Expert Systems With Applications 333 (2027) 133787

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Graph-driven multi-robot coordination with situational awareness for hospital service robotics

Tingjun Lei [a,*], Samuel Steen [a], Chaomin Luo [b], Jun Chen [c], Yaqun Yuan [d]

[a] School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND, 58202, USA
[b] Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, 39762, USA
[c] Department of Electrical and Computer Engineering, Oakland University, Rochester, MI, 48309, USA
[d] Department of Epidemiology and Biostatistics, Michigan State University, East Lansing, MI, 48824, USA

ARTICLE INFO

Keywords:
Multi-robot coordination
Hospital service robotics
CNN-based LiDAR detection
Self-organizing map
Bio-inspired path optimization

ABSTRACT

Hospital service robots must coordinate time-sensitive and heterogeneous tasks while maintaining reliable awareness of nearby robots in structured, human-populated clinical spaces where GPS, wireless communication, and global localization can be degraded. Existing studies typically address perception, task allocation, or path optimization as separate components, which limits their ability to support closed-loop multi-robot coordination under communication-constrained hospital conditions. This paper presents a graph-driven multi-robot coordination framework with situational awareness for hospital service robotics. The framework couples three coordination modules on a shared MAKLINK Graph Theory (MGT) representation of the hospital layout. First, a lightweight CNN-LiDAR module performs point-wise binary segmentation of 2D LiDAR scans to detect peer robots and maintain local fleet awareness during GPS or network outages. Second, the detected peer positions and confidence scores are passed to a priority-weighted Self-Organizing Map (SOM), which allocates asynchronous service tasks according to task urgency, deadline, risk, proximity, and workload balance, with Lazy Theta* generating feasible any-angle initial routes on the MGT graph inside the allocation loop to link assignment decisions to executable navigation. Finally, an improved Moss Growth Optimization (iMGO) module refines the initial routes by optimizing waypoint locations on a compact line-of-sight-preserving MAKLINK domain. Simulation and comparison studies validate the perception, allocation, path-refinement, and integrated pipeline components. The CNN-LiDAR subsystem sustains reliable peer-robot detection during GPS and communication outages, with near-perfect accuracy at the close range most critical for collision avoidance. The priority-weighted SOM improves task-completion behavior and workload balance compared with clustering and non-clustering genetic algorithm baselines. The iMGO module produces shorter and smoother collision-free paths with faster convergence than competing optimizers. The integrated multi-robot simulations further show reduced mission time and travel distance relative to random allocation, supporting the proposed framework as a scalable and safety-aware coordination architecture for hospital service robotics.

1. Introduction

The growing deployment of autonomous service robots in healthcare environments, including hospitals, eldercare facilities, and infectious-disease wards, has introduced new demands for intelligent coordination systems capable of operating safely and efficiently in dynamic, human-populated spaces (Bernhard et al., 2024; Jayaraman et al., 2021; Lei et al., 2021; Sellers et al., 2024, 2026; Short et al., 2023). Unlike industrial or outdoor settings, hospital environments are characterized by constrained corridor layouts, safety-critical interactions with patients and clinical staff, and continuously changing operational conditions such as asynchronous task arrivals, dynamic obstacles, and unreliable communication infrastructure (Devanga et al., 2025; Hicks et al., 2025a,c; Kyrarini et al., 2021; Lei et al., 2022; Liu et al., 2023; Short et al., 2025). These characteristics collectively demand that multi-robot systems jointly address three tightly coupled challenges: *situational awareness* under degraded sensing conditions, *adaptive task allocation* under dynamic and heterogeneous service demands, and *safe and efficient navigation* within structured indoor spaces.

A particular vulnerability arises when hospital robots rely on centralized GPS or wireless communication for mutual localization. Dense medical equipment, reinforced concrete structures, and electromagnetic

* Corresponding author.
E-mail address: Tingjun.Lei@UND.edu (T. Lei).

https://doi.org/10.1016/j.eswa.2026.133787
Received 10 May 2026; Received in revised form 1 July 2026; Accepted 27 July 2026

interference from clinical devices routinely disrupt these signals, leaving individual robots unaware of the positions of their fleet peers and increasing the risk of collision (Rogers et al., 2024; Steen et al., 2025a; Zhong et al., 2024). This challenge is especially critical in scenarios such as infectious disease isolation wards or high-density logistics corridors, where even brief coordination failures may compromise patient safety or disrupt clinical workflows. Light Detection and Ranging (LiDAR) technology offers a robust solution by providing direct, infrastructure-independent spatial sensing, and recent advances in Convolutional Neural Networks (CNNs) have demonstrated that LiDAR-based binary segmentation can reliably detect peer robots in real time on resource-constrained platforms (Rogers et al., 2023; Swapna & Murali Mohan, 2024; Xie et al., 2023). However, existing LiDAR-based approaches address robot detection in isolation, without integrating the resulting situational awareness into a broader multi-robot coordination and path planning pipeline.

Multi-robot task allocation and path planning have long been central problems in autonomous systems, with applications spanning search and rescue, logistics, and healthcare (Chakraa et al., 2023; Jan et al., 2024; Kong et al., 2024; Lei et al., 2024; Riser et al., 2024; Sellers et al., 2023). Market-based strategies, classical combinatorial optimization, and bio-inspired or learning-based methods have all been explored for task allocation (Alarabi et al., 2024; Cui et al., 2024; Lin et al., 2024). Among these, Self-Organizing Maps (SOMs), originally proposed by Kohonen (1990), have demonstrated particular effectiveness for representing spatial task distributions and enabling distributed, adaptive robot coordination without requiring global communication (Black et al., 2024; Faigl & Hollinger, 2018). The SOM's competitive learning mechanism allows robots to self-organize around task clusters and dynamically reassign targets in response to changing operational conditions, making it well-suited to the asynchronous and priority-driven nature of hospital service tasks.

Parallel to task allocation, path planning in constrained indoor environments remains a critical challenge. Grid-based planners, any-angle planners such as Theta* and Lazy Theta*, and sampling-based methods (Chintam et al., 2024; Cruz et al., 2025; Hicks et al., 2025b; Nash et al., 2010) can generate feasible paths, but they often require additional refinement or replanning mechanisms when dynamic obstacles, robot-robot interactions, and real-time environmental changes are present. To overcome these limitations, bio-inspired optimization methods have been increasingly employed for global path refinement (Li et al., 2023; Liu et al., 2023; Wu et al., 2023). Among these, the recently proposed Moss Growth Optimization (MGO) (Zheng et al., 2024) has shown strong capability in balancing exploration and exploitation, emulating the decentralized and adaptive growth patterns of natural moss. When applied on a topological abstraction such as MAKLINK Graph Theory (MGT) (Lei et al., 2023; Yang et al., 2020), MGO enables efficient trajectory refinement with line-of-sight pruning, achieving smooth and collision-free paths even in cluttered indoor layouts (Steen et al., 2025b). Despite these advances, existing studies on bio-inspired path refinement have not yet been integrated with real-time perception and adaptive task allocation into a unified hospital-oriented multi-robot framework. Healthcare environments present structural characteristics that make them particularly suitable for graph-based modeling. Hospital corridors, patient rooms, and service zones follow semi-regular topological layouts that are well-captured by the MAKLINK representation, while the inherently dynamic and priority-driven nature of medical service tasks motivates the use of adaptive neural allocation mechanisms (Choubey et al., 2025; Liu et al., 2025; London et al., 2025a,b). Furthermore, the frequent disruption of communication infrastructure in clinical settings underscores the need for onboard perception capabilities that can maintain fleet-level situational awareness without relying on external positioning systems.

Building on these insights, this paper presents an integrated bio-inspired coordination framework for multi-robot hospital service systems. The proposed system comprises three tightly coupled modules that form a complete perception-to-execution pipeline, as illustrated in Fig. 1. First, a lightweight CNN performs binary segmentation of 2D LiDAR scans to detect peer robots and maintain situational awareness during GPS and communication outages. The resulting real-time positional estimates are fed directly into a priority-weighted SOM-based task allocation module, which dynamically distributes service tasks among robots based on spatial context, task urgency, workload balance, and the current environmental state provided by the perception layer. Finally, initial routes generated by Lazy Theta* on an MGT representation of the hospital layout are refined through an improved Moss Growth Optimization algorithm (iMGO), which incorporates elite-guided adaptive annealing and fleet-level warm-start initialization to accelerate convergence and improve trajectory quality in multi-robot scenarios. The main contributions of this work are as follows:

(1) An integrated bio-inspired coordination framework is proposed for multi-robot hospital service systems. The framework enables autonomous hospital service robots to maintain fleet-level awareness, adaptively allocate service tasks, and navigate safely and efficiently in dynamic clinical environments, without reliance on external positioning infrastructure or centralized communication.
(2) A CNN-LiDAR situational awareness module is proposed for fail-safe peer robot detection during GPS and communication outages in hospital environments. The module employs a lightweight five-layer binary segmentation architecture that classifies each LiDAR return as robot or background in real time, providing continuous fleet awareness without reliance on external positioning infrastructure. Its output directly feeds the task allocation layer, establishing the perception-to-coordination interface of the proposed pipeline.
(3) A priority-weighted SOM-based task allocation mechanism is developed for decentralized multi-robot coordination under limited inter-robot communication. The SOM integrates multiple contextual factors, including task urgency, deadline proximity, robot workload, and real-time environmental state from the CNN-LiDAR module, enabling robots to competitively and adaptively acquire service targets while maintaining balanced fleet-level workloads without explicit centralized control.
(4) An improved Moss Growth Optimization algorithm (iMGO) is proposed for global path refinement on the MGT substrate. Building upon the original MGO framework, iMGO introduces elite-guided adaptive step annealing, which dynamically adjusts the optimization schedule based on solution improvement rates, and fleet-level warm-start initialization, which leverages optimized paths from neighboring robots to accelerate convergence in multi-robot scenarios. Combined with edge-based line-of-sight pruning, iMGO consistently produces shorter, smoother, and collision-free trajectories with faster convergence than standard MGO and competing baselines.

The remainder of this paper is organized as follows. Section 2 introduces the hospital environment modeling process using MAKLINK Graph Theory. Section 3 presents the CNN-LiDAR situational awareness module for fail-safe peer robot detection. Section 4 describes how perception outputs are converted into priority-weighted SOM task-allocation and Lazy Theta* initial routing decisions. Section 5 develops the iMGO path refinement algorithm. Section 6 provides simulation and comparison studies validating each module and the integrated system. Section 7 concludes the paper and outlines directions for future research.

