Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Decentralized Multi-Robot Warehouse Surveillance with Orientation-Aware Planning and Hybrid Collision Avoidance

Praveen Paidi

ppaidi@asu.edu

Research Article

Keywords:

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11245096/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Decentralized Multi-Robot Warehouse Surveillance with Orientation-Aware Planning and Hybrid Collision Avoidance

Praveen Paidi (corresponding author)

Arizona State University, USA

Ppaidi@asu.edu

Abstract

The surveillance of critical or hazardous indoor environments — warehouses, industrial facilities, and convenience stores — is often too risky or expensive for human operators, motivating coordinated multi-robot teams. This paper presents a decentralized pipeline for grid-based multi-robot surveillance: a density-weighted utility function chooses high-priority targets, an orientation-penalized variant of A* plans turn-efficient routes to them, a Linear Quadratic Regulator(LQR) tracks the resulting trajectories, and a three-tier collision-avoidance scheme — escalating from artificial potential fields to grid-aligned shape-navigation boundary tracing to a bounded local A* replan — coupled with a time/cost-based negotiation protocol keeps robots clear of static obstacles, dynamic obstacles, and each other. We demonstrate that the orientation-aware cost function preserves A* optimality using 2 different robot sample sizes.

The orientation-aware planner decreased in-place turning by 70.19% over 40 randomized trials, matching or beating baseline A* in all 40 of the 40, and by 64.85% (N=15) / 59.11% (N=25) on the real targets utilized throughout, cutting real execution time under LQR monitoring by 25.13% (N=15) / 21.60% (N=25). Collision-avoidance escalation resolved 15 of 15 (N=15) and 25 of 25 (N=25) dynamic-obstacle encounters and negotiation resolved all 6 genuine timing conflicts found across both team sizes (1 of 105 pairs at N=15, 5 of 300 at N=25). The results show that the pipeline is a feasible and scalable method to multi-robot surveillance of congested interior settings, with the target-selection mechanism validated for team sizes from 5 to 30 robots and the full pipeline verified at two representative scales, N=15 and N=25.

1. Introduction

Deploying human operators to monitor indoor industrial environments raises concerns about exposure to hazardous compounds, structural instability, and the cost of routine monitoring tasks like inventory verification and safety compliance checks. Multi-robot systems, in which a small fleet of affordable mobile robots patrols a warehouse, factory floor, or retail without human interaction, show potential. These robots must find the most essential monitoring sites, reach them quickly, and avoid static objects, unanticipated moving obstacles, and each other.

Extensive research exists for each of these three criteria independently. Target selection is done using utility- and frontier-based exploration approaches, efficient path planning is performed using A* and its several cost-augmented variants, and collision avoidance is done using artificial potential fields, navigation functions, and reciprocal-avoidance protocols. Even rarer is a single lightweight pipeline that integrates all three for a forward-driving robot model that is grid-constrained. This is the type of inexpensive differential-drive platform that is commonly used in warehouses and must be simple enough to run in real-time on limited onboard computation. This work provides a simulated evaluation and characterization of such a pipeline.

In this work, we propose a density-weighted utility function that selects surveillance targets by trading off the cost of reaching each possible cell against the coverage of undiscovered territory; an orientation-aware modification to the A* cost function that discourages in-place turning, along with evidence that the modification maintains A*'s optimality guarantee; a hybrid scheme for collision avoidance that reverts to grid-aligned shape-navigation boundary tracing from artificial potential fields when the potential field oscillates or stalls in a local minimum due to concave obstacle geometry; and a simple negotiation technique that avoids deadlock without centralized coordination by resolving robot-robot path conflicts based on travel duration and cost. The simulation evaluation was run across various obstacle and multi-robot conditions in a grid map approximating a warehouse, measuring 100 m by 100 m, using the same 15 real robots and targets selected according to utility functions.

2. Literature review

2.1 Target Selection and Coverage Planning

The basic frontier-based exploration algorithm [1] is used to lead the robots to the boundary between explored and unexplored space. The border is considered as a possible target location without taking into account the cost of reaching there. This work follows the utility-minus-cost model of Burgard et al. [2], where the usefulness of a frontier is decreased if another robot is already moving toward it.This study achieved a minimum pairwise target spacing settling around 8-10 cells past N=20, with zero duplicate assignments across 5 randomized trials on the 100 m × 100 m map for any of the team sizes considered from 5 to 30 robots. The discount is instead proportionate to the profit of every cell. Butzke and Likhachev [3] field-validated a multi-objective utility function to expand frontier exploration utilizing competitive victories in unknown-terrain exploration. This function balances information gain, area assignment, inter-robot distance, and goal-switching hysteresis. This work uses a variation of their hysteresis technique to ensure that no robot leaves its target unless a new candidate's profit exceeds a predetermined margin, which is constant over 300 robot-target assignments. The utility function closest to this work was created by Quintero, Oñate López and Bertel [4]. A Gaussian-correlated quantity, the density ratio, exchanged coverage gain for reach cost, with orientation-dependent path cost, implemented using a modified version of Dijkstra's method. However, they employed Dijkstra instead of A*, estimated the utility function using a neural network or evolutionary algorithm instead of computing it directly, and used just negotiation, ignoring reactive obstacle avoidance. Lin and Huang [5] reactively balance individual and team coverage gains based on live sensor sweeps while exploring novel regions. Yao et al. [6] validated their technique on 20 real UAVs and increased the prior-knowledge UAV mission-planning utility estimate by 14.84% with the use of post-knowledge from real route planning. Ahmed et al. [7] employed frontier objectives to lower localization and mapping uncertainty instead of coverage need and found an improvement of 32% over similar techniques. Krupke [8] explicitly addresses turn cost for full-area coverage with verified optimality requirements. The utility function employed by Kobayashi et al. [9] to choose patrol targets considers both the need for patrols and the urgency of reporting. It needs regular revisits to guarantee that there is minimal idle time. In their formalization of priority-weighted coverage, Lee, Kim, Li, and Park [10] ensure that residual Steiner-tree pathways will lead to full coverage in the end. In this simplified version of the concept, the coverage-need term is multiplied by a priority map that does not guarantee full coverage. A 125-work evaluation of coverage-planning efforts spanning visual, learning-based, restricted, 3D, and multi-robot categories was published in a recent study [11].

In contrast to previous works, this one doesn't optimize density-based coverage need but rather visits every cell eventually, targets one-time exploration of unknown terrain, or optimizes an objective such as localization uncertainty,

patrol idleness, base-station awareness instead. Moreover, none of these works choose only the highest-need cells for repeated, selective surveillance of an already-known map without a full-coverage requirement. The discount, hysteresis, and priority-weighting procedures are designed for this more restricted perspective, and their zero-duplicate, stable-assignment outcomes for teams ranging from 5 to 30 robots corroborate this. In this paper, the average number of targets falling in one quadrant increased from 3.60 to 14.80 out of 15 in 20 randomized trials involving 15 robots, with a threefold weight on that quadrant.

