Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence
Canadian AI 2026
25–29 May 2026, Vancouver, BC

ASHC: Quantum-Inspired Hierarchical Clustering for Priority-Aware Coverage Path Planning

Garik Kazanjian and Venkat Margapuri
Department of Computing Sciences
Villanova University
Villanova, PA

Abstract

C overage Path Planning, Quantum Walks, Quantum-Inspired Path Planning, Hierarchical Decomposition, Scalable Path Planning

1. Introduction

Coverage Path Planning (CPP) is a fundamental problem in robotics where the goal is to efficiently traverse an environment to ensure full area coverage, with applications in precision agriculture [1], industrial inspection [2], search-and-rescue [3], and cleaning [4]. Traditional CPP approaches rely on classical heuristics [5] [6] [7], grid decomposition [8] [9], and combinatorial optimization [10] [11], but often overlook environment-specific contextual signals such as spatial heterogeneity or task-driven importance. Scalability remains a further challenge, as the search space grows exponentially with the number of visitation targets.

This work introduces Amplitude-Stratified Hierarchical Clustering (ASHC), a quantum-inspired amplitude-driven hierarchical CPP technique. ASHC leverages quantum amplitude fields to encode contextual variation and group targets into semantically meaningful clusters, decomposing the CPP problem into intra-cluster and inter-cluster subproblems that substantially reduce the combinatorial search space. Key contributions include: (i) a quantum-inspired amplitude-based spatial encoding enabling principled cluster decomposition; (ii) a hierarchical clustering mechanism that reduces search space complexity; and (iii) rigorous theoretical formulations and experimental validation across varied environments.

2. Related Work

Classical CPP: Classical approaches rely on TSP variants and cellular decomposition techniques including trapezoidal [12], voronoi [13], and boustrophedon [14] decomposition, as well as graph-based methods [15–18]. These techniques assume uniform significance across coverage areas and face scalability challenges in complex environments.

Context-Aware and Hierarchical CPP: Tang, Mao, and Ma [19] and Huang et al. [20] implemented priority-based multi-robot CPP, though scalability degrades as prioritized targets increase. Cao et al. [21] and Shen et al. [22] addressed scalability through hierarchical frameworks that decompose the environment into subspaces for multi-resolution planning, motivating the hierarchical context-aware approach taken here.

Quantum-Inspired Planning: Reitzner et al. [23] demonstrated quantum walks can outperform classical pathfinding in specific graph structures. Rao et al. [24] showed runtime improvements for multi-robot CPP using quantum heuristics, and Lathrop et al. [25] demonstrated quadratic speedup potential in sampling-based motion planning via Quantum Amplitude Amplification.

* Emails: gkazanji, vmargapu@villanova.edu

3. Problem Setup and Formulation

The environment is modeled as a finite 2D grid $\mathcal{G} = \{(i,j) \mid i,j \in \{1,\dots,N\}\} \subset \mathbb{N}^2$ with obstacle set $\mathcal{O} \subset \mathcal{G}$, target set $\mathcal{V} \subset \mathcal{G}$ satisfying $\mathcal{O} \cap \mathcal{V} = \emptyset$, and free space $\mathcal{F} = \mathcal{G} \setminus \mathcal{O}$ whose induced subgraph is connected. Targets may be partitioned into m disjoint clusters $\{C_1,\dots,C_m\}$ with $\mathcal{V} = \bigcup_{k=1}^{m} C_k$ and $|C_k| = n_k$.

Each target $v_i \in \mathcal{V}$ carries a scalar context value $p(v_i) \in [0,1]$. In the multi-robot setting with R robots, $\mathcal{V}$ is partitioned into R disjoint sub-regions $\mathcal{V}^{(1)},\dots,\mathcal{V}^{(R)}$ with $\mathcal{V}^{(r)} \cap \mathcal{V}^{(s)} = \emptyset \ \forall r \neq s$.