2. Environment modeling via MAKLINK graph theory

Accurate and efficient environment modeling is the foundation of autonomous navigation and multi-robot coordination, particularly in structured and safety-critical domains such as hospitals. Healthcare facilities are typically composed of long corridors, intersecting hallways, patient rooms, and restricted service areas, resulting in semi-regular but constrained geometric layouts. These spaces are further complicated by dynamic obstacles such as mobile equipment, clinical staff, and patients.

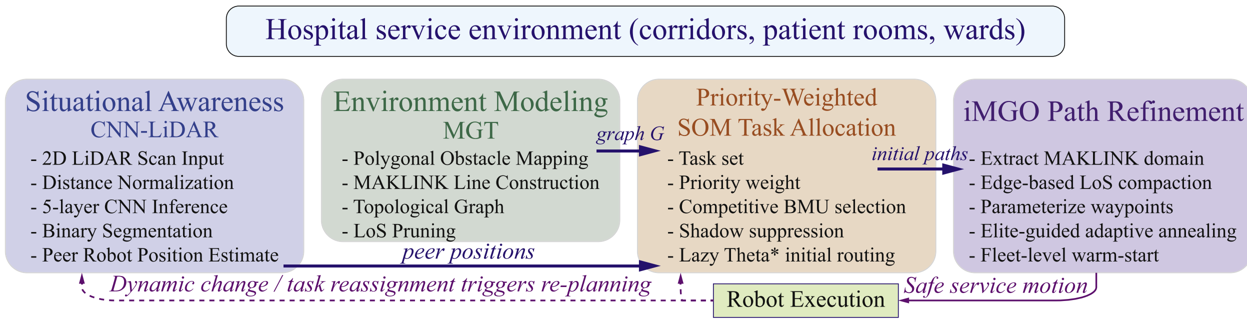


Fig. 1. Overall workflow of the proposed graph-driven multi-robot coordination framework for hospital service environments. The figure distinguishes the top-level functional blocks of situational awareness, MGT environment modeling, SOM-based allocation, iMGO path refinement, and robot execution, while integrating the major technical operations directly inside each corresponding block. Solid arrows indicate the forward perception-modeling-allocation-routing-refinement data flow, and the dashed arrow indicates closed-loop re-planning triggered by dynamic environmental changes or task reassignments.

Developing a model that captures the fixed spatial topology while supporting real-time navigation updates is therefore essential for reliable hospital service robotics. In this work, the hospital environment is represented using *MAKLINK Graph Theory (MGT)*, a graph-based topological abstraction that enables efficient path computation and spatial reasoning. Compared with grid-based occupancy maps that discretize the entire workspace at high computational cost, MGT directly encodes the visible free-space relationships between obstacle boundaries, preserving topological fidelity while avoiding redundant sampling. This makes MGT particularly well-suited to the corridor-dominated geometry of clinical spaces, where the fixed layout can be captured compactly as a connectivity graph.

The hospital floor plan is treated as a two-dimensional (2D) polygonal workspace $\mathcal{W} \subset \mathbb{R}^2$. Static structures such as walls, beds, cabinets, and partitions are modeled as a set of polygonal obstacles $\mathcal{O} = \{O_1, O_2, \dots, O_k\}$, while corridors and open areas constitute the free space $\mathcal{W}_{\text{free}} = \mathcal{W} \setminus \bigcup_{i=1}^{k} O_i$. To maintain safe navigation margins, each obstacle O_i is expanded outward by a collision buffer δ_r that accounts for the robot's body radius and sensing uncertainty, yielding an inflated obstacle set $\mathcal{O}^{\delta}$. Each service robot is modeled as a point mass, allowing efficient geometric reasoning without loss of collision safety.

MAKLINK connections are straight line segments that link vertices of different obstacles, or extend perpendicularly from an obstacle vertex to the environment boundary, subject to the condition that they do not intersect any buffered obstacle in $\mathcal{O}^{\delta}$. Each such line represents a potential collision-free passage through the free space. Let $\mathcal{M} = \{M_1, M_2, \dots, M_p\}$ denote the set of all free MAKLINK lines constructed in $\mathcal{W}_{\text{free}}$. The topological graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is then constructed as follows. The midpoint of each free MAKLINK line M_i, together with the designated start position $\mathbf{q}_s$ and goal position $\mathbf{q}_g$ of each robot, are defined as graph nodes:

$$\mathcal{V} = \left\{ \mathbf{v}_i = \tfrac{1}{2}\left(\mathbf{e}_i^1 + \mathbf{e}_i^2\right) \;\middle|\; M_i \in \mathcal{M} \right\} \cup \{\mathbf{q}_s, \mathbf{q}_g\}, \tag{1}$$

where $\mathbf{e}_i^1$ and $\mathbf{e}_i^2$ are the two endpoints of line M_i. An undirected edge $(\mathbf{v}_i, \mathbf{v}_j)$ is added to $\mathcal{E}$ if nodes $\mathbf{v}_i$ and $\mathbf{v}_j$ lie on mutually visible MAKLINK lines, i.e., the segment $\overline{\mathbf{v}_i\mathbf{v}_j}$ does not intersect any buffered obstacle. Each edge is weighted by the Euclidean distance between its endpoints:

$$w(\mathbf{v}_i, \mathbf{v}_j) = \|\mathbf{v}_i - \mathbf{v}_j\|_2. \tag{2}$$

The resulting graph $\mathcal{G}$ is a concise yet expressive representation of the environment's topological connectivity, and supports both global route generation via graph search and local re-optimization when dynamic changes occur.

An illustrative example of the constructed model is shown in Fig. 2. The blue dashed lines represent the free MAKLINK connections among polygonal obstacles, while orange dots denote the midpoints used as graph nodes. The dashed green lines depict the collision-free topological network derived from $\mathcal{G}$, which forms the navigation backbone for subsequent task allocation and path refinement.

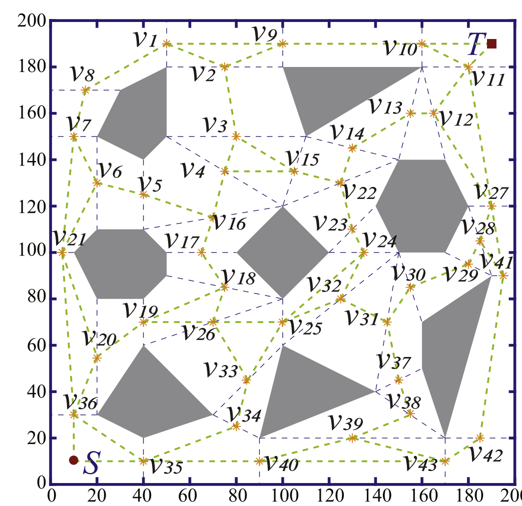


Fig. 2. Illustration of polygonal obstacles and free MAKLINK lines in a hospital floor plan. Blue dashed lines indicate free MAKLINK connections, orange dots denote line midpoints used as graph nodes $\mathcal{V}$, and green dashed lines form the collision-free topological graph $\mathcal{G}$ derived from MGT.

The graph $\mathcal{G}$ serves as the shared geometric substrate for all downstream modules in the proposed framework. In the priority-weighted SOM task allocation module (Section 4), $\mathcal{G}$ provides the topological connectivity used by Lazy Theta* to generate initial collision-free routes between robots and their assigned targets. In the iMGO path refinement module (Section 5), $\mathcal{G}$ defines the set of free MAKLINK lines from which the refinement domain is extracted, and the edge-based line-of-sight compaction operates directly on the endpoint geometry of $\mathcal{M}$. By providing a unified, lightweight topological representation, MGT decouples the geometric structure of the hospital environment from the higher-level coordination and optimization logic, enabling each module to reason efficiently over the same consistent map.

3. CNN-LiDAR based situational awareness

In hospital environments, autonomous service robots frequently operate in areas where GPS signals are unreliable or completely unavailable. Dense reinforced concrete structures, electromagnetic interference from medical equipment such as MRI machines and surgical instruments, and network congestion during peak clinical operations can all degrade or disrupt the communication infrastructure that robots rely on for mutual localization (Steen et al., 2025a; Zhong et al., 2024). When such outages occur, individual robots lose awareness of their fleet peers, increasing the risk of collision in narrow hospital corridors and creating hazardous conditions in safety-critical zones such as operating suites and isolation wards. This section presents a fail-safe situational awareness module based on 2D LiDAR sensing and a lightweight Convolutional Neural Network (CNN). The module operates independently of GPS and centralized communication, enabling each robot to continuously detect the positions of nearby fleet members using only its onboard sensor. The resulting peer position estimates are passed directly to the priority-weighted SOM task allocation module (Section 4), providing real-time environmental state information that informs competitive task assignment and dynamic route adjustment.

3.1. Problem formulation

Each hospital service robot carries a 2D LiDAR sensor providing 360-degree coverage with angular resolution $\Delta\alpha$. At each time step, the sensor produces N distance measurements

$$\mathbf{d} = [d_1, d_2, \ldots, d_N]^T, \tag{3}$$

collected at uniform angular increments over $[0, 2\pi)$. Each measurement d_k records the range to the nearest surface along ray k:

$$d_k = \min\left(d_{\mathrm{wall}}^{(k)}, \min_{i\in\mathcal{R}\setminus R_{\mathrm{self}}} \widetilde{d}_{\mathrm{robot}_i}^{(k)}, r_{max}\right), \tag{4}$$

where $d_{\mathrm{wall}}^{(k)}$ is the distance from ray k to the nearest static surface (wall, door, or medical cabinet), $r_{\max}$ is the maximum sensor range, and $\widetilde{d}_{\mathrm{robot}_i}^{(k)}$ is defined as

$$\widetilde{d}_{\mathrm{robot}_i}^{(k)} = \begin{cases} d_{\mathrm{robot}_i}^{(k)}, & \text{if ray } k \text{ intersects the footprint of peer robot } i, \\ +\infty, & \text{otherwise.} \end{cases} \tag{5}$$

Thus, a peer robot contributes to the range minimum only when it is actually intersected by the corresponding LiDAR ray, avoiding ambiguity in the range measurement model.

The detection problem is formulated as a *point-wise binary classification* task. Given $\mathbf{d}$, the objective is to predict a label vector $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$ where:

$$y_i = \begin{cases} 1 & \text{if ray } i \text{ intersects a peer robot,} \\ 0 & \text{if ray } i \text{ hits a static surface or reaches } r_{max}. \end{cases} \tag{6}$$

Mobile robots create characteristic localized disruptions in the otherwise smooth distance profiles produced by static hospital walls and fixtures. The CNN is trained to recognize these signatures and distinguish them from background returns, enabling reliable detection without hand-crafted geometric features or environment-specific parameter tuning.

