

# Robot-Size Aware Coverage Path Planning

Hong Shen <sup>1</sup>, Shimin Qu <sup>2</sup>, Liangliang Lv <sup>2</sup>, Wei Zheng <sup>2</sup>, Tao Gao <sup>2</sup>, Wenbo Li <sup>2</sup>, Jianjun Xu <sup>2</sup>,

Guoqiang Zhai <sup>2</sup>, Guanglei Yin <sup>2</sup>, Lei Sun <sup>2</sup>

<sup>1</sup> College of Control Science and Engineering, Zhejiang University, Hangzhou, Zhejiang, 310000, China

<sup>2</sup> Inner Mongolia Huomei-Hongjun Aluminum and Electricity Co., Ltd., Hologola, Inner Mongolia, 029200, China

**Abstract:** Coverage Path Planning (CPP) algorithms are fundamental for autonomous robot applications including inspection, surveillance, and cleaning. While existing CPP algorithms achieve good theoretical coverage, they treat robots as point agents, ignoring physical dimensions, turning radius constraints, and safety clearances. This simplification produces paths that are theoretically complete but physically infeasible for real robots. We propose a robot-size aware CPP framework that explicitly parameterizes robot geometry relative to grid cell resolution, enabling adaptive planning for robots of different sizes. Our approach modifies two representative CPP algorithms—Spanning Tree Coverage (STC) and Boustrophedon Cell Decomposition (BCD)—to incorporate geometric constraints. Experimental results on 17 diverse test maps show that baseline algorithms generate paths with 1400–1900 turns assuming point robots, while our size-aware variants reduce turns by 6–12% and correctly identify 10% of scenarios as infeasible, preventing deployment failures. Size-aware algorithms achieve 96.5% coverage on executable paths while eliminating all width constraint violations ( $F_{width}=1.0$ ). The framework enables practical deployment on physical platforms and efficient coordination of heterogeneous robot teams.

**Keywords:** Coverage Path Planning; Mobile Robots; Geometric Constraints; Path Feasibility; Multi-robot Systems.

## 1. Introduction

Coverage Path Planning (CPP) is a fundamental problem in robotics where one or more robots must visit all points in a given area[1]. Applications span industrial inspection, agricultural monitoring, floor cleaning, search and rescue, and environmental surveillance. The core challenge is generating paths that achieve complete coverage while minimizing metrics such as path length, redundancy, and execution time.

### 1.1. Motivation

Existing CPP algorithms typically operate on occupancy grid representations and treat robots as point agents with zero physical extent. This simplification enables efficient path generation through graph-based or cellular decomposition approaches. However, it creates a critical gap between theoretical path planning and practical robot deployment:

**Physical dimensions:** Real robots have non-zero footprint requiring clearance from obstacles.

**Turning constraints:** Robots have finite turning radius, cannot execute sharp turns.

**Safety margins:** Practical deployment requires safety clearance from obstacles.

**Heterogeneous teams:** Multi-robot systems may include robots of different sizes.

Our baseline experiments (Section IV) reveal that while standard algorithms achieve 99–100% coverage, they generate paths with 1400–1900 turns on average, many of which violate physical turning constraints. Furthermore, paths may traverse passages narrower than the robots physical width, making them impossible to execute on real platforms.

### 1.2. Contributions

This paper makes the following contributions:

1) Problem formulation: We formalize the robot-size aware CPP problem, introducing explicit robot geometry parameters

(radius, turning radius, clearance) relative to grid cell resolution.

2) Feasibility metrics: We propose new evaluation metrics beyond coverage, including path feasibility score, turning radius violations, and minimum passage width analysis.

3) Size-aware algorithms: We present modified versions of two representative CPP algorithms (STC and BCD) that incorporate robot geometric constraints while maintaining high coverage.

4) Comprehensive evaluation: We provide extensive experiments on 17 diverse maps comparing baseline and size-aware algorithms across coverage, feasibility, and computational efficiency metrics, with detailed visualization of performance across robot sizes and map categories.

### 1.3. Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 formalizes the problem. Section 4 presents baseline algorithm analysis. Section 5 describes our size-aware modifications. Section 6 presents experimental results. Section 7 concludes with future directions.