4. Solution Approach

4.1. Amplitude Field for Spatial Context Encoding

An amplitude field is constructed via a discrete-time coined quantum walk [26] [27] simulated classically using Qiskit's statevector simulator [28] [29], over a Hilbert space $\mathcal{H} = \mathcal{H}_{pos} \otimes \mathcal{H}_{coin}$ where $\mathcal{H}_{coin}$ is 4-dimensional representing directions $\{\uparrow,\downarrow,\leftarrow,\rightarrow\}$. The walk is initialized uniformly over free-space cells and evolved for $T = 2N$ steps using a Grover coin with reflective boundary conditions. A time-averaged marginal distribution over the final w steps mitigates oscillations, and the result is min-max normalized producing the amplitude field:

$$\mathcal{A}(x,y) = \frac{\bar{P}_T(x,y) - \min_{(i,j)\in\mathcal{F}} \bar{P}_T(i,j)}{\max_{(i,j)\in\mathcal{F}} \bar{P}_T(i,j) - \min_{(i,j)\in\mathcal{F}} \bar{P}_T(i,j)}, \tag{4.1}$$

where $\bar{P}_T(x,y) = \frac{1}{w}\sum_{t=T-w+1}^{T}\sum_c |\psi_t(x,y,c)|^2$. Each target v_i is assigned contextual value $\mathcal{A}(v_i) \in [0,1]$. Given ordered thresholds $0 < \theta_1 < \cdots < \theta_{m-1} < 1$, targets are partitioned into m disjoint clusters:

$$C_k = \{v_i \in \mathcal{V} \mid \theta_{k-1} \leq \mathcal{A}(v_i) < \theta_k\}, \quad k = 1,\dots,m, \tag{4.2}$$

with $\theta_0 = 0$, $\theta_m = 1$. This amplitude-based partitioning decomposes the otherwise $|\mathcal{V}|!$-complexity TSP into **intra-cluster ordering** (a restricted TSP per cluster C_k) and **inter-cluster ordering** (a permutation over the m cluster indices). All inter-cluster permutations are evaluated to find the most cost-efficient sequence.

4.2. Cost Function

Let σ_i denote the visiting sequence over cluster C_i, with collision-free path computed by A* search [30] [31]. The total coverage cost is:

$$\mathcal{J}(\Sigma) = \sum_{i=1}^{m} \mathcal{J}(\sigma_i), \quad \mathcal{J}(\sigma_i) = \alpha \cdot L(\sigma_i) + \beta \cdot TC(\sigma_i) + \gamma \cdot TT(\sigma_i), \tag{4.3}$$

where $\alpha,\beta,\gamma \in \mathbb{R}_{\geq 0}$, $L(\sigma_i) = \sum_{j=1}^{|\sigma_i|-1} \|p_{j+1} - p_j\|_2$ is total path length, $TC(\sigma_i) = \sum_{j=2}^{|\sigma_i|-1} |\theta_j - \theta_{j-1}|$ is total turning cost, and $TT(\sigma_i) = L(\sigma_i)/\mathrm{v} + TC(\sigma_i)/\omega_{\max}$ is traversal time.

5. Formal Analysis

Let Π_{inter} be the set of all permutations over the m cluster indices, Π_k the intra-cluster permutation space for C_k, and $\Omega_{\text{hier}} \coloneqq \Pi_{\text{inter}} \times \prod_{k=1}^{m} \Pi_k$. The hierarchical search space has size $|\Omega_{\text{hier}}| = m! \cdot \prod_{k=1}^{m} |C_k|!$, yielding a reduction factor over the full TSP of:

$$\mathcal{R} = \frac{|\mathcal{V}|!}{m! \cdot \prod_{k=1}^{m} |C_k|!}, \tag{5.1}$$

which grows exponentially with $|\mathcal{V}|$. The underlying optimization problem is NP-hard, as the classical TSP [32] [33] [34] reduces to it in $O(|\mathcal{V}|)$ time by assigning uniform amplitudes (producing $m = 1$) and setting $\alpha = 1, \beta = \gamma = 0$.

6. Experiments and Results

All experiments use 100 randomized environments per grid size with target density 25% and obstacle density 15%.