3.2. CNN architecture

Raw distance measurements are first normalized to $[0, 1]$ for network stability:

$$\mathbf{x} = \frac{\mathbf{d}}{r_{max}}, \tag{7}$$

ensuring consistent input scaling regardless of sensor hardware specifications.

The detection network is a five-layer 1D CNN that processes the sequential LiDAR scan by exploiting the spatial continuity of laser measurements. The convolutional operation at layer l is:

$$z_j^{(l)} = \sum_{i=1}^{C_{l-1}} \mathbf{W}_{ij}^{(l)} * \mathbf{x}_i^{(l-1)} + b_j^{(l)}, \tag{8}$$

where $\mathbf{W}_{ij}^{(l)}$ is the convolutional kernel connecting input channel i to output channel j, $*$ denotes the 1D convolution operation, C_{l-1} is the number of input channels, and $b_j^{(l)}$ is the bias term.

The five layers are designed with progressively widening receptive fields to capture multi-scale spatial features. Conv1 uses a $k_1 \times 1$ kernel with $C_1 = 64$ filters to detect local point-level discontinuities indicative of object boundaries. Conv2 uses a $k_2 \times 1$ kernel with $C_2 = 128$ filters to identify broader patterns spanning multiple scan points. Conv3 uses a $k_3 \times 1$ kernel with $C_3 = 128$ filters sized to match the angular footprint of a typical hospital service robot, enabling direct detection of robot-scale features. Conv4 uses a $k_4 \times 1$ kernel with $C_4 = 64$ filters to refine detections by incorporating wider contextual information and suppressing false positives. Conv5 applies a 1×1 kernel with a single filter to produce the per-point binary output via a learned linear combination of the multi-scale features.

Each convolutional layer incorporates batch normalization for training stability:

$$\hat{z}_j^{(l)} = \gamma^{(l)} \frac{z_j^{(l)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta^{(l)}, \tag{9}$$

where μ_B and σ_B^2 are the batch mean and variance, $\gamma^{(l)}$ and $\beta^{(l)}$ are learned scale and shift parameters, and ϵ prevents numerical instability. ReLU activations introduce nonlinearity after each batch normalization step:

$$a_j^{(l)} = \max\left(0, \hat{z}_j^{(l)}\right). \tag{10}$$

Dropout layers with probability p are inserted after Conv2 and Conv4 to prevent overfitting to specific hospital layout patterns. The final layer applies a sigmoid activation to produce per-point detection probabilities:

$$\hat{y}_i = \sigma\left(z_i^{(5)}\right) = \frac{1}{1 + e^{-z_i^{(5)}}}. \tag{11}$$

3.3. Training

Network training minimizes the point-wise binary cross-entropy loss over all N LiDAR measurements:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right]. \tag{12}$$

Parameters are updated using the Adam optimizer with adaptive learning rates:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{13}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected first and second moment estimates of the gradient, and α is the learning rate. A geometric decay schedule reduces the learning rate every epoch:

$$\alpha(e) = \alpha_0 \cdot \gamma^e, \tag{14}$$

where e is the current epoch, α_0 is the initial rate, and γ is the decay factor, promoting stable convergence without manual intervention.

Training data are generated through simulation in hospital-scale environments with varying corridor layouts, room configurations, and robot densities, ensuring robustness across the range of clinical spaces encountered in practice without requiring costly real-world data collection. The simulation produces labeled datasets that capture the full range of robot-to-robot interactions including partial occlusions from medical equipment and trolleys, varying inter-robot distances, and simultaneous multi-robot encounters.

3.4. Real-time detection pipeline

The trained CNN is integrated into a six-stage real-time detection pipeline, illustrated in Fig. 3. Algorithm 1 presents the complete procedure.

Stage 1 – Distance normalization. Raw measurements $\mathbf{d}$ are scaled to $[0, 1]$ via Eq. (7).

Stage 2 – CNN inference. The normalized input $\mathbf{x}$ is passed through the five-layer network to produce probability estimates $\hat{\mathbf{y}} \in [0, 1]^N$.

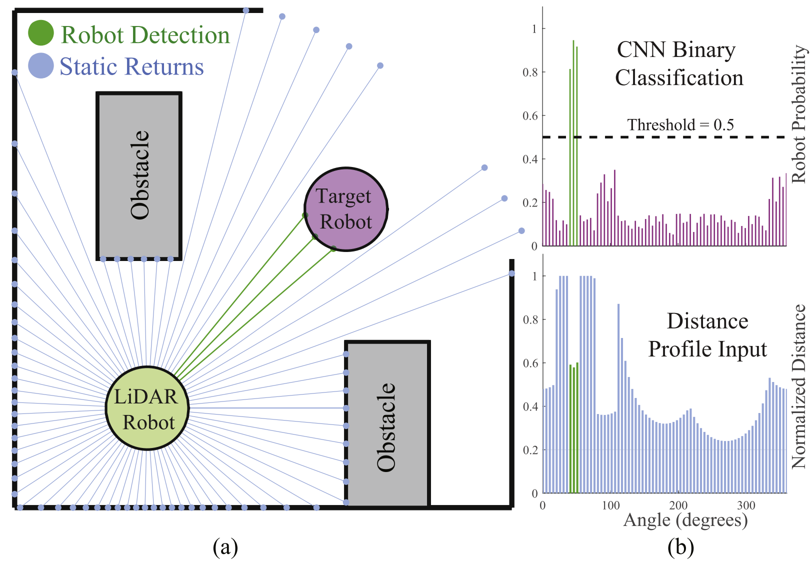


Fig. 3. Binary segmentation process for peer robot detection in a hospital environment. (a) The sensing robot (cyan) emits 360-degree LiDAR rays; blue rays indicate static returns from walls and fixtures while green rays identify a detected peer robot (magenta). (b) The CNN processing pipeline: the distance profile input (bottom) shows normalized scan measurements and the binary output (top) shows per-point detection probabilities, with values above the 0.5 threshold (dashed line) indicating robot detections.

Stage 3 – Thresholding. Robot-indicative points are extracted as $\mathcal{I}_{\text{robot}} = \{k : \hat{y}_k > \tau\}$, where $\tau = 0.5$ is the detection threshold.

Stage 4 – Position estimation. The angular center of the detected cluster is computed as

$$k_{\text{center}} = \text{round}\big(\text{mean}(\mathcal{I}_{\text{robot}})\big), \tag{15}$$

giving the local bearing $\alpha_{\text{local}} = (k_{\text{center}} - 1) \times 2\pi/N$. The peer robot position in global coordinates is then:

$$\hat{\mathbf{p}}_{\text{det}} = \mathbf{p}_{\text{self}} + d_{\text{robot}} \begin{bmatrix} \cos(\theta + \alpha_{\text{local}}) \\ \sin(\theta + \alpha_{\text{local}}) \end{bmatrix}, \tag{16}$$

where $\mathbf{p}_{\text{self}}$ is the sensing robot's current position, $d_{\text{robot}} = \mathbf{d}[k_{\text{center}}]$ is the measured range, and θ is the robot's heading.

Stage 5 – Confidence estimation. A detection confidence score is computed as the mean probability over detected points:

$$c = \frac{1}{|\mathcal{I}_{\text{robot}}|} \sum_{k \in \mathcal{I}_{\text{robot}}} \hat{y}_k. \tag{17}$$

Higher confidence scores indicate strong consensus across multiple LiDAR returns, while lower scores signal potential occlusion or extreme range conditions. These scores are passed downstream to inform the reliability weighting in the task allocation module.

Stage 6 – Mode selection. When GPS is available, the LiDAR-derived estimate is fused with the GPS position for improved accuracy. When GPS is degraded or absent, the system operates autonomously in LiDAR-only mode, maintaining fleet awareness without interruption.

Algorithm 1: CNN LiDAR-based peer robot detection.

Input : CNN model; LiDAR scan $\mathbf{d}$; robot heading θ; robot position $\mathbf{p}_{\text{self}}$; GPS status
Output: Detected peer position $\hat{\mathbf{p}}_{\text{det}}$; confidence c

$\mathbf{x} \leftarrow \mathbf{d}/r_{\max}$ // Normalize to $[0, 1]$
$\hat{\mathbf{y}} \leftarrow$ `CNN.forward`$(\mathbf{x})$ // Inference
$\mathcal{I}_{\text{robot}} \leftarrow \{k : \hat{y}_k > \tau\}$ // Threshold
if $|\mathcal{I}_{robot}| = 0$ **then**
  return $(\varnothing,\ 0)$ // No peer detected
$k_c \leftarrow \text{round}(\text{mean}(\mathcal{I}_{\text{robot}}))$
$\alpha_{\text{local}} \leftarrow (k_c - 1) \times 2\pi/N$
Compute $\hat{\mathbf{p}}_{\text{det}}$ via Eq. (16)
$c \leftarrow \frac{1}{|\mathcal{I}_{\text{robot}}|} \sum_{k \in \mathcal{I}_{\text{robot}}} \hat{y}_k$ // Confidence
if GPS available **then**
  $\hat{\mathbf{p}}_{\text{final}} \leftarrow$ `FuseGPS`$(\hat{\mathbf{p}}_{\text{det}}, \mathbf{p}_{\text{GPS}})$
else
  $\hat{\mathbf{p}}_{\text{final}} \leftarrow \hat{\mathbf{p}}_{\text{det}}$
return $(\hat{\mathbf{p}}_{\text{final}},\ c)$

4. Priority-weighted SOM-based task allocation

Efficient and adaptive task allocation is critical for coordinating multiple hospital service robots operating in constrained and dynamic clinical environments. Service tasks such as medication delivery, specimen transport, disinfection, and patient-assistance requests emerge asynchronously, vary widely in urgency and deadline sensitivity, and must be coordinated under limited inter-robot communication. This section explicitly links the perception output of Section 3 to the task-allocation and routing decisions used by the proposed coordination layer. Section 4.1 first defines how CNN-LiDAR peer-position estimates enter the SOM competition. Section 4.2 then introduces the SOM lattice, Section 4.3 develops the priority-weighted competitive rule, and Section 4.4 explains why Lazy Theta* is used to convert tentative assignments into feasible initial routes. Lazy Theta* operates inside the allocation loop to evaluate travel costs, ETAs, and shadow-suppression decisions.