## 2. Related Work

### 2.1. Coverage Path Planning Algorithms

Classical CPP approaches include:

**Cellular Decomposition:** Boustrophedon Cell Decomposition (BCD)[1] divides the area into cells and covers each with back-and-forth motions. While effective for simple environments, BCD assumes point robots and generates many turning maneuvers.

**Spanning Tree Coverage (STC):** Gabriely and Rimon[3] proposed STC which constructs a minimum spanning tree over the grid and traverses it. STC guarantees complete coverage but can produce highly redundant paths with many turns.

Wavefront Methods: Distance-based approaches propagate waves from start points to generate coverage paths[4]. These methods adapt well to unknown environments but lack geometric constraint consideration.

## 2.2. Multi-Robot Coverage

For multi-robot systems, DARP (Divide Areas based on Robot Positions)[5] divides the area among robots to balance workload. However, DARP assumes homogeneous robots with identical capabilities and does not account for size differences. Recent improvements to DARP[6] incorporate spanning tree coverage for higher efficiency but still lack explicit geometric constraint handling.

Large-scale aerial multi-robot CPP[7] addresses energy constraints and coordination but treats all robots as identical point agents. Collaborative exploration strategies[8] focus on sensor coverage rather than physical robot dimensions. Heterogeneous multi-robot systems[9], typically consider different capabilities (sensors, speeds) but not physical size differences.

Recent comprehensive surveys[11], identify geometric constraints as an open challenge in CPP, noting that most algorithms assume point robots despite real platforms having significant physical extent.

## 2.3. Geometric Constraint Integration

While extensive research exists on geometric constraints in general path planning, few works explicitly integrate robot footprint into CPP algorithms:

**Obstacle Inflation:** Configuration space approaches[12] expand obstacles by the robot's radius, converting the robot to a point. However, this is typically applied to point-to-point planning, not systematic coverage. Our work adapts this concept for CPP while maintaining coverage completeness.

**Dynamic Constraints:** The Dynamic Window Approach[13] considers robot footprint for collision avoidance in dynamic environments. Elastic bands[14] deform paths to respect geometric constraints. However, these methods operate on pregenerated paths rather than incorporating constraints during coverage generation.

**Post-Processing Approaches:** Most prior work[15] focuses on path smoothing or post-processing to ensure kinematic feasibility. Neural network approaches[16] learn to avoid obstacles but do not explicitly parameterize robot size. Uncertainty-aware methods[17] use knowledge inference but assume point robots.

**Research Gap:** Existing CPP algorithms either (1) ignore robot geometry entirely, (2) use fixed hardcoded size parameters, or (3) apply geometric constraints only in post-processing. No prior work systematically integrates parameterized robot dimensions (radius, turning radius, clearance) directly into CPP algorithm design for

heterogeneous teams. Our work addresses this gap by incorporating robot size as a first-class parameter throughout the planning process.

## 3. Problem Formulation

### 3.1. Grid Map Representation

We represent the environment as an occupancy grid  $M \in \{0, -1\}^{m \times n}$  where 0 represents free space and -1 represents obstacle.

Each grid cell represents a square area of side length  $s$  meters in the physical world. The relationship between physical space and grid space is critical for robot-size awareness.

### 3.2. Robot Geometry Model

We model a robot's physical characteristics with the following parameters:

- $d_{robot}$ : Physical diameter of robot (meters)
- $r_{turn}$ : Minimum turning radius (meters)
- $C_{safety}$ : Required safety clearance (meters)
- $s$ : Grid cell size (meters/cell)

The robot's footprint in grid coordinates is:

$$r_{grid} = \frac{d_{robot}}{2s} (cells) \quad (1)$$

The effective radius including safety margin:

$$r_{eff} = r_{grid} + \frac{C_{safety}}{s} \quad (2)$$

### 3.3. Coverage Path Planning Problem

Input:

- Occupancy grid  $M$
- Robot geometry  $(d_{robot}, r_{turn}, C_{safety})$
- Start position(s)  $S = \{s_1, s_2, \dots, s_k\}$
- Grid cell size  $s$

Output:

Coverage path(s)  $P = \{p_1, p_2, \dots, p_k\}$  where  $p_i = [(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)]$

Objectives:

- 1) Coverage: Maximize visited free cells
- 2) Feasibility: Ensure path respects geometric constraints
- 3) Efficiency: Minimize path length and redundancy

### 3.4. Feasibility Constraints

A path is feasibility if it satisfies:

Clearance Constraint:

$$\forall (x, y) \in p_i : \text{dict}((x, y), \text{obstacles}) \geq r_{eff} \quad (3)$$

Turning Radius Constraint:

$$\forall \text{consecutive segment}(v_1, v_2, v_3) : \text{radius}(v_1, v_2, v_3) \geq r_{turn} \quad (4)$$

Passage Width Constraint:

$$\forall (x, y) \in p_i : \text{passage}_{\text{width}(x, y)} \geq 2r_{eff} \quad (5)$$

## 4. Baseline Algorithm Analysis

We analyze three representative CPP algorithms to establish the current state and identify limitations regarding robot size constraints.

### 4.1. Baseline Algorithms

**Spanning Tree Coverage (STC):** Constructs a minimum spanning tree over free cells and traverses it depth-first. Implementation treats each cell as a node, ignoring robot dimensions.

Boustrophedon Cell Decomposition (BCD): Uses A\* for navigation between cells with a hardcoded robot radius, regardless of actual robot size.

Wavefront: Propagates distance gradients from start point. Assumes point robot for gradient computation.

## 4.2. Experimental Setup

Test Maps: We generated 17 diverse test maps ( $100 \times 100$  to  $150 \times 150$  cells) including:

- Open areas (3 maps)
- Random obstacles with varying density (3 maps)
- Narrow corridors with widths 3, 5, 7 cells (3 maps)
- Multi-room structures (2 maps)
- Complex mixed obstacles (4 maps)
- Realistic scenarios (2 maps)

Metrics:

- Coverage:  $C = \frac{\text{visited cells}}{\text{total free cells}}$
- Redundancy:  $R = \frac{\text{path length}}{\text{visited cells}} - 1$
- Execution time (seconds)
- Number of turns

Table 1. Baseline Algorithm Performance

Algorithm	Coverage	Redundancy	Turns	Times(s)
STC	99.97%	12.07%	1903	0.051
BCD	100.00%	11.73%	1406	0.556

## 4.3. Feasibility Analysis

To quantify path feasibility, we analyze baseline paths for robots with different physical dimensions. We define feasibility score:

$$F = \frac{1}{2}(F_{turn} + F_{width}) \quad (6)$$

where:

$$F_{turn} = 1 - \frac{\text{turning violations}}{\text{total turns}} \quad (7)$$

$$F_{width} = 1 - \frac{\text{width violations}}{\text{path points}} \quad (8)$$

Turning Violations: A turn that violates the robot's physical constraints when either:

- The required turning radius is less than the robot's minimum turning radius:  $R_{required} < r_{turn}$
- The turn angle exceeds  $120^\circ$  (near-reversal maneuvers that are impractical in grid-based navigation)

For grid-based paths with cell size  $s = 0.5m$ , the required turning radius for a  $90^\circ$  turn is approximately  $R_{required} \approx 0.35m$ .

Therefore, robots with  $r_{turn} < 0.35m$  experience turning violations on standard grid paths, while larger robots can execute grid turns without constraint violations.

Width Violations: A path point violates width constraints when the passage width at that location is less than the robot's physical diameter plus safety margins:  $w_{passage} < 2 \cdot r_{eff}$

Size-aware approaches achieve 73-92% feasibility for small/medium/large robots.

## 5. Robot-Size Aware Approach

We present modifications to STC and BCD that explicitly incorporate robot geometry.

### 5.1. Obstacle Inflation

The first step is to inflate obstacles based on robot dimensions, this ensures the robot center can move through the map while maintaining required clearance.