6.1. Search Space Reduction

The search space reduction is evaluated across 10×10, 20×20, and 30×30 grid environments. The reduction factor is computed per Equation 5.1. Figure 1 presents the mean $\log_{10}$ reduction with error bars; the dashed line indicates the theoretical expectation under balanced clusters. The proposed technique achieves mean logarithmic reduction factors of 5.04, 25.51, and 56.79 on the three grid sizes respectively. Experimental results fall below the theoretical expectation due to cluster imbalance common in practice: as imbalance increases, factorial growth in the denominator of Equation 5.1 dominates, progressively diminishing the reduction. In the extreme case of high imbalance (one large cluster, two singletons), the reduction reaches zero.

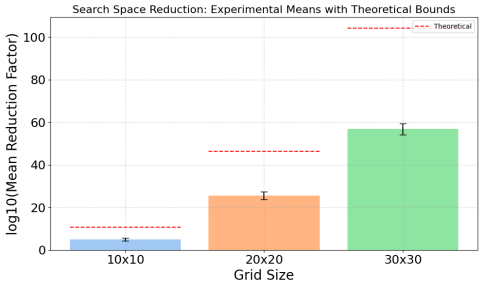

Figure 1. Search space reduction across grid sizes.

6.2. Baseline Comparison and Analysis

ASHC is benchmarked against: **Grid-Split (GS)**, deterministic spatial partitioning; **Random Assignment (RA)**, uniform random target allocation; **KMeans-Priority (KP)** [35] [36], K-means over augmented feature vectors $(x, y, \alpha\mathcal{A})$; **Priority-Global (PG)**, targets globally ordered by descending amplitude with nearest-neighbor traversal [37]; and **Weighted Nearest Neighbor (WNN)** [38], next target selected by minimizing distance-to-priority ratio.

Performance is evaluated using **Priority-Weighted Latency (PWL)**: PWL $= \sum_i w_i t_i$ with $w_i = 1 + \mathcal{A}(x_i, y_i)$ (lower is better), and **Travel Distance** (total Euclidean path length). Experiments use $m \in \{3, 4, 5\}$ clusters. Results are in Table 1.

Single-Robot Performance. PG consistently yields the lowest single-robot PWL by globally ordering all targets by amplitude without structural constraints. ASHC single-robot PWL remains within 2–4% of PG across all grids, indicating that hierarchical partitioning introduces only modest degradation relative to unconstrained global prioritization.

Multi-Robot Performance. Clustering-based methods exhibit substantial PWL reductions under multi-robot execution. On 30×30 with $m = 3$, ASHC reduces PWL from 43326.83 (single) to 13874.43 (multi), outperforming KP (49339.00 to 14408.32). As m increases to 5, ASHC multi-robot PWL further decreases to 7278.32 versus KP's 8518.82.

6.3. Ablation Study

The ablation study isolates the contribution of path length (α), turning cost (β), and traversal time (γ) in the composite objective. Results averaged over 100 randomized environments per grid size are reported in Table 2 for both single- and multi-robot settings.