4.1. Perception-to-allocation interface

The output of Algorithm 1 feeds directly into the priority-weighted SOM task allocation module. Specifically, at each coordination cycle, each robot R_i broadcasts its own position when communication is available and aggregates the peer position estimates $\{\hat{\mathbf{p}}_{\text{det}}^{(j)}\}$ from its onboard detector. These estimates update two quantities in the SOM module: (i) the proximity factor $\phi_{\text{prox}}(T_j, t)$ in the priority weight $A_j(t)$, which reflects the distance from the nearest detected robot to each task location; and (ii) the workload term $v_i(t)$ in the competitive cost function, which accounts for the current path lengths of detected fleet members.

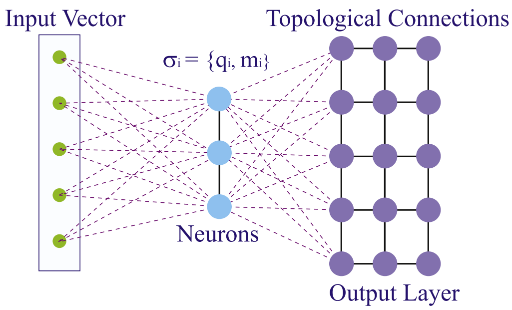


Fig. 4. Two-dimensional SOM lattice mapping hospital service tasks (input layer) to robot neurons (output layer). Each neuron represents a spatial prototype that adjusts through competitive neighborhood learning.

When the confidence score c falls below a threshold $c_{\min}$, the corresponding peer estimate is down-weighted in the SOM competition, preventing unreliable detections from distorting task allocation decisions. This interface ensures that situational awareness and task coordination co-evolve in real time as the hospital environment changes.

4.2. SOM neural network model

The proposed SOM-based neural network emulates the distributed and adaptive characteristics of a biological neural system, comprising two layers: an input layer representing spatially distributed tasks and a competitive output layer representing robot neurons. Let $\mathcal{T} = \{T_1, \dots, T_\sigma\}$ denote the set of active service tasks and $\mathcal{R} = \{R_1, \dots, R_\beta\}$ denote the set of robots. Each task T_j is encoded as a spatial input vector $\mathbf{s}_j = [x_j, y_j]^T$, while each neuron n in the SOM lattice $\mathcal{L}$ maintains a weight vector $\mathbf{w}_n(t) = [w_n^x, w_n^y]^T$ corresponding to a learned spatial prototype. Neurons are arranged in a 2D grid, forming a topological mapping between task space and the robot network.

At each iteration, a task $\mathbf{s}_j$ is presented to the network. The neuron whose weight vector is closest to the task location becomes the *Best Matching Unit* (BMU):

$$n^\star = \arg\min_{n\in\mathcal{L}} \|\mathbf{s}_j - \mathbf{w}_n(t)\|_2. \tag{18}$$

Once the BMU is identified, neighboring neurons update their weights toward the input via a Gaussian neighborhood function:

$$\mathbf{w}_m(t+1) = \mathbf{w}_m(t) + \alpha(t)\, h_{n^\star,m}(t)\,\big(\mathbf{s}_j - \mathbf{w}_m(t)\big), \tag{19}$$

where $\alpha(t)$ is the learning rate and

$$h_{n^\star,m}(t) = \exp\left(-\frac{d_{\mathcal{L}}(n^\star, m)^2}{2\sigma(t)^2}\right), \tag{20}$$

with $d_{\mathcal{L}}(n^\star, m)$ denoting the lattice distance between BMU $n^\star$ and neighbor m. Both $\alpha(t)$ and the neighborhood radius $\sigma(t)$ decay exponentially:

$$\alpha(t) = \alpha_f + (\alpha_0 - \alpha_f)\, e^{-t/\tau},$$
$$\sigma(t) = \sigma_f + (\sigma_0 - \sigma_f)\, e^{-t/\tau}. \tag{21}$$

The SOM's self-organizing property ensures that robots converge toward spatially coherent task clusters aligned with their feasible operating regions, maintaining topological order and avoiding redundant coverage. This is particularly beneficial in structured hospital layouts where the corridor topology is fixed but tasks appear dynamically throughout the shift. Fig. 4 shows the 2D SOM lattice mapping hospital service tasks to robot neurons.

4.3. Priority-weighted competitive learning

In hospital environments, service tasks differ substantially in urgency and consequence, ranging from time-critical medical emergencies to routine supply deliveries. To reflect this, a *priority weight* $A_j(t) \in [0,1]$ is assigned to each task T_j at time t, integrating four contextual factors:

$$A_j(t) = \lambda_1\, \phi_{\text{urg}}(T_j, t) + \lambda_2\, \phi_{\text{due}}(T_j, t) + \lambda_3\, \phi_{\text{prox}}(T_j, t) + \lambda_4\, \phi_{\text{risk}}(T_j), \tag{22}$$

where ϕ_{urg} increases with task criticality (e.g., nurse call or medical emergency), ϕ_{due} reflects the remaining time until the service deadline, ϕ_{prox} measures the distance from the nearest available robot to the task location, and ϕ_{risk} captures environmental constraints such as infectious zone designations or sterilization requirements. The coefficients $\lambda_i \geq 0$ satisfy $\sum_i \lambda_i = 1$ and can be configured by hospital staff to reflect operational policy.

The proximity factor $\phi_{\text{prox}}(T_j, t)$ is computed using the peer robot positions $\{\hat{\mathbf{p}}_{\text{det}}^{(i)}\}$ supplied by the CNN-LiDAR module (Section 3, Eq. (16)):

$$\phi_{\text{prox}}(T_j, t) = 1 - \frac{\min_i \|\mathbf{s}_j - \hat{\mathbf{p}}_{\text{det}}^{(i)}\|_2}{d_{\max}}, \tag{23}$$

where $d_{\max}$ is a normalization constant. When the detection confidence $c^{(i)}$ from Eq. (17) falls below a threshold $c_{\min}$, the corresponding peer estimate is down-weighted in Eq. (23), preventing unreliable detections from distorting the priority score. Each SOM output neuron n is associated with a robot index $i(n) \in \{1, \dots, \beta\}$. In the implementation used here, there is one active neuron per robot; if multiple lattice neurons are used per robot, $i(n)$ denotes the robot that currently owns neuron n after the latest assignment gate. The standard BMU selection in Eq. (18) is replaced by a priority-weighted, workload-balanced competitive cost:

$$C_n(\mathbf{s}_j, t) = \|\mathbf{s}_j - \mathbf{w}_n(t)\|_2 \big(1 + v_{i(n)}(t)\big)\, e^{-\kappa A_j(t)}, \tag{24}$$

where $v_{i(n)}(t) = \frac{P_{i(n)}(t) - \bar{P}(t)}{1+\bar{P}(t)}$ is the relative workload of the robot associated with neuron n, computed from its current accumulated path length $P_{i(n)}(t)$ and the fleet average $\bar{P}(t)$. The path lengths $P_i(t)$ of detected fleet members are updated using the peer positions provided by the CNN-LiDAR module, giving the SOM accurate workload estimates even without centralized communication. The exponential priority term biases high-urgency tasks to dominate the competition field, ensuring rapid acquisition and stable assignment while low-priority tasks remain fluid. The neighborhood function is similarly modulated by the task priority:

$$h_{n^\star,m}(t) = \exp\left(-\frac{d_{\mathcal{L}}(n^\star, m)^2}{2\sigma(t)^2}\right)\big(1 + \eta A_j(t)\big), \tag{25}$$

where η is a gain coefficient. High-priority tasks therefore form tighter neighborhood clusters in the SOM lattice, stabilizing their allocation earlier in the learning process, while low-priority tasks maintain greater flexibility for reassignment as the operational state evolves. Fig. 5 illustrates the competitive learning process.

Each neuron's weight vector can be interpreted as a *virtual target*, an adaptive proxy between the robot's actual navigation goal and its abstract SOM representation. Virtual targets evolve through competitive learning and serve as anchor points for Lazy Theta* path planning in the subsequent stage.

4.4. Integration with Lazy Theta* path planning

Once tasks are tentatively assigned by the SOM, each robot computes a feasible initial path using the Lazy Theta* algorithm. Lazy Theta* is selected because hospital corridors and ward layouts are structured but not naturally restricted to grid-edge motion: an any-angle planner can exploit direct visibility across corridor segments, while the lazy line-of-sight evaluation reduces redundant collision checks on the MGT graph (Nash et al., 2010). This makes it well suited for repeated route evaluation inside the SOM allocation loop, where travel cost and ETA estimates must be updated whenever task priorities or peer positions change. For robot R_i assigned to task T_j, the search cost over the MAKLINK graph $\mathcal{G}$ is:

$$f(n) = g(n) + h(n), \tag{26}$$

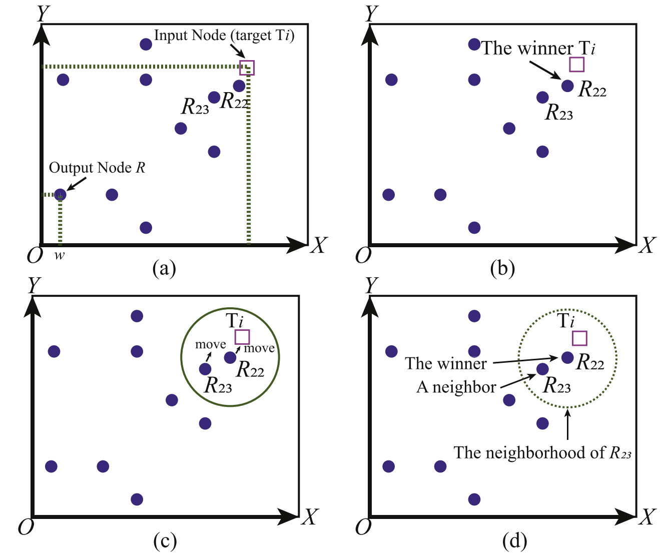


Fig. 5. Competitive learning process of the priority-weighted SOM: (a) initial task presentation; (b) BMU selection via Eq. (24); (c) neighborhood adaptation; (d) updated neuron positions after priority-weighted learning.

where $g(n)$ is the cost from the robot's current position to node n and $h(n)$ is the Euclidean heuristic to the task location. To bias paths toward high-priority regions, the path cost is modulated by a task-density term:

$$g_{\text{density}}(n) = g_{\text{base}}(n)\,\big(1 - \beta D(n)\big), \tag{27}$$

where $D(n)$ is the normalized task density at graph node n and β is a weighting factor. During execution, each robot continuously updates its path and re-estimates its expected time of arrival (ETA). If a competing robot is predicted to reach the same task sooner, a *shadow suppression* mechanism reduces the priority weight $A_j(t)$ of that task for the slower robot:

$$A_j(t) \leftarrow \rho_d\, A_j(t), \quad \rho_d \in (0,1), \tag{28}$$

triggering automatic reassignment without requiring global broadcast. SOM neuron weights are continuously updated along the new Lazy Theta* path segments to maintain consistency between the neural allocation map and actual navigation trajectories. The complete procedure is summarized in Algorithm 2.