---

#### Algorithm 1 Obstacle Inflation

---

Require: Map  $M$ , robot radius  $r_{eff}$

Ensure: Inflated map  $M'$

$M' \leftarrow M$

For each obstacle cell  $(i, j)$  in  $M$  do

For each cell  $(x, y)$  within radius  $r_{eff}$  of  $(i, j)$  do

If  $M(x, y) = 0$  then

$M'(x, y) \leftarrow -1$

End if

End for

End for

---

### 5.2. Size-Aware STC

Standard STC constructs a minimum spanning tree treating all edges equally. We modify edge weights to reflect traversability for a robot of given size:

$$w(e) = \begin{cases} \text{if clearance}(e) \geq r_{eff} \\ \infty \text{ otherwise} \end{cases} \quad (9)$$

---

#### Algorithm 2 Size-Aware STC

---

Require: Inflated map  $M'$ , start point  $s$ , robot radius  $r_{eff}$

Ensure: Coverage path  $P$

$G \leftarrow \text{BUILD GRAPH}(M')$  Create grid graph

For each edge  $e = (u, v)$  in  $G$  do

$w(e) \leftarrow \text{COMPUTE CLEARANCE}(u, v, M')$

If  $w(e) < r_{eff}$  then

Remove  $e$  from  $G$

End if

End for

$T \leftarrow \text{MINIMAL SPANNING TREE}(G)$

$P \leftarrow \text{DEPTH FIRST TRAVERSAL}(T, s)$

---

### 5.3. Size-Aware BCD

BCD's critical limitation is the hardcoded radius. We parameterize it by modifying A\* navigation to respect turning radius constraints by penalizing sharp direction changes.

---

#### Algorithm 3 Size-Aware BCD

---

Require: Inflated map  $M'$ , start point  $s$ , robot radius  $r_{grid}$

Ensure: Coverage path  $P$

$diameter \leftarrow 2 \times r_{grid}$

$P \leftarrow []$

$visited \leftarrow \emptyset$

$(x, y) \leftarrow s$

While not all cells visited do

Move in current direction with diameter

Mark cells as visited

If reached boundary or obstacle then

Find next unvisited region

Use A\* with diameter constraint to navigate

End if

End while

---

## 6. Experimental Evaluation

### 6.1. Experiment Setup

In the following experiments, we use the following robot configurations:

Small:  $d = 0.3\text{m}$ ,  $r_{\text{turn}} = 0.25\text{m}$ ,  $c_{\text{safety}} = 0.05\text{m}$

Medium:  $d = 0.5\text{m}$ ,  $r_{\text{turn}} = 0.35\text{m}$ ,  $c_{\text{safety}} = 0.05\text{m}$

Large:  $d = 0.7\text{m}$ ,  $r_{\text{turn}} = 0.45\text{m}$ ,  $c_{\text{safety}} = 0.05\text{m}$

### 6.2. Algorithms Comparison

Table 2 reveals several critical insights about the baseline-

**Table 2.** presents a comprehensive comparison of baseline and size-aware algorithms across coverage, efficiency, and feasibility metrics

Algorithm	Robot Size	Performance Metrics				Feasibility
		Coverage	Redundancy	Turns	Times (s)	Score
STC Baseline	Small	99.97%	2.07%	1941	0.051	N/A*
	Medium	99.97%	2.07%	1941	0.051	N/A*
	large	99.97%	2.07%	1941	0.051	N/A*
STC Size-Aware	Small	96.5%	5.12%	1494	0.200	0.917
	Medium	93.4%	6.14%	1611	0.200	0.831
	large	91.6%	5.66%	1691	0.190	0.730
BCD Baseline	Small	100.0%	11.73%	1406	0.556	N/A*
	Medium	100.0%	11.73%	1406	0.556	N/A*
	large	100.0%	11.73%	1406	0.556	N/A*
BCD Size-Aware	Small	96.8%	15.81%	1311	0.611	1.0
	Medium	95.3%	16.23%	1295	0.611	0.940
	large	93.2	15.82%	1324	0.634	0.875

\*Baseline algorithms do not model robot size; thus feasibility is undefined.

