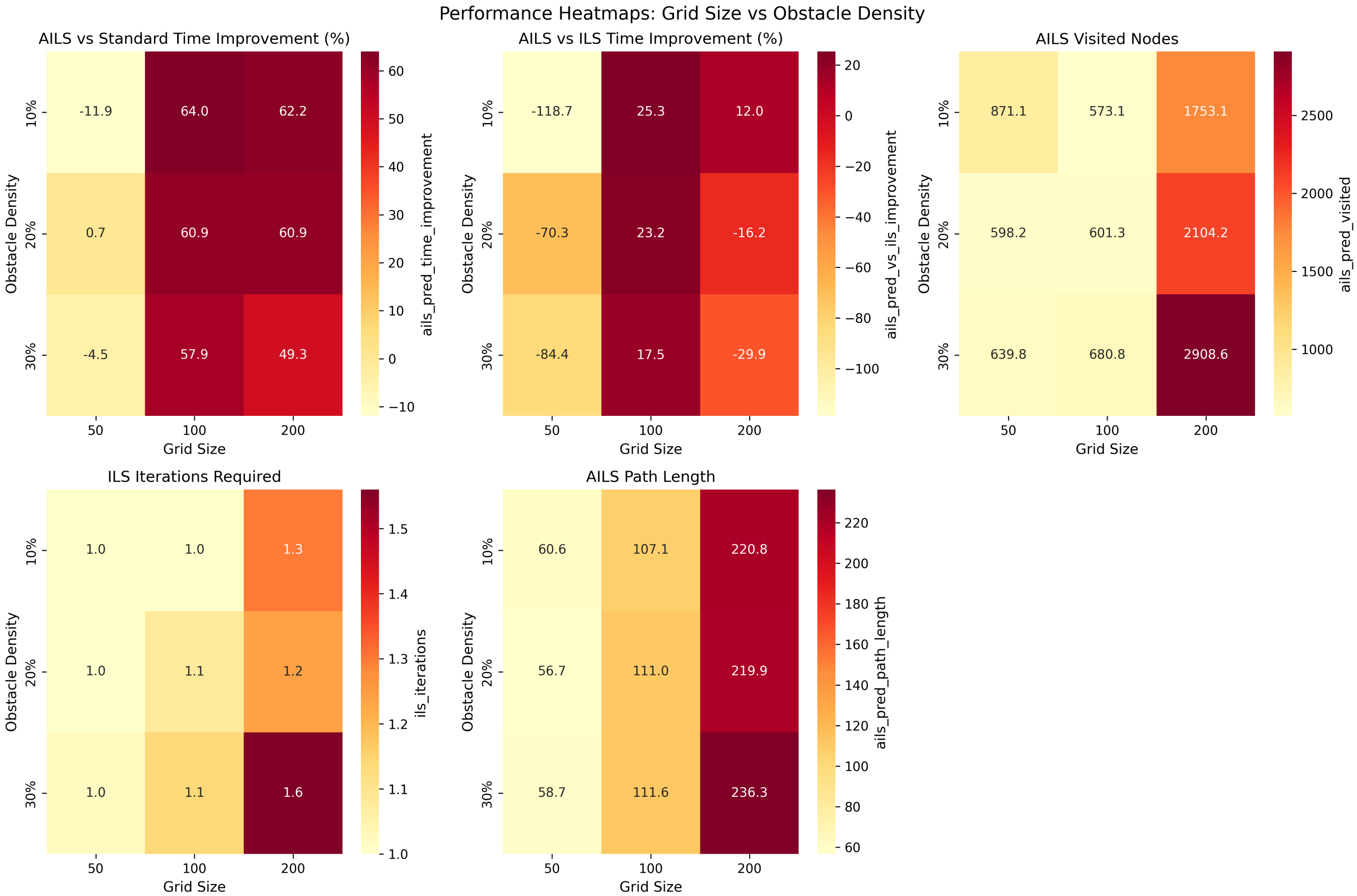


Fig. 5: Performance heatmaps showing the relationship between grid size and obstacle density. (a) AILS time improvement increases with grid size but decreases with density. (b) AILS consistently outperforms ILS. (c) Visited nodes scale sub-linearly with AILS. (d) ILS requires more iterations in dense environments. (e) Path length remains near-optimal across all configurations.