2.2 Cost-Augmented Path Search

Although A* [12] guarantees minimum-cost pathways under an acceptable heuristic and remains the grid-based standard, its classical formulation allows orientation modifications to regulate execution time for axis-aligned robots without penalizing them. Wang [13] removes the turning-cost term from the universal mobile robot path planning A* evaluation algorithm. Xiang et al. [14] reformulated the assessment method to eliminate unnecessary nodes and directly reduced cumulative turning angle, lowering total turning angle by 75% and path length by 5%. Cao et al. [15] weigh the cost term by obstacle density and inflate obstacle boundaries for safety margin to reduce search nodes by 80% without orientation. Liu et al. [16] presented workshop AGVs with turning weights in the A* evaluation function that had a 69% lower turning angle and 10% shorter path than standard A*. Wang et al. [17] found that dynamically weighted heuristics and dynamic windows minimize turning spots by 71.4%. A broader evaluation categorizes the prevailing methodologies into heuristic weighting and hybrid integration with local planners (DWA, APF, PSO). The decrease in effort for this study (Section 5.2) - 70.19% on randomized trials, 64.85% (N=15) / 59.11% (N=25) on the actual targets employed - falls within the same range as the three prior AGV-centric investigations ([14]: 75%, [16]: 69%, [17]: 71.4%); Din, Ushigome, and Fujimoto [18] propose an Orientation-Aware Advanced A* trading path length against turning effort that minimizes directional changes by 60% on a wheelchair robot using a heuristic-admissibility proof paralleling Section 4.3. Spline smoothing, adaptive neighborhood expansion, and redundant-turn removal are used by Li, Li, and Dai [19]. Obstacle clearance, bidirectional search, and turn-reducing smoothing reduce right-angle turns by 100%, according to Wang et al. [20]. The 70.19%/64.85% drop in this study is comparable to the data presented in [14], [16], [17], and [18] – [20] showing orientation-aware A* is active. The work is distinguished by a formal A* optimality demonstration (Section 4.3), a 40-trial baseline comparison, and implementation in a multi-robot surveillance pipeline (Section 2.5) rather than as a standalone planner.

2.3 Potential-Field and Navigation-Function Obstacle Avoidance

Simple and localized artificial potential fields include problems like oscillation around concave obstructions and local minima [21]. This study modifies the shape-navigation methodology of Hsieh, Loizou, and Kumar [22] — originally designed for continuous sphere-world or star-world environments — to accommodate axis-aligned grid impediments through piecewise-linear boundary segments, serving as a contingency when the potential field has difficulties with concave barriers. Tandogan [23] employs a conventional artificial potential field controller for internal navigation and does not explicitly consider the problem of concave obstructions. Vasilopoulos and Koditschek [24] generalize the navigation-function framework to partially known non-convex environments but assume the robot will eventually learn the precise shapes of each obstacle when it approaches them. Sawant, Polushin and Tayebi [25] provide a hybrid feedback controller with an improved guarantee of global asymptotic stability with Zeno-free switching, but with considerably increased complexity over the piecewise-linear tracing given in this paper. Kim et al. [26] start the wall-following behavior reactively when the norm of the force of the potential field becomes close to zero, and it outperforms existing solutions in terms of success rates in decentralized multi-robot navigation. In a similar manner, Woo and Hirota [27] set up virtual walls around the local minima that are detected, to start wall-following in the case of a single robot. The present work differs from the reactive pair [26], [27] in that it traces the actual grid boundary directly via discrete, axis-aligned segments instead of a heuristically triggered behavior, and differs from the stronger guarantees in [24], [25] in that it sacrifices a stronger (global, geometry-independent) stability result for a much simpler controller that is designed for axis-aligned warehouse obstacles instead of arbitrary non-convex geometries.

2.4 Multi-Robot Negotiation and Decentralized Avoidance

A decentralized, priority-driven negotiation mechanism that includes waiting time and path cost resolves robot conflicts [4]. By centrally investigating a hierarchy of priority arrangements, Conflict-Based Search [28] finds demonstrably optimal, collision-free team solutions, whereas ORCA [29] continuously and symmetrically handles conflicts without prioritization or communication. This protocol defines priorities unilaterally, more like decentralized priority-planning than ORCA's balanced modification. Wu, Bhattacharya, and Prorok [30] prioritize based on the number of topologically unique path alternatives available to a robot, unlike this study's arrival-time-and-cost criterion; Tang, Mao, and Ma [31] resolve conflicts for up to 100 robots using a centralized, offline MAPF phase.

2.5 Positioning of This Work

This research consolidates utility-oriented target selection, cost-aware A*, a three-tier potential-field navigation avoidance escalation, and efficient negotiation into a cohesive system intended for budget-friendly, grid-limited warehouse robots, with each segment supported by a controlled comparison rather than an independent demonstration (Sections 5.1–5.4). This constitutes a more comprehensive set of controlled, measurable findings throughout each phase of the pipeline that any singular research examined above delivers for its respective part.

3. Problem Formulation and System Model

The monitoring area is shown as a confined 100 m × 100 m occupancy grid consisting of 1 m by 1 m cells, filled with stationary obstacles positioned to simulate a warehouse layout. Each robot is designed as a circular platform with a radius of 0.4 meters, outfitted with camera and LIDAR sensors, and possesses complete awareness of the static map. Robots traverse the four primary directions, incurring a unit expense for each cell moved and an extra two-unit charge for reorienting, performed in place via a front steering wheel.

In accordance with this model, the surveillance aim is to (i) determine the specific grid cells requiring surveillance attention at a particular moment, (ii) allocate each available robot to a designated target cell, (iii) devise a cost-effective route — factoring in orientation expenses — from each robot's existing location to its assigned target, and (iv) implement that route while circumventing collisions with fixed obstacles, unforeseen dynamic obstacles, and other robots navigating their respective paths simultaneously.

4. Methodology

The pipeline has six phases, carried out in accordance with the planning cycle: utility-driven target selection, robot-target allocation, orientation-sensitive path planning, trajectory monitoring, hybrid collision prevention, and negotiation for scheduling conflicts among robots.

4.1 Utility Function for Target Selection

The grid map is conceptualized as a binary occupancy representation, where traversable grid cells are allocated the value 1 and obstacle grid cells are allocated the value 0. The agents obtain $S_a(x, y)$ via a predetermined exclusion

mask that marks the cells near each robot's existing location as previously filled, and the obstacles are indicated by $S_b(x, y)$ through the obstacle occupancy grid. The relationship between these two establishes a coverage-density function that peaks in wide spaces devoid of any robotic presence, shown in Eq. (1).

$$C(x, y) = \frac{S_a(x,y)}{S_b(x,y)} \quad (1)$$

For every candidate cell j, the information term I_{ij} represents the min-max normalized coverage density:

$$I_{ij} = \frac{C(x_j,y_j) - C_{min}}{C_{max} - C_{min}} \quad (2)$$

where $C_{min}$and C_{max} represent the least and greatest values of $C(x, y)$ over all cells inside the grid. For every robot i, the cost component C_{ij} integrates the Euclidean distance from the robot to cell j together with an orientation penalty that reflects the robot's present heading, estimating the rotations required to arrive at the cell, and is normalized to a range of [0, 1]. Each robot identifies the cell with the highest profit as its subsequent goal, while disregarding obstacle cells and its present location. The profit function combines the two in the following manner:

$$P_{ij} = \alpha * I_{ij} - \beta * C_{ij} \quad (3)$$

Three upgrades are layered onto this essential profit function. At the outset, once a target is designated to a robot, a Gaussian discount is applied to the profit map focused on this target before the next robot makes its target selection.

$$P_{ij} = P_{ij} \times (1 - 0.9 \times \sum_{k \in T} \exp\{-\frac{d(j,k)^2}{2.\sigma^2}\}) \quad (4)$$

where T is the collection of cells that have been designated as targets by robots assigned earlier, $d(j, k)$ represents the Euclidean distance between candidate cell j and the previously designated target k, and sigma regulates the extent of each discount's effect. A cell in proximity to several already-acquired targets gathers a reduction from each. Secondly, a robot will only transition to a new candidate if the candidate's profit exceeds the current profit of the prior target by a factor of gamma; otherwise, it retains its old target. Third, a preferential weighting may be assigned to the information term prior to the profit calculation,

$$I'_{ij} = I_{ij} * \omega_j \quad (5)$$

where ω_j is a designed priority weight assigned to cell j. Generally, ω_j equals 1; however, for cells situated in specific high-value or dangerous areas, $\omega_j > 1$, rendering them highly appealing irrespective of their current level of coverage and I'_{ij} substitutes I_{ij} in the profit calculation for all ensuing target choices.