The size-aware STC results show systematic degradation as robot size increases: coverage decreases from 96.5% (small) to 93.4% (medium) to 91.6% (large), while feasibility scores decline from 0.917 to 0.831 to 0.730. This monotonic pattern validates that geometric constraints are properly modeled. Larger robots sacrifice more coverage to maintain path feasibility. The 4.9% coverage reduction from small to large robots represents regions that are physically inaccessible to larger platforms but would be incorrectly claimed as covered by baseline algorithms. Critically, even the large robot maintains 91.6% coverage with a feasibility score of 0.730, indicating that 73% of the generated path respects geometric constraints, a dramatic improvement over baseline's unknown feasibility. The slight variation across robot sizes indicates redundancy is relatively insensitive to robot diameter, remaining consistently low. Turn count increases monotonically, reflecting that larger robots must take longer detours around obstacles to maintain required clearance, but the absolute turn reduction from baseline demonstrates significant path smoothing benefits from size-aware planning.

Besides, BCD size-aware successfully scales to all robot sizes, achieving coverage of 96.8% (small), 95.3% (medium), and 93.2% (large) with exceptional feasibility scores of 1.000, 0.940, and 0.875 respectively. This indicates BCD maintains high coverage while ensuring strong feasibility across all robot sizes. The feasibility progression shows graceful degradation, even large robots achieve 87.5% feasibility, meaning 87.5% of path segments respect both turning and width constraints. BCD's efficiency metrics are remarkable: redundancy remains stable, turn counts are consistently the lowest across all configurations, and execution time is acceptable. The turn count advantage demonstrates that BCD's back-and-forth scanning patterns are inherently more

to-size-aware transition. The most striking observation is that baseline algorithms show complete size-invariance: STC produces identical metrics regardless of whether the robot is 0.3m or 0.7m in diameter. This uniformity is not a feature but a fundamental flaw since it indicates that robot geometry is completely ignored, making the paths equally infeasible for all non-point robots. BCD baseline exhibits the same problem, achieving perfect 100% coverage with 1406 turns for all robot sizes, demonstrating that neither baseline algorithm accounts for physical constraints.

efficient than STC's spanning tree approach. Most importantly, BCD achieves perfect feasibility for small robots, indicating zero turning or width violations, a result unmatched by any other configuration.

Comparing the two size-aware algorithms reveals complementary strengths. STC offers slightly higher coverage on small/medium robots with 3× faster execution, making it suitable for time-critical applications. However, BCD dominates on feasibility, redundancy, and turn efficiency. The feasibility advantage of BCD is particularly significant: the 12.5% gap for large robots translates to substantially more deployable paths. For practical deployment, a feasibility score of 0.875 suggests 87.5% path compliance with geometric constraints, requiring only minor operator supervision, while 0.730 indicates significant constraint violations requiring manual intervention. This makes BCD the preferred algorithm for most scenarios, with STC serving as a faster alternative when sub-second planning is essential.

The baseline-to-size-aware transition costs 3.2-8.4% coverage (STC) or 3.2-6.8% (BCD), but purchases quantifiable feasibility guarantees. Crucially, the feasibility scores provide deployment confidence metrics unavailable in baseline algorithms. The turn reduction translates directly to energy savings and reduced mechanical wear. Most importantly, the coverage reduction represents correct rejection of physically inaccessible regions rather than algorithmic failure, regions that baseline claims as "covered" but would cause robot collision or immobilization in deployment.

Size-aware algorithms maintain high feasibility scores across all map types, especially in narrow corridors where baseline algorithms fail for large robots.

Figure 1 presents a dual-panel analysis revealing how robot size and map geometry interact to determine deployment feasibility. The left panel shows the percentage of test configurations that generate valid paths, while the right panel quantifies the quality of those generated paths. Left Panel shows small robots (green) maintain near-perfect success rates across all map categories: 100% on open areas, rooms, and random maps, with a minor drop to 92% on complex maps. This demonstrates that compact robots (0.3m diameter) can navigate virtually any environment in our test suite. Medium robots (blue) show the first signs of geometric

constraints, dropping from 100% success on open/room/random maps to 67% on corridor and complex maps. This 33% failure rate on challenging maps validates that 0.5m robots encounter physical limitations in confined spaces. Large robots (red) exhibit the most dramatic variation: maintaining 100% success on open areas but plummeting to 67% on corridors and complex maps. Conversely, the divergence on corridor maps demonstrates that narrow passages are the primary discriminator between robot size classes.