To further illustrate the behavior of the priority-weighted SOM, Fig. 6 visualizes how each robot perceives and interacts with its surrounding task field through virtual target representations. Each robot maintains a local priority field that governs its sensitivity to nearby tasks. Virtual targets adjust continuously according to competition outcomes and priority gradients, allowing the SOM to interpolate task positions smoothly even when robots have limited or delayed information about real targets. Robots thereby negotiate task ownership locally without explicit centralized communication, while the SOM preserves a consistent topological mapping across all agents.

The priority-weighted SOM with Lazy Theta* integration provides a biologically inspired and computationally efficient mechanism for coordinating hospital service robots. The SOM enables continuous adaptation to new tasks and dynamic reallocation based on contextual priority and real-time fleet awareness from the CNN-LiDAR module, while Lazy Theta* ensures energy-efficient, any-angle navigation within the MGT graph. By coupling task competition and path learning within a single iterative loop, the system exhibits both global coordination and local adaptability, forming the initial path set $\Pi^{(0)}$ that is subsequently refined by the iMGO algorithm.

Algorithm 2: Priority-weighted SOM with Lazy Theta* integrated Task Allocation.

Input : Task set $\mathcal{T}(t)$; robot states $\{(\mathbf{p}_i, P_i(t))\}$; peer estimates $\{\hat{\mathbf{p}}^{(i)}_{\text{det}}, c^{(i)}\}$ from CNN-LiDAR; SOM lattice $\mathcal{L}$; MAKLINK graph $\mathcal{G}$
Output: Task–robot assignments; initial paths $\Pi^{(0)}$

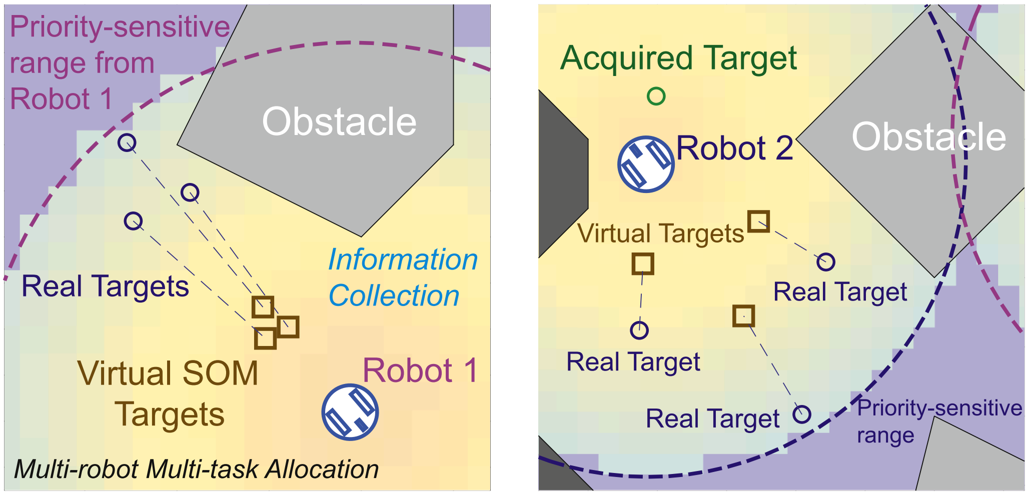
Initialize $\{\mathbf{w}_n(0)\}$, $\alpha(0)$, $\sigma(0)$, τ
Update $P_i(t)$ using $\{\hat{\mathbf{p}}^{(i)}_{\text{det}}\}$; compute fleet average $\bar{P}(t)$
foreach *task* $T_j \in \mathcal{T}(t)$ **do**
  Compute ϕ_{prox} via Eq. (23) using $\{\hat{\mathbf{p}}^{(i)}_{\text{det}}, c^{(i)}\}$
  Compute priority weight $A_j(t)$ via Eq. (22)
  foreach *neuron* $n \in \mathcal{L}$ **do**
    Evaluate $C_n(\mathbf{s}_j, t)$ via Eq. (24)
    Apply shadow suppression if ETA conflict detected (Eq. (28))
  $n^\star \leftarrow \arg\min_n C_n(\mathbf{s}_j, t)$
  Assign $R_{i(n^\star)}$ to T_j
  Run Lazy Theta* on $\mathcal{G}$ for $R_{i(n^\star)} \to T_j$ (Eqs. (26)–(27))
  Compute $\text{ETA}_{i(n^\star)}$
  if *assignment gate passes* **then**
    Accept; lock neuron $n^\star$ to $R_{i(n^\star)}$
    Update virtual target along Lazy Theta* segment
  else
    Defer: $A_j \leftarrow \rho_d A_j$; requeue T_j
  foreach *neighbor* m *of* $n^\star$ **do**
    Compute $h_{n^\star,m}(t)$ via Eq. (25)
    $\mathbf{w}_m(t+1) \leftarrow \mathbf{w}_m(t) + \alpha(t)\, h_{n^\star,m}(t)\,(\mathbf{s}_j - \mathbf{w}_m(t))$
Anneal $\alpha(t)$, $\sigma(t)$ via Eq. (21)
Dispatch $\Pi^{(0)}$ to iMGO for global refinement

5. Path refinement via improved moss growth optimization

Following the priority-weighted SOM task allocation and Lazy Theta* any-angle routing, this section develops a global path refinement stage based on an improved Moss Growth Optimization algorithm, denoted **iMGO**. iMGO is motivated by the fact that the Lazy Theta* route is intentionally computed quickly inside the allocation loop and therefore provides a feasible but not necessarily smooth or clearance-aware trajectory. Directly executing this initial route in a hospital corridor may introduce unnecessary turning, reduced clearance near buffered obstacles, and avoidable overlap with other robots. Rather than replanning from scratch, iMGO accepts the Lazy Theta* trajectory $\Pi^{(0)}$ as a warm-started feasible solution and constructs a reduced, visibility-preserving search domain on the MAKLINK graph $\mathcal{G}$. It then continuously shifts waypoints along MAKLINK edges to shorten the path, increase obstacle clearance, and smooth curvature, while maintaining strict line-of-sight (LoS) feasibility throughout. Beyond the baseline MGO framework (Zheng et al., 2024), iMGO introduces two targeted enhancements designed for multi-robot hospital scenarios: *elite-guided adaptive step annealing*, which accelerates convergence by dynamically adjusting the optimization schedule based on solution improvement rates, and *fleet-level warm-start initialization*, which leverages optimized paths from neighboring robots to reduce redundant search across the fleet.

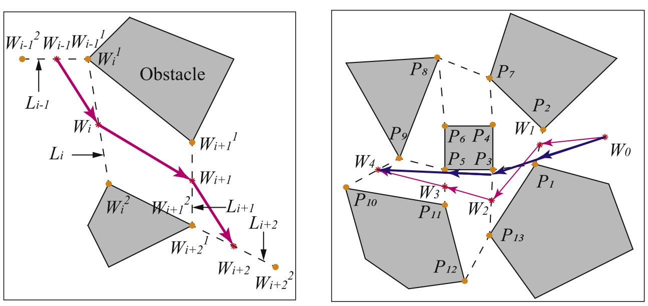
5.1. From initial path to MAKLINK refinement domain

Let the initial path produced by Lazy Theta* be $\Pi^{(0)} = \{\mathbf{W}_0, \mathbf{W}_1, \ldots, \mathbf{W}_{d+1}\}$, where $\mathbf{W}_0$ is the robot's current position and $\mathbf{W}_{d+1}$ is the assigned task location. Each segment $(\mathbf{W}_i, \mathbf{W}_{i+1})$ crosses one or more free MAKLINK lines in $\mathcal{M}$. Enumerating these crossings



(a) (b)

Fig. 6. Priority-weighted SOM multi-robot task allocation. (a) Robot 1 collects situational information within its priority-sensitive range, generating virtual SOM targets aligned with real task clusters. (b) Robot 2 updates its virtual target representations after acquiring nearby tasks. Colored dashed circles indicate the influence fields of different robots; shaded regions denote obstacle-induced visibility constraints.



(a) (b)

Fig. 7. Initial path induces a MAKLINK refinement domain. (a) The initial any-angle path (pink) crosses a set of free MAKLINK lines (dashed black), forming the raw search dimensions. (b) iMGO optimizes continuous waypoints on these lines and yields the refined path (blue).

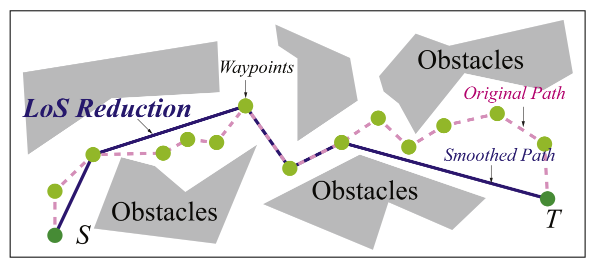


Fig. 8. Edge-based LoS pruning principle. Each MAKLINK line is represented by its two endpoints; both outer-edge and cross-edge visibility must hold so that any waypoint pair selected along consecutive lines remains mutually visible during continuous optimization.

in traversal order yields a raw set of lines $\mathcal{L}_{\text{hit}} = \{L_{k_1}, L_{k_2}, \dots, L_{k_n}\}$. Fig. 7 illustrates this transition: panel (a) shows the initial any-angle path and all intersected MAKLINK lines, which together define the raw search dimensions for refinement; panel (b) shows the refined trajectory obtained by sliding the crossing points along these lines after iMGO.

Although $\mathcal{L}_{\text{hit}}$ guarantees feasibility, it is often unnecessarily high-dimensional, slowing continuous optimization. We therefore compress it using edge-based LoS pruning (Section 5.2), producing a compact ordered set $\widetilde{\mathcal{L}} = \{\widetilde{L}_1, \dots, \widetilde{L}_m\}$ with $m \ll n$, without sacrificing any visibility guarantees needed for continuous waypoint motion along edges.