4.2 Robot-Target Assignment

The robot-target assignment begins by enumerating all possible robot-target pairings, N robots and M candidates, which leads to N×M pairs, then computing each pair a real A* cost considering the orientation, ordered in increasing order.

4.2.1 Multi-Start Assignment

The greedy strategy of a single sweep across a sorted collection of pairs results in a speedy complete assignment but is prone to an arbitrary early choice. The least expensive pair can constrain a suboptimal range of remaining choices for every other robot. To surmount this challenge, the approach creates many candidate assignments—a multi-start heuristic—where each is a greedy iteration commencing from a separate pair, advancing from the globally most economical to each pair in succession, ensuring the retention of the lowest-cost solution.

This approach possesses an intrinsic quality floor, given that the candidate originating from the least expensive pair globally is fundamentally identical to a singular greedy pass. Therefore, the lowest value across all candidates can only be equal to or better than the greedy method, never inferior. This is an authentic but humble guarantee, not an optimality limit, and a less formal style of evidence than the A* optimality demonstration (Section 4.3).

We intentionally incorporate all seeds rather than only the most economical options, as the optimal seed is not invariably the least expensive worldwide, and a limited search may overlook the most feasible allocation by examining only a small subset of a sorted array (Section5.1). A more restricted threshold under the total N×M seeds might regain a significant portion of the benefit at a reduced expense, serving as a basis for future adjustments.

4.2.2 Handling unequal robots and target counts

A comprehensive pairing necessitates a minimum of N candidate targets for N robots. When M is more than or equal to N, the process operates as previously outlined. When M is less than N, the M robots with the lowest optimal cost

engage in multi-start assignment as a legitimate M-robot subproblem; the remaining N−M robots independently choose targets based on the utility function from Section 4.1, which is assessed across the entire grid and adjusted for conflicts with the rest of the team using a uniform Gaussian discount instead of a distinct rule (Section5.1).

4.3 Orientation-Aware A Path Planning*

The conventional A* search algorithm seeks to minimize $f(n) = g(n) + h(n)$, with $g(n)$ representing the cumulative cost from the initial node and h(n) denoting the projected cost to the goal. Due to the robot's grid-constrained forward-drive kinematics rendering orientation adjustments costly relative to linear motion, an orientation expense is included into the accumulated-cost component: each instance the search progresses from parent to child.

$$g(child) = g(parent) + move_cost + r(parent, child) \quad (6)$$

where $move_cost$ represents the constant expense of traversing a single grid cell, and $r(parent, child)$ denotes twice the least number of 90-degree turns required to transition from the parent's orientation to the child's orientation. This modification is implemented at every stage from the initial node to node n. It is clearly indicated that $g(n)$ signifies the cumulative sum of both movement and orientation costs incurred over the entire path, rather than only the movement cost.

$$g(n) = \sum_{i}^{n} [move_{cost_i} + r_i] \quad (7)$$

The orientation expense is thus included in each successive application of $g(n)$, as seen in the total cost function below—it is inherently integrated inside $g(n)$ as a cumulative total rather than being explicitly stated as an independent factor.

$$f(n) = g(n) + c * h(n) \quad (8)$$

$f(n)$ refers to $g(n)$precisely because g(n) already includes the orientation cost. The heuristic h(n), the true Euclidean distance to the goal, is weighted down by a tunable factor c so the search doesn't over-prioritize straight-line proximity at the expense of that accumulated cost.

This modification preserves A* optimality under the augmented cost model. Because the search finalizes each node on expansion and never reopens it, the stronger property of consistency — which implies admissibility — is required. $c * h(n)$ is consistent: for any node n and neighbor n′ reachable by a single move,

$$c.h(n) \leq cost(n, n') + c.h(n') \quad (9)$$

Each move is one step in a cardinal direction on the grid, so the Euclidean distance between n and n' is precisely 1. Then the triangle inequality provides

$$h(n) \leq 1 + h(n') \quad (10)$$

$$c.h(n) \leq 1 + c.h(n') \leq \ cost(n, n') + c.h(n') \quad (11) \text{ as } 0 < c <= 1$$

Independently, the true expense of this transition is $cost(n, n') = move_cost + r(n, n')$, where $move_cost$ = 1, and the orientation penalty $r(n, n')$, is always non-negative, therefore $cost(n, n') \geq 1$.. Integrating the two limits' results in consistency criteria:

$$c.h(n) \leq cost(n, n') + c.h(n') \quad (12)$$

This is precisely the requirement of consistency. A reliable heuristic ensures that once a node is extended, no subsequently found path to it may be less expensive, hence the search employment

$$f(n) = g(n) + c * h(n) \quad (13)$$

yields a route with minimal overall enhanced expense from origin to destination, given that $0 < c \leq 1$.

4.4 Trajectory Tracking via LQR

Upon devising a certain trajectory, a Linear Quadratic Regulator is employed to navigate the robot along the generated waypoints. The robot's continuous time kinematic state is represented as $q = [\, x, y, \theta, v]^T$, with dynamics $\dot{x} = v.\cos(\theta)$, $\dot{y} = v.\sin(\theta)$, $\dot{v} = a$, $\dot{\theta} = v.\frac{\tan(\delta)}{r}$, where δ denotes the steering angle, r signifies the turning radius, and a indicates the longitudinal acceleration. Linearization centered on the reference trajectory utilizing the tracking-error

state $s = \left[e, \dot{e}, \theta_e, \dot{\theta}_e, \delta_v \right]^T$ encompassing lateral position error, its derivative, heading error, its derivative, and velocity error -- alongside control inputs $u = [\, \delta, accel]^T$ yields a discrete-time error model of the form

$$s(k+1) = A * s(k) + B * u(k) \quad (14)$$

A and B are populated based on the present velocity, turning radius, and timestep, and recalculated at each control interval instead of being predetermined offline. Given that A and B are contingent on velocity, the discrete algebraic Riccati equation is recalculated in each iteration to derive a gain matrix K suitable for the robot's current velocity, instead of relying on a singular gain determined for a specific nominal operating condition. This is a real-time gain-scheduled LQR controller: the control law $u^* = -K^* s$ utilizes this newly calculated K at each iteration, optimizing a quadratic cost with state-weighting matrix Q and control-weighting matrix R. The steering directive comprises a feedforward element reliant on local path curvature and a feedback adjustment from the LQR control law:

$$\delta = atan2(r \cdot \kappa, 1) + pi_range(u_1{}^*) \quad (15)$$

where kappa signifies the curvature of the reference path at the nearest tracked waypoint (distinct from the gain matrix K), r represents the turning radius, and pi_{range} confines the output within the range of $[-\pi, \pi)$. The directive for acceleration is clearly stated as $accel = u_2{}^*$.

The intended speed diminishes progressively to an acceptable minimum as the robot approaches the goal, prompting the controller to lower speed to avoid impact at full velocity. In the assessment (Section 5.2), it was determined that ongoing monitoring of each linear segment through curves led to an actual, confirmed obstacle collision when a sharp turn succeeded a lengthy straight path; consequently, the controller was reengineered to monitor each segment with comprehensive feedback control and to perform turns as a distinct stop-and-rotate action, without tracking during the turn.

4.5 Hybrid Collision Avoidance

Collision avoidance escalates through three tiers only as needed.