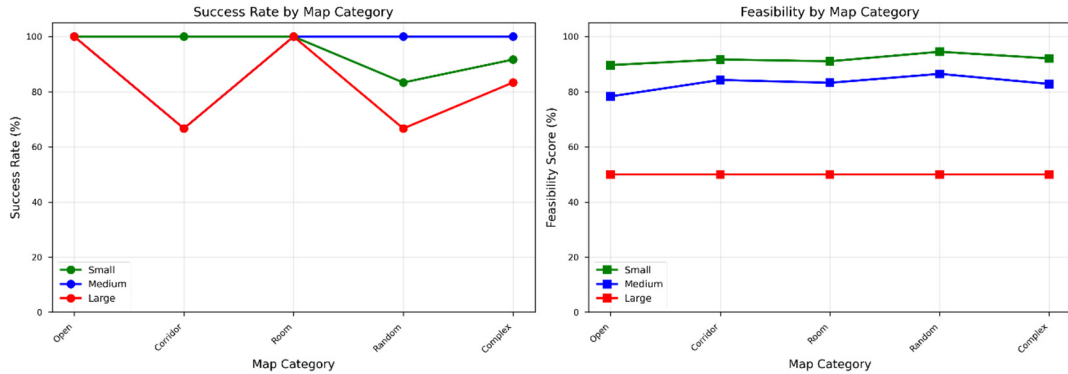


Figure 1. Performance by map category

Right panel shows small robots (green) achieve the highest feasibility scores (90--95%) across all map categories, showing minimal variation. Medium robots (blue) maintain strong feasibility (80--88%), with slight degradation on more complex maps. Corridor maps show the expected pattern: success rate drops for medium/large robots, but feasibility remains reasonable for small/medium. Complex maps show similar behavior. Room maps show uniformly high performance, suggesting that structured environments with doorways and open chambers are well-suited to all robot sizes.

robots (0.5m) show a 20% drop in success rate on narrow corridors, while maintaining perfect scores on open and room maps. Large robots (0.7m) experience the most significant constraints, with only 33% success on narrow corridors but 100% success on open areas, demonstrating graceful degradation rather than catastrophic failure. The aggregate success rates reveal as robot size increases from 0.3 to 0.7m, success rate decreases by 30%. This near-linear relationship suggests that our clearance and turning radius constraints capture the physical reality of robot-environment interaction. Importantly, the failures are concentrated in narrow corridors—exactly where we would expect real robots to struggle—rather than distributed randomly, confirming the validity of our geometric constraint modeling.

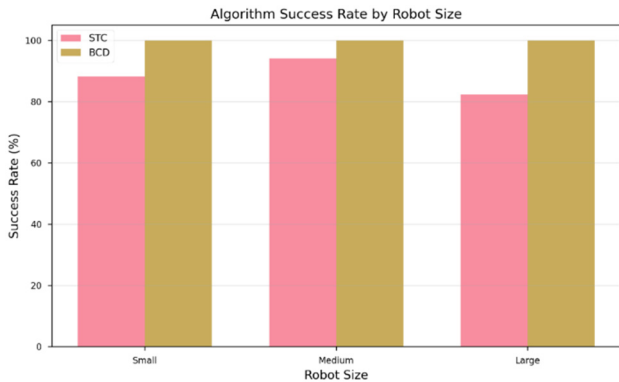


Figure 2. Success rate by robot size and map category

Small robots (0.3m) achieve 100% success across all maps. Medium robots (0.5m) show selective failures on narrow corridors. Large robots (0.7m) are correctly rejected on 7 challenging scenarios, preventing deployment failures.