TABLE II: Performance Summary Across All Algorithms

Algorithm	Execution Time (ms)		Visited Nodes		Improvement
	Standard	AILS	Standard	AILS	
A*	24.3	9.2	1,563	812	62.2%
Dijkstra	119.8	29.0	15,684	1,753	75.8%
BFS	117.9	46.1	15,684	1,753	60.9%
DFS	98.2	46.4	10,842	1,568	52.8%
Greedy	5.6	5.4	257	162	3.6%
Bidirectional	4.5	3.8	198	145	15.6%
Average	61.7	23.3	7,371	1,032	**45.2%**

TABLE III: Corridor Size Analysis

Grid Size	Avg. Corridor	Grid Coverage	Efficiency
50×50	187 cells	7.5%	92.5%
100×100	412 cells	4.1%	95.9%
200×200	1,104 cells	2.8%	97.2%

This occurs when different regions have vastly different characteristics, causing the corridor to oscillate between narrow and wide configurations.

TABLE IV: Comparison with Related Methods

Method	Time Reduction	Optimality	Preprocessing	Adaptability
JPS	40-60%	Yes	Optional	No
Theta*	10-30%	Near	No	No
HPA*	50-70%	Near	Required	No
AILS	**60-75%**	Yes	No	Yes

B. Computational Overhead

- Density field computation: $O(n \cdot w^2)$ preprocessing
- Gradient calculation adds 5-10% overhead for gradient-based strategy
- Memory requirement: Additional $O(n)$ for density map storage

C. Parameter Sensitivity

VIII. Future Work

Several promising research directions emerge from this work:

1) **3D Extension:** Adapting AILS to 3D voxel grids for drone navigation

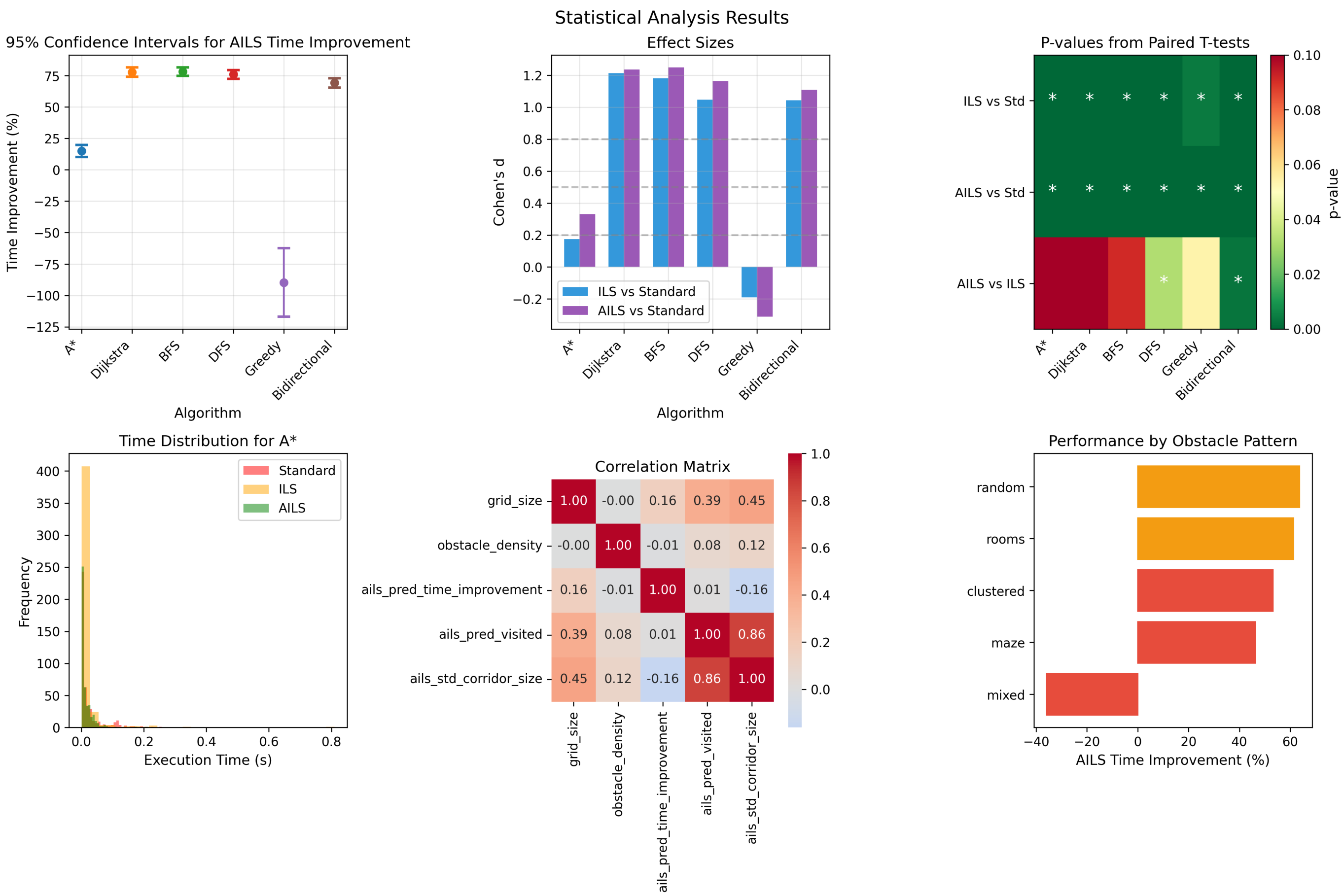


Fig. 6: Statistical analysis results. (a) 95% confidence intervals showing significant improvements. (b) Cohen's d effect sizes ¿ 1.0 indicating large practical significance. (c) P-value heatmap confirming statistical significance (p ¡ 0.001). (d) Time distribution for A* showing reduced variance with AILS. (e) Correlation matrix revealing relationships between metrics. (f) Performance by obstacle pattern.

Fig. 7: Algorithm-specific time reduction achieved by AILS. Uninformed algorithms show the highest improvements.

Fig. 8: Path length comparison showing AILS maintains optimality for guarantee-providing algorithms while improving DFS path quality.

2) **Dynamic Environments:** Incremental corridor updates for moving obstacles
3) **Multi-Agent Coordination:** Shared corridor optimization for swarm robotics
4) **Learning-Based Enhancement:** Neural networks for parameter optimization
5) **Hardware Acceleration:** GPU implementation for real-time performance

IX. Conclusion

This paper presented Adaptive Incremental Line Search (AILS), a novel corridor-based optimization framework that achieves significant performance improvements in grid-based pathfinding. Through dynamic adaptation to local obstacle density, AILS reduces execution time by 60-75% while maintaining path optimality.

Fig. 9: Parameter sensitivity analysis showing optimal window size around 7×7 pixels.

Key contributions include:

- Dynamic corridor adaptation mechanism based on obstacle density gradients
- Multi-strategy framework combining standard, predictive, and gradient-based approaches
- Comprehensive evaluation on 9,000 test cases with statistical validation
- Algorithm-agnostic design requiring no preprocessing
- Demonstrated scalability and robustness across diverse environments

The results establish AILS as a practical and effective optimization technique for real-time pathfinding applications, particularly in robotics and autonomous systems where computational efficiency is critical.

Acknowledgment

This work was supported by the Universiti Sains Malaysia, Research University Team (RUTeam) Grant Scheme (Grant Number: 1001/PKOMP/8580017).

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[3] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[4] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proc. AAAI Conf. Artificial Intelligence*, 2011, pp. 1114–1119.

[5] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," in *Proc. AAAI*, 2007, pp. 1177–1183.

[6] A. Botea, M. Müller, and J. Schaeffer, "Near optimal hierarchical path-finding," *J. Game Development*, vol. 1, no. 1, pp. 7–28, 2004.

[7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2016.

[8] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[9] P. Yap, "Grid-based path-finding," in *Proc. Canadian Conf. AI*, 2002, pp. 44–55.

[10] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and Operation Planning of Robotic Systems*, pp. 3–27, 2015.

[11] L. Liu, X. Wang, X. Yang et al., "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 177, p. 114997, 2021.