5.2. Edge-based line-of-sight pruning

A naive pruning that checks LoS only between *midpoints* of consecutive MAKLINK lines is insufficient: since iMGO later slides waypoints to arbitrary positions along each edge, midpoint-only visibility does not imply that *every* point on one edge sees *every* point on the next. Such gaps cause refined segments to clip buffered obstacles. We therefore adopt a stricter **four-way edge-based LoS** test that preserves feasibility for all continuous waypoint choices (Fig. 8).

For two consecutive lines L_a and L_b with endpoints $(\mathbf{e}_a^1, \mathbf{e}_a^2)$ and $(\mathbf{e}_b^1, \mathbf{e}_b^2)$, we test four segments against the buffered obstacle set $\mathcal{O}^\delta$:

$$\begin{aligned} \mathcal{S}_{\text{outer}}(L_a, L_b) &= \left\{ \overline{\mathbf{e}_a^1\mathbf{e}_b^1}, \overline{\mathbf{e}_a^2\mathbf{e}_b^2} \right\}, \\ \mathcal{S}_{\text{cross}}(L_a, L_b) &= \left\{ \overline{\mathbf{e}_a^1\mathbf{e}_b^2}, \overline{\mathbf{e}_a^2\mathbf{e}_b^1} \right\}, \\ \text{LoS}(L_a, L_b) &= \bigwedge_{s \in \mathcal{S}_{\text{outer}} \cup \mathcal{S}_{\text{cross}}} \left(s \cap O = \emptyset,\ \forall O \in \mathcal{O}^\delta \right). \end{aligned} \tag{29}$$

If all four segments are collision-free, then every pair of points on L_a and L_b is mutually visible by convexity of the visibility kernel induced by the endpoints, and L_b can be merged into the connectivity of L_a without introducing a new optimization dimension. If any check fails, L_b is retained as a live dimension. We scan $\mathcal{L}_{\text{hit}}$ greedily in traversal order, merging lines while four-way LoS holds and restarting whenever a check fails, to obtain the compact live set $\widetilde{\mathcal{L}}$. Fig. 9 shows a worked example where 13 raw lines compress to 4 live dimensions.

5.3. Edge parameterization and optimization objective

For each live line $\widetilde{L}_i$ with endpoints $(\mathbf{E}_i^1, \mathbf{E}_i^2)$, a refined waypoint is defined by a scalar parameter $\gamma_i \in [0, 1]$:

$$\mathbf{Z}_i(\gamma_i) = \mathbf{E}_i^1 + \gamma_i (\mathbf{E}_i^2 - \mathbf{E}_i^1), \quad i = 1, \dots, m. \tag{30}$$

The refined path is $\Pi(\boldsymbol{\gamma}) = \{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_{m+1}\}$, with $\mathbf{P}_0 = \mathbf{W}_0$, $\mathbf{P}_{m+1} = \mathbf{W}_{d+1}$, and $\mathbf{P}_i = \mathbf{Z}_i(\gamma_i)$ for $1 \le i \le m$. By construction, consecutive

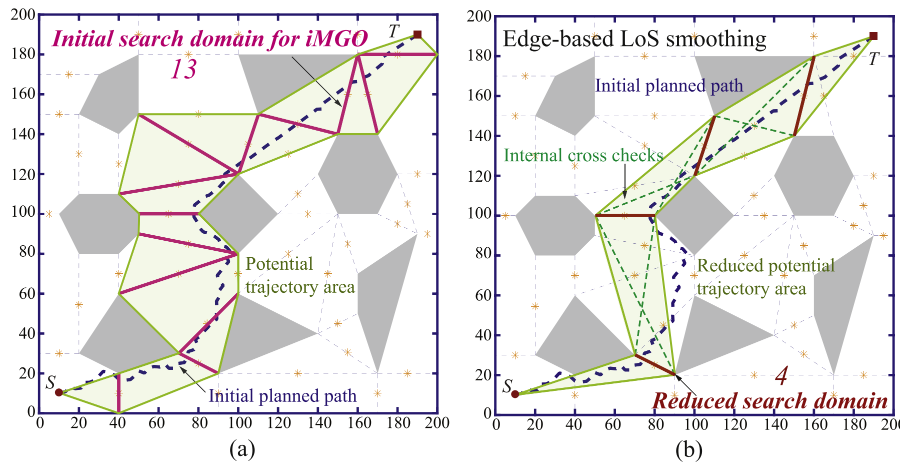


Fig. 9. LoS compaction workflow. (a) The initial path (dashed blue) traverses 13 MAKLINK lines; inner (light green dashed) and outer (light green solid) edge checks are performed pairwise. (b) Only four lines pass all four checks and are retained as live dimensions (maroon), preserving feasibility for all continuous waypoint placements and substantially reducing the optimization dimension.

waypoints preserve LoS for all $\gamma_i \in [0,1]$ due to the four-way endpoint checks.

The optimization minimizes a multi-criterion path cost:

$$\begin{aligned} J(\boldsymbol{\gamma}) = &\sum_{i=0}^{m} \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2 \\ &+ \omega_\kappa \sum_{i=1}^{m-1} \|\mathbf{P}_{i+1} - 2\mathbf{P}_i + \mathbf{P}_{i-1}\|_2 \\ &+ \omega_c \sum_{i=0}^{m+1} \psi\big(d_{\min}(\mathbf{P}_i)\big) \; + \; \omega_e \sum_{i=0}^{m} \mathcal{E}(\mathbf{P}_i, \mathbf{P}_{i+1}), \end{aligned} \tag{31}$$

where the second term penalizes curvature via discrete second differences. For reproducibility, the clearance penalty is implemented as a squared hinge loss

$$\psi(d_{\min}(\mathbf{P}_i)) = \left[\frac{\max(0, d_{\text{safe}} - d_{\min}(\mathbf{P}_i))}{d_{\text{safe}} + \varepsilon}\right]^2, \tag{32}$$

where $d_{\min}(\mathbf{P}_i) = \min_{O\in\mathcal{O}^\delta} \text{dist}(\mathbf{P}_i, O)$ is the Euclidean distance from waypoint $\mathbf{P}_i$ to the nearest buffered obstacle, d_{safe} is the desired clearance margin, and ε avoids division by zero. The energy proxy combines segment length and turning effort:

$$\mathcal{E}(\mathbf{P}_i, \mathbf{P}_{i+1}) = \ell_i\big(1 + \lambda_\theta|\Delta\theta_i|\big), \quad \ell_i = \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2, \tag{33}$$

where $\Delta\theta_i = \text{wrap}(\theta_i - \theta_{i-1})$, $\theta_i = \text{atan2}(P_{i+1,y} - P_{i,y}, P_{i+1,x} - P_{i,x})$, and $\lambda_\theta \geq 0$ controls the penalty on sharp heading changes. The weights $\omega_\kappa, \omega_c, \omega_e \geq 0$ can be tuned per robot platform. Edge bounds $\gamma_i \in [0,1]$ enforce domain constraints, while LoS feasibility is guaranteed by compaction.

5.4. iMGO: improved moss growth optimization

The baseline MGO algorithm (Zheng et al., 2024) maintains a population $\{\boldsymbol{\gamma}^{(q)}\}_{q=1}^{N}$ and updates each candidate via four bio-inspired operators adapted to the low-dimensional MAKLINK edge domain:

Wind direction (global drift). A direction vector from the population mean to the incumbent best $\boldsymbol{\gamma}^\star$ moves all candidates toward the most promising region while preserving diversity.

Spore dispersal (stochastic exploration). Zero-mean random jumps with exponentially decaying step energy enable long-range moves early in optimization and fine perturbations near convergence.

Dual propagation (local refinement). A per-dimension mixture of attraction-to-best and self-perturbation sharpens candidate solutions near promising waypoint configurations.

Cryptobiosis (elite memory). A short elite list retains the best solutions found; if progress stalls, the best-so-far is revived and the step schedule is reheated to escape local plateaus.

The generic update for candidate q at iteration t is:

$$\boldsymbol{\gamma}^{(q)} \leftarrow \text{Proj}_{[0,1]^m}\Big[\boldsymbol{\gamma}^{(q)} + \eta_g(t)\,\mathcal{D}_{\text{wind}} + \eta_s(t)\,\xi_{\text{spore}} + \eta_p(t)\,\xi_{\text{prop}}\big(\boldsymbol{\gamma}^{(q)}, \boldsymbol{\gamma}^\star\big)\Big], \tag{34}$$

where the projection enforces $\gamma_i \in [0,1]$, and the step sizes η_g, η_s, η_p decay exponentially in the baseline MGO.

5.4.1. Elite-guided adaptive step annealing

In the baseline MGO, the step sizes η_g, η_s, and η_p follow a fixed exponential schedule regardless of the actual progress of the search. This can be inefficient in the MAKLINK domain: when the population is converging rapidly, premature fine-grained annealing limits further improvement, while stagnation is not detected until the schedule has already cooled. iMGO replaces the fixed schedule with an *elite-guided adaptive annealing* mechanism that dynamically adjusts each step size based on the relative improvement rate of the best solution over a sliding window of W iterations:

$$\rho(t) = \frac{J(\boldsymbol{\gamma}^\star_{t-W}) - J(\boldsymbol{\gamma}^\star_t)}{J(\boldsymbol{\gamma}^\star_{t-W}) + \varepsilon}, \tag{35}$$

where ε is a small constant for numerical stability. The step sizes are then modulated as:

$$\eta_k(t+1) = \begin{cases} \min\big(\eta_k(t)/\mu_d,\ \eta_k^{\max}\big) & \text{if } \rho(t) > \rho_{\text{high}}, \\ \max\big(\eta_k(t)\cdot\mu_d,\ \eta_k^{\min}\big) & \text{if } \rho(t) < \rho_{\text{low}}, \\ \eta_k(t) & \text{otherwise}, \end{cases} \tag{36}$$

for $k \in \{g, s, p\}$, where $\mu_d > 1$ is a step modulation factor and $\rho_{\text{high}} > \rho_{\text{low}}$ are improvement thresholds. When the elite solution is improving rapidly ($\rho > \rho_{\text{high}}$), the step sizes are increased to sustain momentum. When improvement stalls ($\rho < \rho_{\text{low}}$), they are reduced to focus the search, effectively triggering a soft cryptobiosis recovery without requiring full population reinitialization. This adaptive schedule replaces the fixed $e^{-t/\tau}$ decay of baseline MGO in Eq. (34), with the projection in Eq. (34) unchanged.