Artificial potential fields — default layer

The potential of each grid cell integrates a quadratic appealing component directed toward the objective alongside a repulsive component aggregated from all obstacles within the robot's collision radius.

$$U(x,y) = \frac{1}{2}(k_a . d_{goal}(x,y)^2 + sum\ over\ obstacles\ o\ of\ U_{rep}(x,y,o)) \quad (16)$$

$$U_{rep}(x,y,o) = \frac{1}{2}(k_r . \left(\frac{1}{d_0(x,y)} - \frac{1}{r_c}\right)^2\ if\ d_0 \le r_c, else\ 0. \quad (17)$$

where d_{goal} represents the distance to the objective, d_0 denotes the distance to obstacle o, and r_c signifies the collision radius (0.4, a slight margin beyond the physical dimensions). Calculating the total for each proximate obstruction accurately assesses cells confined between two barriers as more perilous than if only one existed, instead of categorizing them as secure. The potential is calculated first, normalized, and the robot proceeds to the adjacent cell with the lowest score — a finite grid exploration. An oscillation detector identifies a stall if the subsequent selected cell duplicates one of the previous eight cells visited, or if the planner surpasses a certain step budget without achieving convergence.

Shape navigation — escalation

When the potential field loses its efficacy, the robot navigates the obstacle's perimeter in a consistent cardinal direction progressing in linear paths only, turning at designated corners rather than tracing a continuous curve utilizing the same stop-and-rotate method described in the foundational A* and LQR framework.

An initial closed-form navigation function was assessed but demonstrated uncontrolled lateral drift during extended evaluations, due to its boundary being derived from a coordinate misaligned with its intended target; two attempts to rectify the underlying formula worsened the drift rather than alleviating it. Grid-aligned boundary tracing entirely supplants the closed-form trajectory -- a separate procedure, rather than a fix to the one it replaces.

Local A replan — final fallback*

In instances where shape navigation is unable to attain the reconnect target due to overly intricate surrounding obstacle geometry that precludes a direct linear slide, a constrained local A* search is executed directly from the escalation's present location to the original reconnect target, employing the identical orientation-sensitive cost model as the

principal planner (Section 4.3). This search is scoped only to the local detour, not the robot's complete trajectory, and uses the same planning technique already validated in Sections 4.3 and5.2 -- not a separate, unproven mechanism introduced solely as a fallback.

4.6 Negotiation (conflict resolution)

The trajectory of each robot is transformed into projected arrival times for each cell. Each combination of pathways is examined for a common cell accessed simultaneously; the robot with lower priority (later arrival, greater cost as a tiebreaker) has its route prolonged with a delay prior to reaching the conflict, and this procedure continues until all conflicts are resolved. Among the existing collection of 15 robot trajectories, one inherent conflict occurs and is resolved in a single pass.

5. Simulation Results

The complete pipeline was executed in Python, encompassing utility functions, A* planning, robot-target allocation, artificial potential fields, shape navigation, negotiation, and LQR trajectory tracking, and was assessed on a 100 m by 100 m grid warehouse layout.

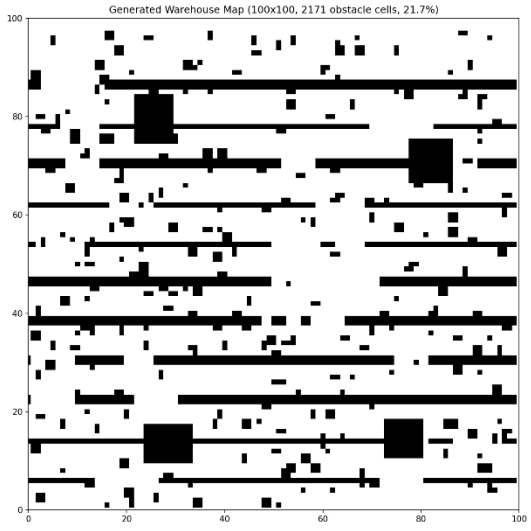


Figure 1 --100 m × 100 m warehouse map

Findings from two separately assessed team sizes, N=15 and N=25, demonstrate the varying limits of the priority zone's actual capacity; a thorough range from 5 to 30 robots confirms the trend as well. Rather than executing a solitary representative run, each component was evaluated through randomized setups: the orientation-aware planner was scrutinized with 40 start-goal pairs, target selection and assignment were assessed across 5 to 20 randomized configurations for team sizes between 5 and 30, and collision avoidance and negotiation were analyzed within the scenarios outlined in Sections 5.3 and 5.4 at both team sizes — all performed on a single map with randomized positions, goals, and team compositions.

5.1 Target Selection and Assignment Results

The target-selection discount associated with previously claimed cells (Section 4.1) aggregates with each prior claim during a robot's turn, not solely the latest claim, which accounts for the zone possessing a tangible, quantifiable capacity instead of an infinite one: as additional robots claim targets within it, the cumulative discount eventually pushes the remaining zone cells below what any cell outside the zone can offer. To determine if this capacity constraint persists across various team sizes, a range is examined from 5 to 30 robots.

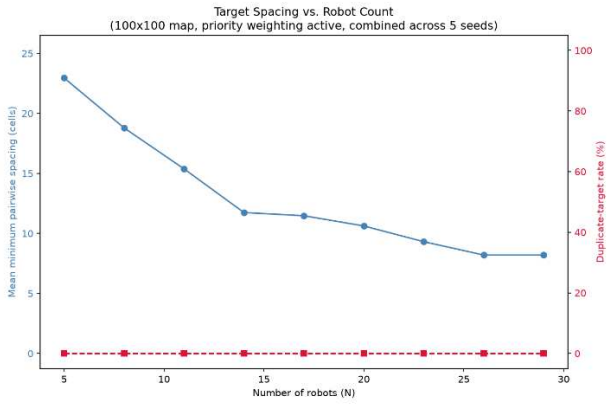


Figure 2 -- Target spacing versus team size.

Two specific locations on this curve illustrate the phenomenon: at N=15, all targets fall within the priority zone; at N=25, the zone's capacity is evidently surpassed, resulting in several targets being diverted. The size of a team produces two opposing, perhaps conflicting trends. As the team grows, the minimum pairwise distance decreases and stabilizes between 8-10 cells, aligned with the spatial width parameter of the discount, after around 20 robots,

rather than converging to zero, while the mean of all pairwise distances rises — a separate inquiry regarding the team's overall distribution, not a contradiction.

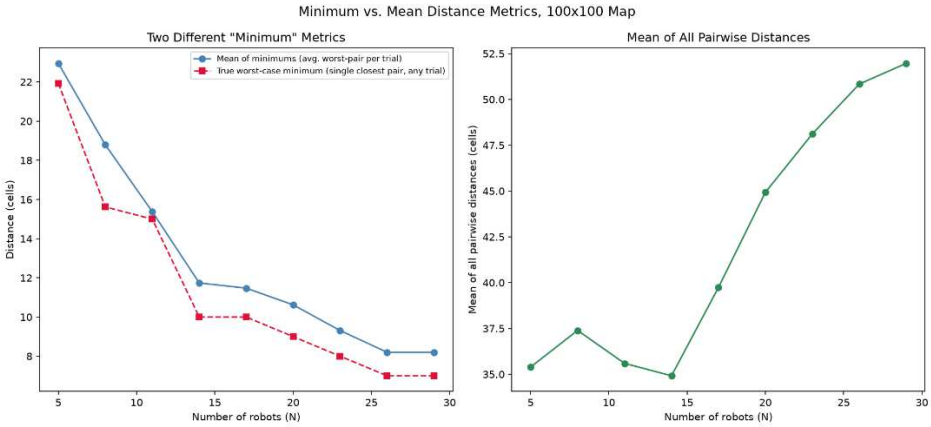


Figure 3 -- Left: two minimum-spacing measures. Right: mean of all pairwise distances.

Two distinct team sizes streamline this transition: at N=15, every target is situated within the priority zone (top-left quadrant, 3x weight); at N=25, the zone's capacity is clearly surpassed, with 19 of 25 staying within and 6 diverted to higher-profit cells in other areas of the map.