Grid	Method	$m = 3$		$m = 4$		$m = 5$	
		PWL ↓	Travel ↓	PWL ↓	Travel ↓	PWL ↓	Travel ↓
10×10	KP	615.75 / 166.21	48.25 / 40.39	635.17 / 116.84	50.48 / 36.27	659.87 / 88.66	53.51 / 33.07
	GS	681.95 / –	64.31 / –	728.92 / –	67.82 / –	733.26 / –	72.71 / –
	PG	511.04 / –	44.84 / –	511.04 / –	44.84 / –	511.04 / –	44.84 / –
	RA	900.43 / 278.39	70.50 / 66.80	996.77 / 228.05	78.29 / 70.92	1068.19 / 199.02	83.52 / 72.22
	WNN	531.30 / –	46.09 / –	531.30 / –	46.09 / –	531.30 / –	46.09 / –
	ASHC	434.72 / 118.21	46.74 / 39.83	463.22 / 98.64	49.27 / 34.62	503.88 / 82.14	47.11 / 29.38
20×20	KP	9801.69 / 2810.56	200.12 / 187.55	10009.44 / 2066.22	203.54 / 181.76	10653.06 / 1656.58	212.22 / 180.02
	GS	10223.43 / –	233.00 / –	10385.47 / –	238.53 / –	11009.18 / –	256.85 / –
	PG	8299.71 / –	189.91 / –	8299.71 / –	189.91 / –	8299.71 / –	189.91 / –
	RA	15237.80 / 4621.94	317.20 / 312.16	17842.16 / 3977.03	362.48 / 357.18	19575.61 / 3521.89	395.33 / 389.31
	WNN	8388.93 / –	192.17 / –	8388.93 / –	192.17 / –	8388.93 / –	192.17 / –
	ASHC	8093.24 / 2483.36	198.87 / 188.28	8131.22 / 1878.69	199.73 / 177.27	8341.34 / 1730.24	204.25 / 171.68
30×30	KP	49339.00 / 14408.32	454.05 / 435.39	50331.37 / 10618.60	459.59 / 428.85	51896.41 / 8518.82	474.60 / 427.55
	GS	49733.59 / –	533.99 / –	50887.37 / –	561.02 / –	52736.25 / –	582.11 / –
	PG	42637.72 / –	430.61 / –	42637.72 / –	430.61 / –	42637.72 / –	430.61 / –
	RA	78342.36 / 23515.31	722.79 / 720.70	91178.02 / 20295.98	829.58 / 823.28	101744.00 / 17963.76	922.67 / 913.44
	WNN	42839.29 / –	433.41 / –	42839.29 / –	433.41 / –	42839.29 / –	433.41 / –
	ASHC	43326.83 / 13874.43	446.17 / 442.21	43487.41 / 9213.94	473.22 / 427.74	43712.28 / 7278.32	478.84 / 423.48

Each entry is Single / Multi. "–": method does not produce multi-robot assignments. GS and PG generate a single global tour and cannot be directly extended to multi-robot execution.

Table 1. Single- and multi-robot performance across grid sizes and clustering levels.

Table 2. Mean and Std Dev of cost components across grid sizes.

Grid Size	Config	α	β	γ	Single-Robot Cost		Multi-Robot Cost	
					Mean	Std Dev	Mean	Std Dev
10x10	Baseline	1.00	1.00	1.00	747.89	232.65	708.00	154.68
	No Length	0.00	1.00	1.00	551.25	173.29	519.25	116.13
	No Time	1.00	1.00	0.00	352.59	109.50	334.10	72.72
	No Turns	1.00	0.00	1.00	591.94	182.94	562.65	121.13
20x20	Baseline	1.00	1.00	1.00	5135.70	1844.14	5352.24	1025.40
	No Length	0.00	1.00	1.00	3599.71	1296.08	3748.93	724.15
	No Time	1.00	1.00	0.00	2443.56	877.10	2546.90	487.38
	No Turns	1.00	0.00	1.00	4228.13	1516.09	4408.65	841.02
30x30	Baseline	1.00	1.00	1.00	12881.20	8468.86	15370.09	4851.00
	No Length	0.00	1.00	1.00	8840.41	5817.48	10543.87	3338.75
	No Time	1.00	1.00	0.00	6151.55	4043.90	7340.70	2315.78
	No Turns	1.00	0.00	1.00	10770.44	7078.06	12855.60	4050.77

Traversal time (γ) is the dominant cost component, with its removal reducing single-robot cost by $\sim$52% across all grid sizes, consistent across both robot settings. Path length (α) is the second most influential component, with nullifying it yielding reductions of 26–31% that grow modestly with environment size, reflecting increasing geometric complexity.

7. **Conclusion**

This study presents ASHC, a hierarchical cluster-based CPP approach leveraging quantum-walk-inspired amplitude fields for semantic target decomposition in obstacle-populated 2D grid environments. Formal analysis establishes exponential search space reduction over classical TSP, and experiments confirm robustness across single-robot and multi-robot settings. Future work will extend the framework to dynamic environments, heterogeneous robots, 3D scenarios, and learning-based clustering heuristics.

Acknowledgment

This work is supported by the Pollinator Health: Research and Application no. 2024-67014-42301 from the U.S. Department of Agriculture's National Institute of Food and Agriculture.