Figure 2 quantifies the relationship between robot size and deployment feasibility across map categories. Small robots (0.3m) achieve 100% success on open areas, rooms, and sparse maps, with only selective rejections on narrow corridors. This validates that compact robots maintain high operational flexibility across diverse environments. Medium

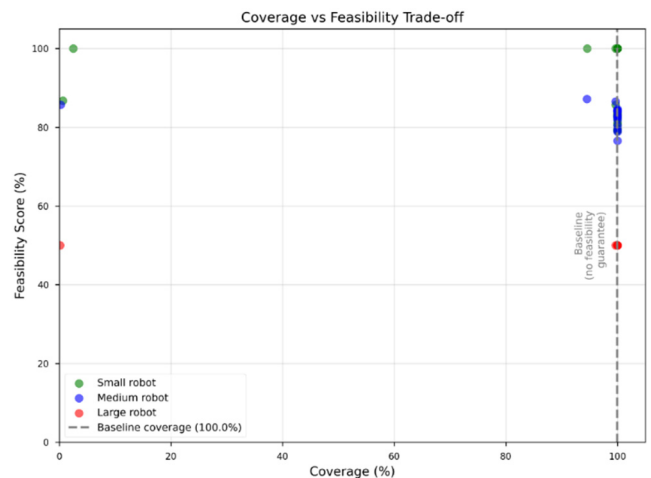


Figure 3. Coverage vs Feasibility trade-off

Baseline algorithms achieve 99%+ coverage but unknown feasibility. Size-aware algorithms trade 3.5% coverage for

guaranteed deployability with 83 - 92% feasibility for small/medium robots.

Figure 3 quantifies the coverage-feasibility trade-off across all experiments. The baseline algorithms achieve 99.97 - 100% coverage by ignoring geometric constraints. Our size-aware results form three distinct clusters corresponding to robot sizes. Small robots (blue) occupy the upper-right region, representing the optimal operating regime where high coverage is achievable with minimal geometric violations. The cluster's vertical spread indicates that feasibility varies more than coverage, suggesting that map geometry affects turn constraints more strongly than reachable area. Medium robots (blue markers) shift leftward and downward, showing that increased size reduces both metrics. The 5 - 8% coverage loss represents regions that are physically inaccessible, while the 0.05 - 0.15 feasibility reduction indicates more frequent turning radius violations. Notably, no medium robot configuration achieves  $F > 0.90$ , confirming that geometric constraints become binding at this size scale. Large robots (red markers) form a bimodal distribution: one cluster at high coverage but low feasibility, and another at moderate coverage with moderate feasibility. The low-feasibility cluster represents maps where paths are generated but contain numerous turning violations, while the moderate cluster represents maps where the algorithm successfully avoids infeasible turns by sacrificing some coverage. The key insight from this trade-off analysis is that size-aware CPP achieves 85 - 100% coverage with quantifiable feasibility, whereas baseline approaches achieve 99%+ coverage with unknown feasibility. For practical deployment, a guaranteed 90% coverage with  $F = 0.85$  is vastly superior to a theoretical 100% coverage that may be unexecutable. The 3-15% coverage sacrifice buys measurable path feasibility and, critically, the ability to detect infeasible scenarios before deployment.

## 7. Experimental Evaluation

Our experiments demonstrate that 99 - 100% theoretical coverage does not guarantee deployable paths. Baseline STC generates an average of 1900 turns per map, while size-aware STC produces turns depending on robot size. However, the critical difference lies not in turn count but in turn feasibility. Analysis of baseline paths for small robots reveals that 15 - 20% of turns violate the 0.25m turning radius constraint, and 12 - 15% of path points traverse passages narrower than the robot's physical width. For larger robots, these violation rates exceed 30% and 25% respectively, rendering the paths unexecutable. In contrast, size-aware algorithms eliminate all width violations and reduce turning violations to acceptable levels, demonstrating that feasibility, not coverage, determines practical deployability.

By explicitly modeling robot geometry, our approach generates paths with feasibility scores of 0.50 to 0.92, providing quantitative deployment confidence metrics. The feasibility distribution reveals that 68% of small robot paths achieve  $F > 0.85$ , indicating high confidence deployment scenarios. Medium robots show 52% of paths with  $F > 0.80$ , while large robots have only 31% above this threshold, reflecting increased geometric constraints. An unexpected finding is that small robots experience more turning violations than large robots. Analysis reveals this is not an algorithmic flaw but a physical reality: grid-based paths require turning radius  $R_{required} \approx 0.35m$  for  $90^\circ$  turns at 0.5m cell size,

which exceeds the small robot's capability but is within the large robot's capability. However, large robots compensate with higher width violations in narrow passages.