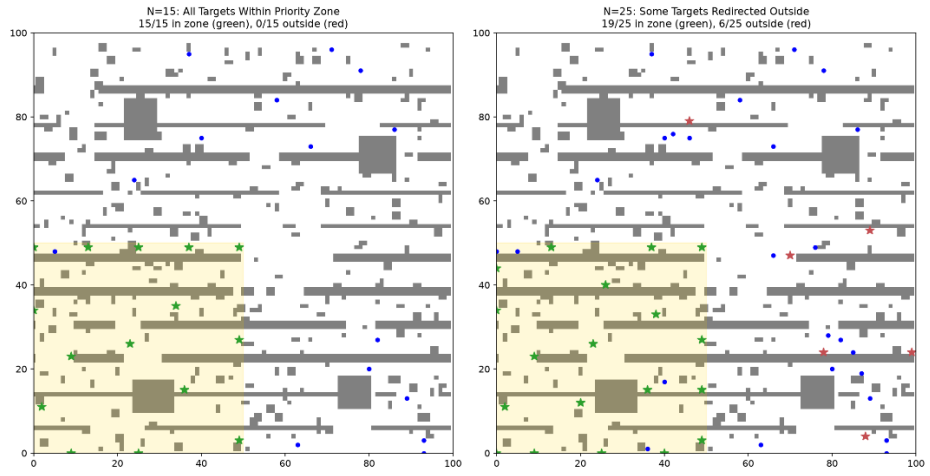


Figure 4 -- Left: N=15, every target inside the shaded priority zone. Right: N=25, six targets are redirected outside

This N=25 experiment places 76% of targets within the zone, nearly aligning with the sweep's 5-seed mean of 73.60% anticipated variation between an individual trial and a multi-seed average. The minimum spacing of 9.00 cells is similarly near the average of the sweep, which is 9.31 cells.

Three enhancements applied to the foundational utility function were evaluated across both team sizes, with 20 randomized trials conducted for each. The proportional discount, in contrast to a strict exclusion zone, increases the average minimum pairwise spacing from 7.63 to 11.65 cells at N=15 (worst case 4.47 to 10.63), and from 4.01 to 9.40 cells (worst case 1.00 to 7.00) at the greater density of N=25 -- a comparably strong improvement at both team sizes, with N=25 showing the larger relative gain (134% vs. 53% at N=15). Hysteresis maintains all 300 distinct assignments constant at N=15, and all 500 at N=25, across a recurring planning cycle with unchanged robot positions. Priority weighting, applying a threefold emphasis on a certain quadrant, elevates the average number of targets in that quadrant from 3.60 out of 15 to 14.80 out of 15 at N=15, and from 6.10 out of 25 to 19.10 out of 25 at N=25 -- demonstrating a consistent and significant redirection impact across both scales.

Robot-target alignment employs one of two methodologies. Greedy traverses the ordered list of all potential pairings exactly once; multi-start repeats that traversal starting from every possible initial pair, and keeps the cheapest complete outcome.

Team size (N)	Greedy time	Multi-start time	Time ratio	Cost improvement
5	0.03 ms	0.08 ms	3x	8.0%
10	0.06 ms	0.70 ms	12x	6.0%
15	0.09 ms	3.12 ms	34x	4.2%
20	0.16 ms	9.21 ms	59x	4.1%
25	0.27 ms	23.71 ms	89x	5.0%

Table 1 -- Assignment cost versus computation time, mean over 5 randomized configurations per team size.

At N=15, the multi-start approach incurs a cost of 3.12 ms compared to greedy's 0.09 ms, yielding a 4.2% cost decrease; at N=25, the disparity escalates to 89 times yet remains below 300 ms in total, achieving a 5.0% reduction. Multi-start is employed at both scales on this premise -- the computational expense is minimal compared to a planning cycle that is not executed many times per second, at either team size tested.

Metric	N=15: Greedy	N=15: Multi-start	N=25: Greedy	N=25: Multi-start
Mean total cost	906.10	860.50	1263.80	1190.00
Trials won outright	0/10	10/10	0/10	10/10

Table 2 -- Assignment quality, simple greedy vs. multi-start, both team sizes, 10 trials each.

The importance of examining all seeds instead of only the least expensive ones is evident in the source of optimal outcomes, and this impact becomes increasingly significant as team size increases. In 15 randomized trials with N=15,

the rank of the winning seed varied from 4 to 117 among 225 total seeds, and in all 15 of the 15 trials, the winning seed was not the least expensive pair. At N=25, within the identical 15 trials, the rank of the winning seed varied from 16 to 430 among 625 total seeds -- and in every one of the 15 trials, the champion was not the least expensive pair. With 625 potential beginning seeds rather than 25, the least expensive pair is correspondingly less likely to yield the optimal full assignment.

A comprehensive examination assesses if the three refinements to the utility function collectively surpass a simplistic nearest-cell benchmark, across both team sizes:

Metric	N=15: Naive	N=15: Full	N=25: Naive	N=25: Full
Mean min spacing	7.63	11.65	4.01	9.40
Worst-case spacing	4.47	10.63	1.00	7.00
Duplicate-target trials	0/20	0/20	0/20	0/20
Targets in zone (mean)	3.40/15	14.95/15	6.30/25	19.20/25
Mean total profit	14.58	40.03	24.29	57.28

Table 3 -- Combined utility-function test, naive baseline vs. full mechanism

The full utility function wins decisively across every metric at both team sizes, without exception. The profit row provides the clearest comparison, as it evaluates each assignment against the identical shared objective (coverage need versus travel cost) that the mechanism optimizes, rather than relying on a proxy such as spacing or raw distance: the full mechanism realizes 174.6% greater total profit than the naive approach at N=15, and 135.8% more at N=25 -- both a substantial benefit, unlike the more team-size-sensitive spacing metrics discussed above.

5.2 Orientation-Aware A Path Optimization and LQR Results*

Both team sizes utilize the identical targets chosen by the utility function in Section 5.1 to formulate authentic A* pathways from the actual positions of the robots to their respective objectives, applying both cost models -- every path corresponds to a target that the utility function picked. The gain-scheduled LQR controller (Section 4.4) utilizes comprehensive feedback control to follow each straight section and stops to rotate in place at each turn, with tracking ceasing during the turn. During its movement along a straight path, the robot follows the reference line previously validated by the planner as obstacle-free, and as rotation does not induce lateral drift, the robot's actual course remains aligned with the confirmed safe line.

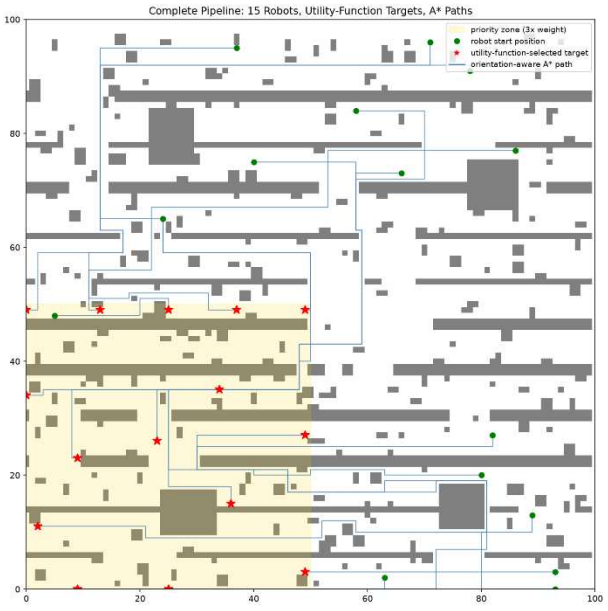


Figure 5 -- 15 robots: A paths.*

At N=15, orientation-aware A* lowers average turning from 19.53 to 6.87 (64.85%); at N=25, from 17.12 to 7.00 (59.11%) -- a comparably strong reduction on an independently generated set of 25 real targets.