References

[1] A. Basiri, V. Mariani, G. Silano, M. Aatif, L. Iannelli, and L. Glielmo. "A survey on the application of path-planning algorithms for multi-rotor UAVs in precision agriculture". In: *Journal of Navigation* 75.2 (2022), 364–383. DOI: 10.1017/S0373463321000825.

[2] G. K. Fischer, M. Bergau, D. A. Gómez-Rosal, A. Wachaja, J. Graeter, M. Odenweller, U. Piechottka, F. Höflinger, N. Gosala, N. Wetzel, et al. "Evaluation of a Smart Mobile Robotic System for Industrial Plant Inspection and Supervision". In: *IEEE Sensors Journal* (2024).

[3] B. Hao, J. Zhao, H. Du, Q. Wang, Q. Yuan, and S. Zhao. "A search and rescue robot search method based on flower pollination algorithm and Q-learning fusion algorithm". In: *PloS one* 18.3 (2023), e0283751.

[4] R. Memmesheimer, M. Overbeck, B. Kral, L. Steffen, S. Behnke, M. Gersch, and A. Roennau. "Cleaning Robots in Public Spaces: A Survey and Proposal for Benchmarking Based on Stakeholders Interviews". In: *Robot World Cup*. Springer, 2024, pp. 373–385.

[5] C. S. Tan, R. Mohd-Mokhtar, and M. R. Arshad. "A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms". In: *Ieee Access* 9 (2021), pp. 119310–119342.

[6] C. Gehring, M. Asai, R. Chitnis, T. Silver, L. Kaelbling, S. Sohrabi, and M. Katz. "Reinforcement learning for classical planning: Viewing heuristics as dense reward generators". In: *Proceedings of the international conference on automated planning and scheduling*. Vol. 32. 2022, pp. 588–596.

[7] P. Haslum, F. Ivankovic, M. Ramirez, D. Gordon, S. Thiébaux, V. Shivashankar, and D. S. Nau. "Extending classical planning with state constraints: Heuristics and search for optimal planning". In: *Journal of Artificial Intelligence Research* 62 (2018), pp. 373–431.

[8] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, A. V. Le, and P. Veerajagadheswar. "Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem". In: *IEEE access* 8 (2020), pp. 225945–225956.

[9] T. Kusnur and M. Likhachev. "Complete, decomposition-free coverage path planning". In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2022, pp. 1431–1437.

[10] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. "Differentially private combinatorial optimization". In: *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2010, pp. 1106–1125.

[11] F. Da Ros, M. Soprano, L. Di Gaspero, and K. Roitero. "Large language models for combinatorial optimization: A systematic review". In: *ACM Computing Surveys* (2025).

[12] A. Abbadi and V. Přenosil. "Safe path planning using cell decomposition approximation". In: *Distance Learning, Simulation and Communication* 8 (2015), pp. 1–6.

[13] M. R. H. Al-Dahhan and K. W. Schmidt. "Voronoi boundary visibility for efficient path planning". In: *Ieee Access* 8 (2020), pp. 134764–134781.

[14] R. Bähnemann, N. Lawrance, J. J. Chung, M. Pantic, R. Siegwart, and J. Nieto. "Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem". In: *Field and Service Robotics: Results of the 12th International Conference*. Springer. 2021, pp. 277–290.

[15] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis. "Graph-based path planning for autonomous robotic exploration in subterranean environments". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 3105–3112.

[16] H. Ma. "Graph-based multi-robot path finding and planning". In: *Current Robotics Reports* 3.3 (2022), pp. 77–84.

[17] D. T. Wooden. "Graph-based path planning for mobile robots". PhD thesis. 2006.

[18] D. Kularatne, S. Bhattacharya, and M. A. Hsieh. "Going with the flow: a graph based approach to optimal path planning in general flows". In: *Autonomous Robots* 42.7 (2018), pp. 1369–1387.

[19] J. Tang, Z. Mao, and H. Ma. "Large-scale multi-robot coverage path planning on grids with path deconfliction". In: *IEEE Transactions on Robotics* (2025).