5.4.2. Fleet-level warm-start initialization

In a multi-robot scenario, multiple robots simultaneously refine their paths on the same MAKLINK graph $\mathcal{G}$. Robots assigned to spatially adjacent tasks traverse overlapping or geometrically similar corridor segments, so their optimal waypoint parameters $\boldsymbol{\gamma}^\star$ are often close in the parameter space $[0,1]^m$. Baseline MGO initializes each robot's population independently near the midpoints of live MAKLINK lines, discarding this shared geometric structure.

iMGO introduces a *fleet-level warm-start initialization* that leverages already-completed optimizations within the fleet. When robot R_i begins refinement for task T_j, it queries the set of recently completed path solutions from other fleet members:

$$\mathcal{S}_{\text{fleet}} = \left\{\boldsymbol{\gamma}^{\star(k)} \;:\; R_k \text{ has completed iMGO for a task within } d_{\text{ws}} \text{ of } T_j\right\}, \tag{37}$$

where d_{ws} is a spatial proximity threshold. If $\mathcal{S}_{\text{fleet}}$ is non-empty, at most $N_{\text{ws}} = \min(N, |\mathcal{S}_{\text{fleet}}|)$ prior solutions are used to seed the initial population of R_i:

$$\boldsymbol{\gamma}^{(q)}(0) = \begin{cases} \mathcal{M}_{k_q\to i}(\boldsymbol{\gamma}^{\star(k_q)}) + \boldsymbol{\epsilon}_q & q \le N_{\text{ws}}, \\ \mathbf{U}[0,1]^m & q > N_{\text{ws}}, \end{cases} \tag{38}$$

where $\boldsymbol{\gamma}^{\star(k_q)}$ is the qth nearest solution from the fleet pool and $\mathcal{M}_{k_q\to i}(\cdot)$ maps that previous solution onto the current live MAKLINK line set $\widetilde{\mathcal{L}}_i$ of robot R_i. Because different robots may retain different live line sets after LoS compaction, the mapping is implemented geometrically: each waypoint from the previous optimized path is projected onto the nearest current live line in traversal order; missing dimensions are initialized at the corresponding line midpoint, and multiple projections onto the same live line are averaged before clipping to $[0,1]$. The perturbation $\boldsymbol{\epsilon}_q \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{ws}}^2\mathbf{I})$ maintains diversity, and $\mathbf{U}[0,1]^m$ fills the remainder of the population with uniform random samples. When $\mathcal{S}_{\text{fleet}}$ is empty (e.g., for the first robot to be dispatched), the initialization falls back to the standard midpoint-centered approach of baseline MGO. The fleet warm-start reduces the effective search space by placing a portion of the initial population near previously validated high-quality solutions, substantially accelerating convergence in the early iterations of iMGO. It requires only the broadcast of the scalar vector $\boldsymbol{\gamma}^\star \in [0,1]^m$ from each robot upon task completion, imposing negligible communication overhead.

Algorithm 3 summarizes the complete single-robot path refinement procedure, incorporating both iMGO enhancements. The algorithm accepts the initial path $\Pi^{(0)}$ from the SOM with Lazy Theta* stage and returns a refined, collision-free path $\Pi^\star$. When upstream task assignments change (e.g., due to SOM reassignment triggered by new CNN-LiDAR detections), the initial path is recomputed by Lazy Theta* and iMGO restarts from the previous $\boldsymbol{\gamma}^\star$ restricted to the new $\widetilde{\mathcal{L}}$, combining the robustness of warm-start with the adaptivity of the elite-guided annealing to re-converge rapidly under dynamic hospital conditions.

Algorithm 3: Single-robot path refinement via iMGO.

Input : Initial path $\Pi^{(0)}$; MAKLINK graph $\mathcal{G}$; buffered obstacles $\mathcal{O}^\delta$; fleet solution pool $\mathcal{S}_{\text{fleet}}$; iMGO parameters ($N$, maxIter, W, μ_d, ρ_{high}, ρ_{low}, d_{ws}, σ_{ws})
Output: Refined path $\Pi^\star$; solution $\boldsymbol{\gamma}^\star$ for fleet pool

Collect $\mathcal{L}_{\text{hit}}$ from $\Pi^{(0)}$ (Fig. 7(a))
Apply four-way LoS compaction to obtain $\widetilde{\mathcal{L}} = \{\widetilde{L}_1, \ldots, \widetilde{L}_m\}$ (Figs. 8, 9)
Define $\mathbf{Z}_i(\gamma_i)$ via Eq. (30) for each $\widetilde{L}_i$
`// Fleet-level warm-start (Enhancement 2)`
Initialize population $\{\boldsymbol{\gamma}^{(q)}(0)\}_{q=1}^N$ via Eq. (38) using $\mathcal{S}_{\text{fleet}}$
Initialize elite list $\mathcal{E} \leftarrow \emptyset$; set step sizes η_g, η_s, η_p
for $t = 1$ **to** maxIter **do**
  Evaluate $J(\boldsymbol{\gamma}^{(q)})$ via Eq. (31) for all q
  Update $\boldsymbol{\gamma}^\star$ and elite list $\mathcal{E}$
  `// Elite-guided adaptive annealing (Enhancement 1)`
  Compute $\rho(t)$ via Eq. (35)
  Update η_g, η_s, η_p via Eq. (36)
  Compute wind direction $\mathcal{D}_{\text{wind}}$ (population mean $\to \boldsymbol{\gamma}^\star$)
  foreach *candidate* $\boldsymbol{\gamma}^{(q)}$ **do**
    Draw ξ_{spore}; compute $\xi_{\text{prop}}(\boldsymbol{\gamma}^{(q)}, \boldsymbol{\gamma}^\star)$
    Update $\boldsymbol{\gamma}^{(q)}$ via Eq. (34); project to $[0,1]^m$
  if *stagnation detected from* $\mathcal{E}$ **then**
    Revive best from $\mathcal{E}$; reheat η_g, η_s, η_p
  if $\rho(t) < \varepsilon$ *and step sizes below threshold* **then**
    break `// Convergence`
Broadcast $\boldsymbol{\gamma}^\star$ to $\mathcal{S}_{\text{fleet}}$
return $\Pi^\star = \{\mathbf{W}_0, \mathbf{Z}_1(\gamma_1^\star), \ldots, \mathbf{Z}_m(\gamma_m^\star), \mathbf{W}_{d+1}\}$

6. Simulation and comparison studies

To comprehensively evaluate the proposed framework, simulation studies are organized into four categories corresponding to the three modules and their integrated operation. First, the CNN-LiDAR situational awareness module is validated through training convergence analysis, point-wise classification metrics, and GPS outage simulation in a hospital corridor environment. Second, the priority-weighted SOM task allocation is compared against clustering and non-clustering genetic algorithm (GA) baselines (Janati et al., 2017) under identical multi-robot multi-task scenarios. Third, iMGO path refinement is benchmarked against ACO and IGA under different initial planners and in dynamic obstacle environments. Finally, an integrated simulation combines all three modules to demonstrate the end-to-end coordination capability of the complete framework. All experiments are conducted in simulation; each module is validated independently, and the integrated scenario demonstrates the pipeline-level interaction described in Sections 3–5.

6.1. CNN-LiDAR situational awareness validation

6.1.1. Simulation environment and data collection

The CNN-LiDAR module is evaluated in a simulated hospital corridor environment modeled as a rectangular floor plan of $150\ m \times 100\ m$, divided into three parallel halls separated by internal walls of $5\ m$ thickness with $35\ m$ openings at each end. This configuration represents the corridor-dominated topology of a typical hospital ward floor, where robots navigate between patient rooms and service areas through defined passageways, and partial occlusions from medical equipment and trolleys are common.

Each robot is modeled as a circular entity with radius $r_{\text{robot}} = 3\ m$ and moves at a constant forward speed of $2.0\ m/s$, with smooth heading variations:

$$\theta_i(t + \Delta t) = \theta_i(t) + \sigma_\theta \cdot \mathcal{N}(0,1), \tag{39}$$

where $\sigma_\theta = 0.02\ rad$ maintains predominantly straight motion typical of hospital service robots following designated routes. Note that r_{robot} here represents the effective detection footprint used in simulation ray-intersection labeling rather than the physical chassis radius of a real hospital robot; it is intentionally set to encompass both the robot body and a conservative safety margin, ensuring that the CNN learns robust signatures across the full proximity zone relevant for collision avoidance. Each robot carries a 2D LiDAR sensor with 360-degree coverage, 1-degree angular resolution ($N = 360$ measurement points), and maximum range $r_{\max} = 80\ m$.

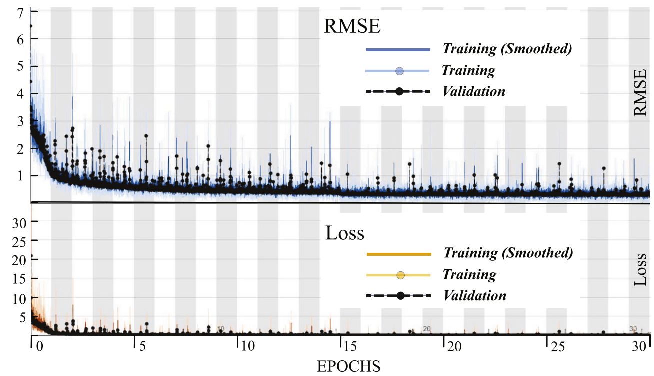


Fig. 10. CNN training convergence over 30 epochs. Blue lines show training RMSE and loss with smoothed averages; black dashed lines show validation performance. The piecewise learning rate schedule produces stable convergence throughout training.

Training data are generated through automated multi-robot simulation at a 1 Hz update rate over 6 h of operation, producing 21,600 samples per robot. A ten-robot configuration is used during data collection to ensure high robot-encounter density in each scan: 1.1% of scans contain no detections, 11.3% contain one, 23.2% contain two, 28.8% contain three, and 35.6% contain four or more robot detections. Ground truth labels are generated automatically from ray intersection analysis. The dataset is split 80/20 between training and validation. Training employs the Adam optimizer with $\alpha_0 = 0.001$, mini-batch size 32, and 30 epochs with geometric learning rate decay to a final rate of 0.0001. The network uses kernel sizes k_1=7, k_2=11, k_3=15, k_4=11 and channel counts C_1=64, C_2=128, C_3=128, C_4=64.

Fig. 10 shows training convergence over 30 epochs. Validation RMSE and loss closely track training metrics throughout, confirming good generalization without significant overfitting to the simulated hospital corridor geometry.