Metric	N=15 Baseline	N=15 Orientation-aware	N=25 Baseline	N=25 Orientation-aware
Mean turns per path	19.53	6.87	17.12	7.00
Mean A* path cost	142.33	117.40	133.36	113.44
Mean execution time	88.51s	66.27s	82.11s	64.38s
Time savings	—	25.13%	—	21.60%
Mean final distance	0.0384	0.0414	0.0386	0.0415
Collisions	0/15	0/15	0/25	0/25

Table 4 -- Orientation-aware A and LQR tracking results*

Reduced turns speed execution by 25.13% and 21.60% at N=15 and N=25, respectively, since each turn has a consistent rotation period plus deceleration and re-acceleration around it. The overall A* path cost, which integrates movement and turn penalties within the same model validated in Section 4.3, is similarly reduced for orientation-aware strategies at both team sizes (17.52% and 14.94%), proving a truly more economical path under its own cost framework, not merely fewer turns at a hidden cost elsewhere.

5.3 Artificial Potential Field Behavior and Its Limits, Shape-Navigation Escalation

The artificial potential field addresses a dynamic impediment that arises on a pre-established A* trajectory by calculating a small diversion and rejoining the initial path beyond the obstruction, instead of reconfiguring the entire route. A moving obstruction was incorporated at the middle of each of the 15 real A*-planned trajectories from Section 5.2, and the reroute-and-reconnect functionality was evaluated directly:

Outcome	Trials
Successfully reconnected to the original path	11/15
Oscillation, did not reconnect	4/15

Table 5 -- Local dynamic-obstacle avoidance outcomes, 15 real A-planned paths*

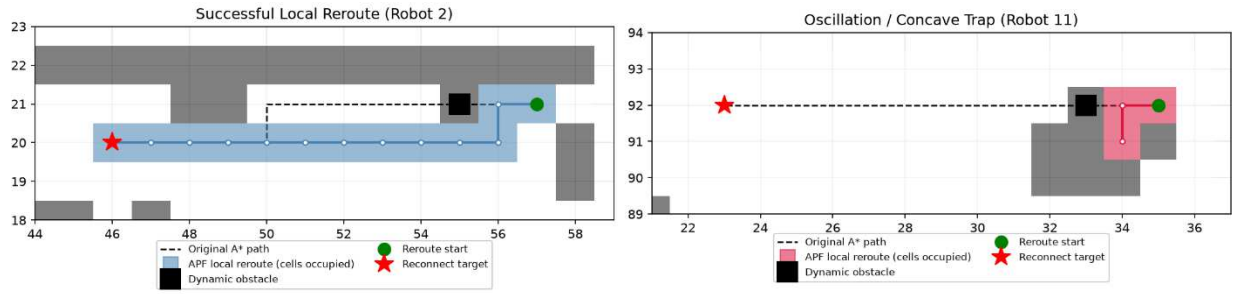


Figure 6 -- Left: a successful case; Right: a failure case,

In the favorable scenario, the impediment is located in a clear area, and the robot navigates past it with a brief, straightforward maneuver before seamlessly returning to the initial path. In the event of failure, the initiation point of the detour is constrained on three sides, providing just one slender exit that the local, greedy descent of the potential field fails to identify.

The figures near the stall point now indicate a genuine dead end rather than a two-cell back-and-forth cycle: at the stall point (91,34), only one neighboring cell is free -- (92,34), with potential 0.0085 -- and it is the cell the robot just came from. The other three neighbors are all blocked by the surrounding obstacle geometry. Continuing would mean immediately re-selecting a cell already visited, which the oscillation detector correctly flags as a stall rather than genuine progress, since the field retains no memory of having just left that cell.

The method of escalating to shape navigation rectifies 2 of the 4 N=15 failures by following the true grid boundary instead of relying on a continuous mathematical approximation: the robot maneuvers along the obstacle's perimeter in a consistent cardinal direction, employing the identical linear segment, stop-and-rotate technique as described in the

primary pipeline (Section 5.2). This eliminates the previous drift, as a linear, axis-aligned movement possesses no orthogonal direction available for drifting into:

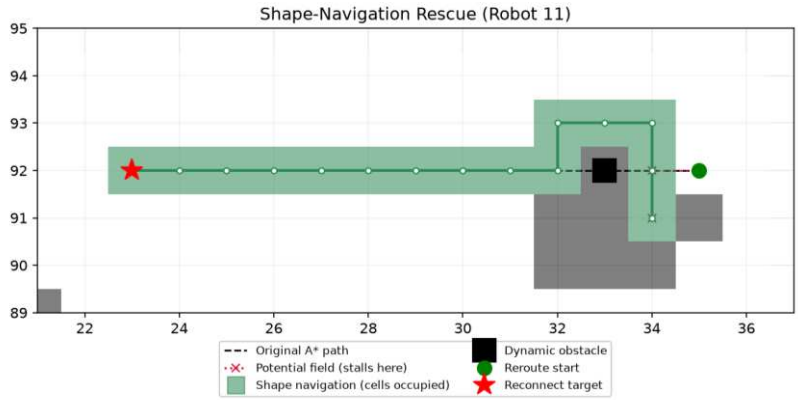


Figure 7 – Shape navigation rescue of APF failure.

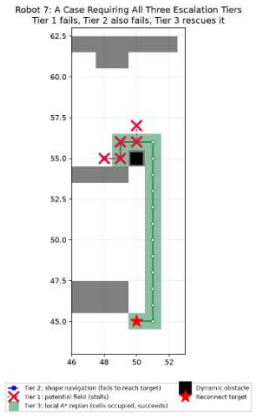
The illustrated example -- the same genuine concave-trap case shown failing in Figure 6 -- attains its objective in 3 turns, striking precisely at the reconnection point, one of the real cases resolved by the first two tiers alone.

A dynamic impediment was positioned at the midpoint of each genuine orientation-aware trajectory at N=25, and the comprehensive three-tier escalation—potential field, followed by grid-aligned shape navigation, and finally a constrained local A* replan upon both failures (Section 4.5)—was executed for every instance where the obstacle placement yielded a valid test.

Tier	N=15 result	N=25 result
Potential field (direct)	11/15 (73%)	21/25 (84%)
+ Shape navigation	13/15 (87%)	22/25 (88%)
+ Local A* replan (tier 3)	15/15 (100%)	25/25 (100%)

Table 6 -- Collision-avoidance escalation results, both team sizes

With N=15, the first two tiers are adequate for 13 of the 15 instances, but 2 of the 15 require a local A* replan. At N=25, three instances need this local A* -- each of the five tier-3 total cases across both team sizes succeeds.



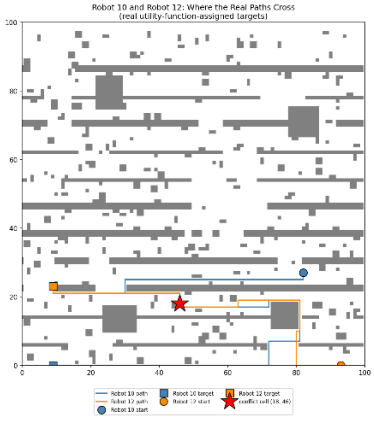
418

419 *Figure 8 -- One of the three N=25 cases genuinely requiring all three tiers.*

420 5.4 Negotiation

421 Negotiation addresses temporal conflicts arising when two robots reach the same cell at the same moment -- not merely
422 a shared cell, but one reached simultaneously -- as opposed to static obstacle collisions, which have been previously
423 managed. At N=25, examining all combinations of the 25 real orientation-aware pathways reveals five inherent timing
424 conflicts, which emerge due to the increased likelihood of a denser team having paths that intersect in both spatial and
425 temporal dimensions independently -- with N=15 also showing one such natural conflict.