[20] S.-K. Huang, W.-J. Wang, and C.-H. Sun. "A path planning strategy for multi-robot moving with path-priority order based on a generalized Voronoi diagram". In: *Applied Sciences* 11.20 (2021), p. 9650.

[21] C. Cao, J. Zhang, M. Travers, and H. Choset. "Hierarchical Coverage Path Planning in Complex 3D Environments". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 3206–3212. DOI: 10.1109/ICRA40945.2020.9196575.

[22] Z. Shen, B. Shirose, P. Sriganesh, and M. Travers. "CAP: A Connectivity-Aware Hierarchical Coverage Path Planning Algorithm for Unknown Environments using Coverage Guidance Graph". In: *arXiv preprint arXiv:2503.00647* (2025).

[23] D. Reitzner, M. Hillery, and D. Koch. "Finding paths with quantum walks or quantum walking through a maze". In: *Physical Review A* 96.3 (2017), p. 032323.

[24] P. U. Rao, F. Speelman, B. Sodhi, and S. Kinge. "A Quantum Computing Approach for Multi-robot Coverage Path Planning". In: *arXiv preprint arXiv:2407.08767* (2024).

[25] P. Lathrop, B. Boardman, and S. Martínez. "Quantum search approaches to sampling-based motion planning". In: *IEEE Access* 11 (2023), pp. 89506–89519.

[26] C. Chandrashekar. "Discrete-time quantum walk-dynamics and applications". In: *arXiv preprint arXiv:1001.5326* (2010).

[27] A. M. Childs. "On the relationship between continuous-and discrete-time quantum walk". In: *Communications in Mathematical Physics* 294.2 (2010), pp. 581–603.

[28] A. J. Gangapuram, A. Läuchli, and C. Hempel. "Benchmarking quantum computer simulation software packages: State vector simulators". In: *SciPost Physics Core* 7.4 (2024), p. 075.

[29] A. Li, B. Fang, C. Granade, G. Prawiroatmodjo, B. Heim, M. Roetteler, and S. Krishnamoorthy. "Sv-sim: scalable pgas-based state vector simulation of quantum circuits". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021, pp. 1–14.

[30] T. Lozano-Pérez and M. A. Wesley. "An algorithm for planning collision-free paths among polyhedral obstacles". In: *Communications of the ACM* 22.10 (1979), pp. 560–570.

[31] S. Lee and J. Park. "Neural computation for collision-free path planning". In: *Journal of Intelligent Manufacturing* 2.5 (1991), pp. 315–326.

[32] S. Arora. "Approximation schemes for NP-hard geometric optimization problems: A survey". In: *Mathematical Programming* 97.1 (2003), pp. 43–69.

[33] S. P. Fekete. "Simplicity and hardness of the maximum traveling salesman problem under geometric distances". In: *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*. 1999, pp. 337–345.

[34] M. Hidalgo-Herrero, P. Rabanal, I. Rodriguez, and F. Rubio. "Comparing problem solving strategies for NP-hard optimization problems". In: *Fundamenta Informaticae* 124.1-2 (2013), pp. 1–25.

[35] S. F. Javed, M. E. Khan, Z. Yahya, M. J. Idrisi, and W. Tenna. "Performance analysis of three-dimensional passive micromixers using k-means priority clustering with AHP-based sustainable design optimization". In: *Scientific Reports* 15.1 (2025), p. 18140.

[36] W. Sun, Y. Zou, N. Guan, X. Zhang, J. Liu, and M. H. Farzaneh. "A systematic flexible-window-based scheduling framework for time-sensitive networking". In: *IEEE Internet of Things Journal* 12.11 (2025), pp. 15636–15649.

[37] T. S. Alemayehu and J.-H. Kim. "Efficient nearest neighbor heuristic TSP algorithms for reducing data acquisition latency of UAV relay WSN". In: *Wireless Personal Communications* 95.3 (2017), pp. 3271–3285.

[38] S. Cost and S. Salzberg. "A weighted nearest neighbor algorithm for learning with symbolic features". In: *Machine learning* 10.1 (1993), pp. 57–78.