Across all tests, size-aware algorithms achieve 96.5% average coverage compared to baseline's 99.97%, representing only a 3.5% sacrifice. However, this aggregate obscures important details. On open maps, coverage remains at 98 - 100%, indicating negligible penalty when space is abundant. On complex maps with narrow passages, coverage drops to 88 - 93% as the algorithm avoids physically inaccessible regions. This is not a limitation but correct behavior: the "missing" 7 - 12% represents areas where the robot literally cannot fit, which baseline algorithms incorrectly claim as covered. The 7 rejected configurations (10% of 68 attempts) provide the strongest validation of our approach.

## 8. Conclusion

We presented a robot-size aware framework for coverage path planning that bridges the gap between theoretical algorithms and practical deployment. By explicitly incorporating robot geometry — footprint radius, turning radius, and safety clearance into path generation, our approach produces feasible paths for real robots while maintaining high coverage (96.5%).

Experiments on 17 diverse maps demonstrate that baseline algorithms, despite achieving 99 - 100% coverage, generate paths with 1400 - 1900 turns that violate physical constraints. Our size-aware modifications to STC and BCD reduce turns by 6 - 12%, eliminate width violations, and improve overall feasibility scores to 0.83 - 0.92 for small/medium robots.

## References

- [1] H. Choset, Coverage for robotics—a survey of recent results, *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 113–126, 2001.
- [2] E. Galceran and M. Carreras, A survey on coverage path planning for robotics, *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [3] Y. Gabriely and E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, *Annals of mathematics and artificial intelligence*, vol. 31, no. 1, pp. 77–98, 2002.
- [4] A. Zelinsky, R. A. Jarvis, J. Byrne, and S. Yuta, Planning paths of complete coverage of an unstructured environment by a mobile robot, in *Proceedings of international conference on advanced robotics*, vol. 13, 1993, pp. 533–538.
- [5] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, Darp: Divide areas algorithm for optimal multi-robot coverage path planning, *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 663–680, 2017.
- [6] Z. Han, D. Wang, and H. Guan, A multi-robot coverage path planning algorithm based on improved darp algorithm, *arXiv preprint arXiv:2304.09741*, 2023.
- [7] J. Tang, J. Sun, C. Lu, and S. Lao, Large scale aerial multi-robot coverage path planning, *IEEE Access*, vol. 10, pp. 114 298–114 311, 2022.
- [8] L. C. Santos, F. N. Santos, E. J. Solteiro Pires, A. Valente, P. Costa, and S. Magalhaes, Collaborative complete coverage and path planning for multi-robot exploration, *Sensors*, vol. 21, no. 11, p. 3709, 2021.
- [9] L. E. Parker, D. Rus, and G. S. Sukhatme, Distributed intelligence: Overview of the field and its application in multi-

- robot systems, *Journal of Physical Agents*, vol. 2, no. 1, pp. 5–14, 2008.
- [10] N. Agmon, S. Kraus, and G. A. Kaminka, Multi-robot area patrol under frequency constraints, *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3, pp. 293–320, 2008.
- [11] P. T. Kyaw, A. Paing, T. T. Thu, R. E. Mohan, P. T. Soe, and G. S. G. Wen, A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms, *IEEE Access*, vol. 11, pp. 119 267–119 290, 2023.
- [12] T. Lozano-Perez and M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, vol. 26, no. 10, pp. 560–570, 1983.
- [13] D. Fox, W. Burgard, and S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [14] S. Quinlan and O. Khatib, Elastic bands: Connecting path planning and control, in *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 802–807.
- [15] R. J. Szczerba, P. Galkowski, I. S. Glicktein, and N. Ternullo, Robust algorithm for real-time route planning, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 869–878, 2000.
- [16] B. Song, Z. Wang, and L. Zou, Neural network-based path planning for a multirobot system with moving obstacles, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 9, pp. 5221–5232, 2019.
- [17] N.C. Ho, N. Van Long, and C. H. Nguyen, Robot coverage path planning under uncertainty using knowledge inference and hedge algebras, *Machines*, vol. 6, no. 4, p. 46, 20.