426 The genuine trajectories of Robot 10 and Robot 12 intersect at coordinates (18,46), with both arriving
427 simultaneously:



428

429 *Figure 9 – Negotiation map view*

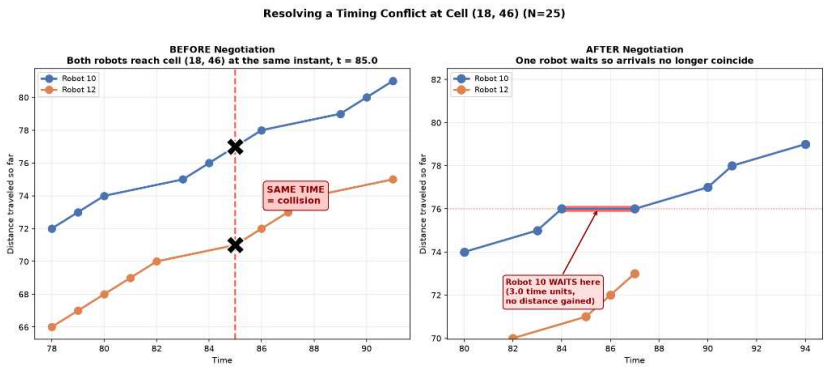


Figure 10 – Negotiation resolution

Robot 10 waited 3 time units so both robots avoided the collision and the deadlock it would otherwise have caused, at N=15. At N=25, all 5 of 5 natural conflicts were resolved the same way, in 1 to 4 passes each.

5.5 Integrated Multi-Scenario Results

The independent verification of Section 5.3's three-tier escalation and Section 5.4's negotiation is established above; this section affirms their proper composition when implemented concurrently on the same robot, across both team sizes.

At N=15, the two robots requiring the full three-tier escalation had their adjusted trajectories checked against every other robot's schedule directly: no new conflict was created by either detour. At N=25, the three robots requiring the full three-tier escalation (Table 6) were checked the same way: no new conflict was created by any of the detours either.

The underlying principle is procedural, not team-size-specific: any route change affects arrival timing, so conflicts must always be re-checked afterward.

Integrating every mechanism validated here -- target selection, orientation-aware planning, LQR tracking, the three-tier escalation, and negotiation -- across both team sizes yields 100% collision-avoidance success and 100% conflict resolution at N=15 and N=25 alike, differing only in how much escalation machinery each scale requires.

5.6 Single-Case Summary and Results

Robot 17, a real robot referenced throughout this paper and the most turn-heavy case among those tested, demonstrates the full impact of orientation-aware planning through a specific instance: identical starting and target points, completed by two distinct methodologies.

Metric	Baseline A*	Orientation-aware A*	Improvement
Turns	54	20	63.0% fewer
A* path cost (movement + turn penalty)	352.00	286.00	18.8% lower
Real LQR execution time	221.45s	163.15s	26.3% faster
Collisions	0	0	—
Final distance to target	0.0329 units	0.0329 units	both essentially exact

Table 7 -- Robot 17, same start and target, baseline vs. orientation-aware planning and execution.

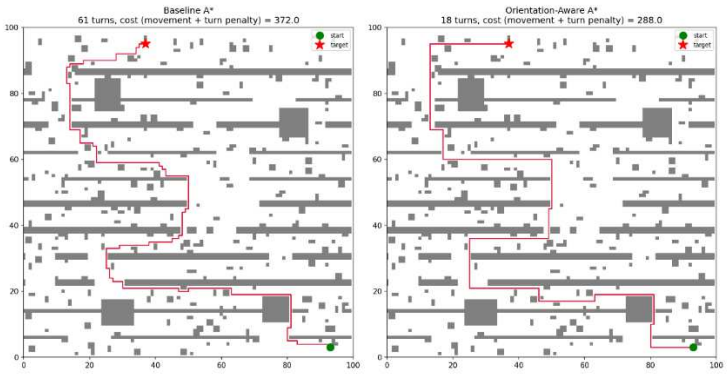


Figure 11 – Integrated case baseline vs orientation awareness.

A significant decrease in both turns and actual execution time, as the expense of turning greatly surpasses the geometric deviation required to circumvent it -- in this particular instance, the overall A* path cost diminishes by 18.8%, exceeding the 17.52%/14.94% aggregate reduction observed across all real paths in Section 5.2, aligning with Robot 17 being the most turn-intensive scenario in that aggregate. This individual instance aligns with the overall findings recorded for both team sizes throughout this paper, fully summarized here, using cost or computational-cost data whenever applicable:

Section	Result	Finding
5.1	Target spacing, proportional discount vs. hard exclusion	N=15: 7.63 -> 11.65 cells; N=25: 4.01 -> 9.40 cells
5.1	Priority weighting redirection	N=15: 3.40/15 -> 14.95/15; N=25: 6.30/25 -> 19.20/25
5.1	Duplicate-target assignments, N=5 to 30	0 duplicates at every team size tested

5.1	Multi-start vs. greedy assignment cost	N=15: 906.10 -> 860.50 (5.0% lower); N=25: 1263.80 -> 1190.00 (5.8% lower)
5.1	Multi-start vs. greedy, trials won	N=15: 10/10; N=25: 10/10
5.1	Multi-start computation-time cost	3x (N=5) growing to 89x (N=25), absolute time under 300ms throughout
5.1	Full utility function vs. naive baseline (profit)	N=15: +174.6%; N=25: +135.8% total profit
5.2	Turn count, real utility-derived targets	N=15: 64.85% reduction; N=25: 59.11% reduction
5.2	Real LQR execution time	N=15: 25.13% faster; N=25: 21.60% faster
5.2	A* path cost (movement + turn penalty)	N=15: 17.52% lower; N=25: 14.94% lower
5.2	Tracking accuracy and safety	final distance under 0.05 units, 0 collisions at either size
5.3	Potential-field local reroute success	N=15: 11/15 (73%); N=25: 21/25 (84%)
5.3	Remaining failures, shape-navigation rescue	N=15: 2/2 rescued; N=25: 1/4 rescued
5.3	Remaining failures, local A* replan (tier 3)	N=15: 2/2 rescued; N=25: 3/3 rescued -- 100% combined both sizes
5.4	Natural timing conflicts	N=15: 1 of 105 pairs; N=25: 5 of 300 pairs (all resolved)
5.5	Integrated escalation + negotiation, combined	N=15: 0 new conflicts from tier-3 detours; N=25: 0 new conflicts from tier-3 detours
5.6	Single-case demonstration (Robot 17)	63.0% fewer turns, 26.3% faster execution

Table 8 -- Complete summary of quantitative results across both team sizes, N=15 and N=25.

All outcomes in this table were directly assessed using authentic data from the identical connected pipeline for both team sizes -- utilizing the same robots, the same utility-function-selected targets, and the same planned trajectories -- instead of relying on distinct or idealized test scenarios, and each result was validated.

6. Conclusion and Future Work

This study introduced a decentralized multi-robot surveillance framework that integrates utility-driven target selection, orientation-sensitive A* planning, LQR trajectory tracking, a hierarchical collision-avoidance mechanism ranging from potential fields to grid-aligned boundary adherence, a constrained local replanning approach, and conflict resolution through negotiation, assessed with two separately created team sizes, N=15 and N=25.

The utility function identifies evenly distributed targets for team sizes ranging from 5 to 30, surpassing a simplistic baseline by a substantial margin at both scales, though less uniformly than earlier findings suggested, based on its optimized objective (174.6% / 135.8% increased profit). The potential field addresses the vast majority of dynamic-obstacle interactions independently, with its direct success rate actually rising with density on this map rather than falling (73% at N=15, 84% at N=25); shape navigation and, when needed, a constrained local A* replan account for the remainder, achieving 100% at both densities -- with the local A* replan proving necessary at both team sizes.

Negotiation addressed all issues examined at both scales, including natural conflicts that emerged at both team sizes. Orientation-aware A* reduces mean turns by 64.85% (N=15) / 59.11% (N=25), resulting in 25.13% / 21.60% quicker execution and a significantly reduced overall path cost according to its optimality framework, rather than just minimizing turns at an undisclosed expense elsewhere.

Two authentic flaws were identified and rectified. The LQR controller produced a real, verified collision when tracking continuously through a sharp turn following a long straight run, resolved by tracking each segment with full feedback control and executing turns as a discrete stop-and-rotate action. The closed-form trajectory of shape navigation exhibited unbounded lateral drift due to a misaligned boundary coordinate; two direct formula corrections worsened the issue, and the verified-safe alternative employs grid-aligned boundary tracing. Neither flaw was discernible from final-position checks alone; both required validating every intermediate trajectory point against the actual map, now the standard for this evaluation. A locally reactive approach reaching its endpoint is not adequate proof the trajectory taken was safe.

Future scope: Expanding validation to include other team sizes and map configurations and assessing the local A* replan fallback over more obstacle geometries beyond the five confirmed examples across both team sizes, is a sensible next step. Directly addressing the closed-form drift defect in shape navigation, rather than continuing to rely on the grid-aligned alternative, and formally characterizing that alternative as a general-purpose replacement, remain open. Future work also includes incorporating the CNN-driven density estimation from the original project scope, broadening local recovery to general star-world and sphere-world geometries beyond axis-aligned obstacles, characterizing runtime in relation to map dimensions, and conducting a direct comparison against prior methodologies.

Declarations

Funding: This research received no funding.

Clinical trial number: Not Applicable.

Ethics, Consent to Participate, and Consent to Publish declarations: Not Applicable.

Data Availability

The datasets during and/or analysed during the current study are available in the Zenodo repository, https://doi.org/10.5281/zenodo.23271558.

References

[1] Yamauchi B (1997) A frontier-based approach for autonomous exploration. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp 146-151

[2] Burgard W, Moors M, Fox D, Simmons R, Thrun S (2000) Collaborative multi-robot exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), vol 1, pp 476-481

[3] Butzke J, Likhachev M (2011) Planning for multi-robot exploration with multiple objective utility functions. In: Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3254-3259

[4] Quintero CG, Onate Lopez J, Bertel FA (2012) Intelligent exploration and surveillance algorithms for multi-agents robotics systems. In: Proceedings of the XXXVIII Conferencia Latinoamericana en Informatica (CLEI), pp 1-10

[5] Lin HY, Huang YC (2021) Collaborative complete coverage and path planning for multi-robot exploration. Sensors 21(11):3709

[6] Yao W, Chen Y, Fu J, Qu D, Wu C, Liu J, Sun G, Xin L (2022) Evolutionary utility prediction matrix-based mission planning for unmanned aerial vehicles in complex urban environments. IEEE Transactions on Intelligent Vehicles

[7] Ahmed MF, Fremont V, Fantoni I (2023) Active SLAM utility function exploiting path entropy. In: Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)

[8] Krupke DM (2023) Near-optimal coverage path planning with turn costs. arXiv:2310.20340

[9] Kobayashi K, Kobayashi S, Ueno S, Higuchi T (2025) Distributed algorithm with emergent area partitioning and base station's situation awareness for multi-robot patrolling. Journal of Robotics and Mechatronics 37(4):927-944

[10] Lee K, Kim H, Li J, Park J (2026) Priority-aware multi-robot coverage path planning. IEEE Robotics and Automation Letters

[11] Shen Z, Gupta S, Zhao S, Zhou D, Wang G, Ren Z, Ou Y, Zhai Y, Chen CLP (2026) Coverage path planning: classical foundations, recent advances, and future directions. arXiv:2607.10649

[12] Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 4(2):100-107

[13] Wang B (2021) Path planning of mobile robot based on A* algorithm. In: Proceedings of the 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI), pp 524-528

[14] Xiang D, Lin H, Ouyang J, Huang D (2022) Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. Scientific Reports 12:13273

[15] Cao B et al (2023) Research on the A* algorithm for safe path planning. In: Proceedings of the 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), pp 105-109

[16] Liu N, Ma C, Hu Z, Guo P, Ge Y, Tian M (2024) Workshop AGV path planning based on improved A* algorithm. Mathematical Biosciences and Engineering 21(2):2137-2162

[17] Wang T, Li A, Guo D, Du G, He W (2024) Global dynamic path planning of AGV based on fusion of improved A* algorithm and dynamic window method. Sensors 24(6):2011

[18] Din MN, Ushigome S, Fujimoto Y (2026) Path finding for autonomous vehicles by Orientation-Aware Advanced: A* algorithm in a grid map. International Journal of Intelligent Robotics and Applications

[19] Li P, Li Y, Dai X (2024) VNS-BA*: an improved bidirectional A* path planning algorithm based on variable neighborhood search. Sensors 24(21):6929

[20] Wang H, Lou S, Jing J, Wang Y, Liu W, Liu T (2022) The EBS-A* algorithm: an improved A* algorithm for path planning. PLOS ONE 17(2):e0263841

[21] Rimon E, Koditschek DE (1992) Exact robot navigation using artificial potential functions. IEEE Transactions on Robotics and Automation 8(5):501-518

[22] Hsieh MA, Loizou S, Kumar V (2007) Stabilization of multiple robots on stable orbits via local sensing. In: Proceedings of the 2007 IEEE International Conference on Robotics and Automation, pp 2312-2317

[23] Tandogan T (2015) Collision-free indoor navigation using an artificial potential field controller. Dissertation, [University]

[24] Vasilopoulos V, Koditschek DE (2018) Reactive navigation in partially known non-convex environments. In: International Workshop on the Algorithmic Foundations of Robotics (WAFR), Springer, pp 406-421

[25] Sawant M, Polushin I, Tayebi A (2024) Hybrid feedback control design for non-convex obstacle avoidance. IEEE Transactions on Automatic Control (exact volume/issue/pages not confirmed -- verify against the final published version; arXiv:2304.10598)

[26] Kim J, Park S, Lee W, Kim W, Doh N, Nam C (2024) Escaping local minima: hybrid artificial potential field with wall-follower for decentralized multi-robot navigation. arXiv:2409.10332

[27] Woo H, Hirota S (2025) Local minimum avoidance in potential field method through virtual-wall generation and wall following. Journal of Robotics and Mechatronics 37(5):1094

[28] Sharon G, Stern R, Felner A, Sturtevant NR (2015) Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence 219:40-66

[29] van den Berg J, Guy SJ, Lin M, Manocha D (2011) Reciprocal n-body collision avoidance. In: Robotics Research: The 14th International Symposium ISRR, Springer Tracts in Advanced Robotics vol 70, pp 3-19

[30] Wu W, Bhattacharya S, Prorok A (2020) Multi-robot path deconfliction through prioritization by path prospects. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)

[31] Tang J, Mao Z, Ma H (2025) Large-scale multirobot coverage path planning on grids with path deconfliction. arXiv:2411.01707

[32] Xuan DT, Hung NT, Thang VT (2025) A comprehensive review of improved A* path planning algorithms and their hybrid integrations. Automation 6(4):52

[33] Chandra R, Zinage V, Bakolas E, Stone P, Biswas J (2025) Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions. Autonomous Robots 49(2):12