6.1.2. Detection performance

The trained CNN achieves strong point-wise classification performance across the hospital corridor environment. Binary classification metrics computed over all LiDAR measurements yield an overall accuracy of 99.88%, precision of 99.76%, recall of 99.41%, and an F1 score of 0.996. Per-scan analysis shows that 95.3% of scans achieve correct robot count detection. Detection latency averages $3.03\ ms \pm 1.56\ ms$ on a simulated edge processor, enabling real-time operation at typical robot control frequencies without disrupting navigation.

Table 1 presents detection accuracy broken down by robot density and distance range, evaluated over 20 h of simulation across 93,474 data points with GPS disabled. Close-range detections (≤20 m) achieve near-perfect accuracy of 99.4% across all robot density conditions, confirming reliable performance in the most safety-critical operating range for hospital corridor navigation. Medium-range accuracy (≤40 m) averages 92.8%, and far-range accuracy (≤60 m) averages 82.3%, with performance improving at longer distances when more robots are present. The overall detection rate of 90.7% across all conditions is consistent with the 95.3% validation accuracy, confirming that the model generalizes well beyond its training distribution to the varied encounter geometries of the hospital environment.

6.1.3. GPS outage simulations

System performance under realistic failure conditions is evaluated using alternating GPS availability cycles of 10 s ON and 10 s OFF. Fig. 11 illustrates the operational sequence across four phases. In Fig. 11(a), robots navigate with full GPS awareness across the three-hall hospital layout. Fig. 11(b) shows GPS outage initiation: the CNN-LiDAR module activates immediately, identifying peer robots via green detection rays while tracking lines (pink dashed) maintain fleet awareness. Fig. 11(c) demonstrates sustained tracking throughout the outage. Fig. 11(d) presents a polar coordinate visualization of detected peer positions relative to static walls, confirming precise bearing and range estimation under GPS-denied conditions.

Table 1
CNN-LiDAR detection accuracy by robot density and distance range during GPS outage conditions (93,474 data points).

Distance range	Robots in environment			Average
	1	2	3	
Close (≤20 m)	99.3%	99.5%	99.4%	99.4%
Medium (≤40 m)	92.4%	93.4%	92.6%	92.8%
Far (≤60 m)	74.1%	85.2%	87.7%	82.3%

The binary segmentation approach handles varying robot densities robustly, from single-robot encounters in isolated corridors to complex multi-robot configurations at ward junctions, and generalizes across robot movement patterns without overfitting to specific trajectory templates.

6.2. Priority-weighted SOM task allocation comparison

Comparison with Genetic Algorithm Baselines.. The priority-weighted SOM is compared against the method of Janati et al. (2017), which partitions tasks into spatial clusters, assigns robots to clusters, and solves per-cluster assignment using either a non-clustering GA or a clustering GA. Following their evaluation protocol, four robots with identical start states are deployed to complete 20 tasks. Fig. 12 shows representative best-solution trajectories for the three methods.

Table 2 presents per-robot and total path lengths. The priority-weighted SOM achieves a total path length of 1653.64 units, reducing overall distance by 1.6% relative to the non-clustering GA (1680.04) and 1.9% relative to the clustering GA (1685.83). These gains arise from the SOM's ability to form spatially coherent, dynamically adjustable task clusters through priority-weighted competition, allowing robots to self-organize around feasible operating regions without relying on rigid predetermined partitions. The shadow suppression mechanism further prevents redundant travel by triggering micro-reassignments when competing ETAs diverge, a capability that rigid GA-based clustering methods lack entirely.

6.3. iMGO path refinement

6.3.1. Comparison under different initializers

iMGO is evaluated as a universal refinement stage under two initial planners: MGT-Dijkstra (graph-optimal but midpoint-biased) and Lazy Theta* (any-angle, LoS-driven). Both are tested on a 200×200 map with eight obstacles following the environment setting of Yang et al. (2020). For each initializer, the MAKLINK lines intersected by the initial path are extracted, edge-based LoS pruning reduces the domain to a compact set of live lines, and iMGO optimizes the continuous waypoint parameters $\boldsymbol{\gamma}$ to minimize the composite objective J in Eq. (31).

For Dijkstra initialization, the initial route hugs the corridor perimeter due to discrete midpoint connectivity (Fig. 13). LoS pruning reduces the domain from 7 to 4 live lines (42.9% dimensionality reduction), after which iMGO rapidly shifts waypoints toward geometric boundaries and converges faster than ACO. For Lazy Theta* initialization, the initial path is more flexible but geometrically irregular (Fig. 14). After pruning, six live lines remain; two require interior waypoint placements, so iMGO requires additional fine-tuning iterations. Even so, iMGO converges faster and to shorter paths than ACO, owing to wind-driven global exploration and elite-guided adaptive annealing that sustains progress when improvement rates are high and focuses the search when they stall.

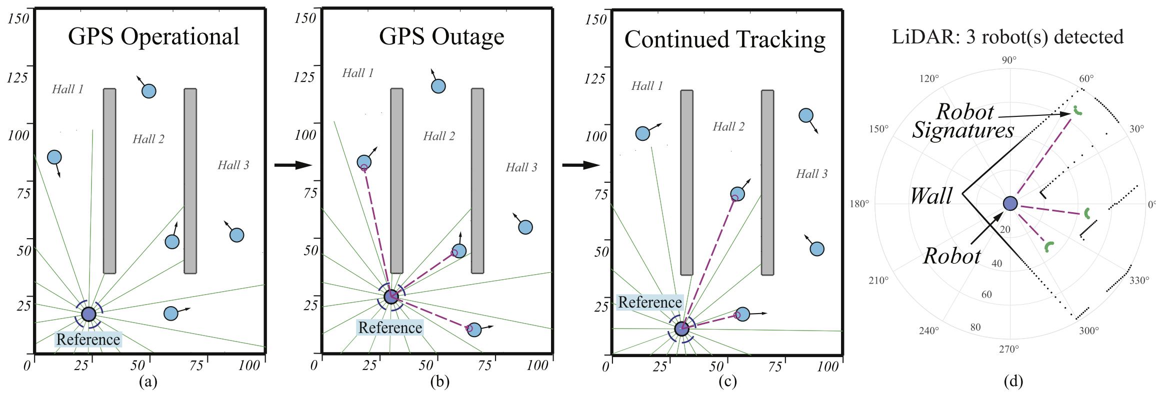


Fig. 11. Hospital corridor simulation during GPS outage transition. (a) Normal GPS operation with full fleet positioning. (b) GPS outage initiation: CNN-LiDAR activates, green rays identify peer robots, pink dashed lines show tracking. (c) Sustained multi-robot tracking during outage. (d) Polar coordinate visualization of detected peer positions; wall returns in black, robot detections in green.

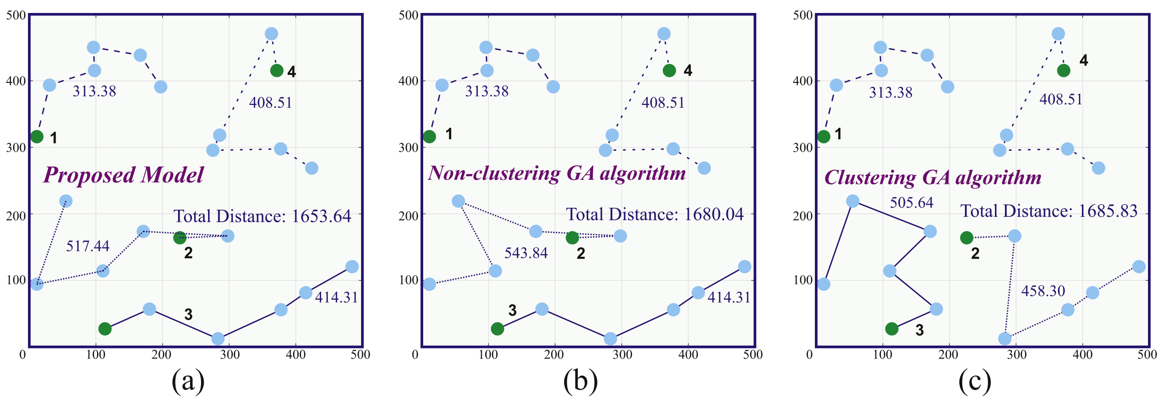


Fig. 12. Best-solution trajectories for (a) proposed priority-weighted SOM with Lazy Theta*; (b) non-clustering GA; (c) clustering GA (Janati et al., 2017). The proposed method achieves the shortest total route with reduced inter-robot overlap in narrow passages.

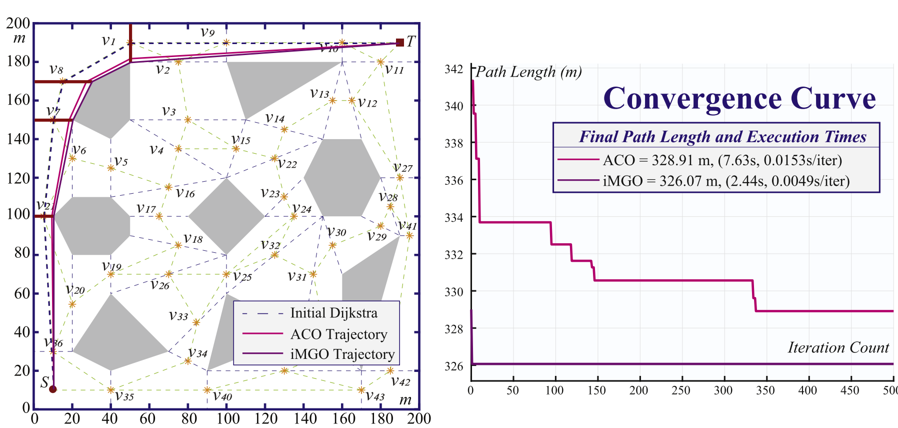


Fig. 13. Dijkstra-based initialization (blue) and refined trajectories: ACO (pink) vs. iMGO (purple).

Table 2
Multi-robot task allocation: per-robot and total path length, consistent with Fig. 12.

Method	R1	R2	R3	R4	Total
Proposed SOM + Lazy Theta*	313.38	517.44	414.31	408.51	**1653.64**
Non-clustering GA	313.38	543.84	414.31	408.51	1680.04
Clustering GA	313.38	505.64	458.30	408.51	1685.83

Table 3 aggregates results across both initializers, with and without LoS pruning, over 10 runs of 500 iterations. The best configuration, Lazy Theta* with LoS pruning, yields minimum and average path lengths of 257.29 and 260.57 units respectively under iMGO, compared to 258.47 and 262.43 under ACO. Across all configurations, iMGO produces on average 1.6% shorter paths in 45.4% less computation time than ACO. The elite-guided adaptive annealing introduced

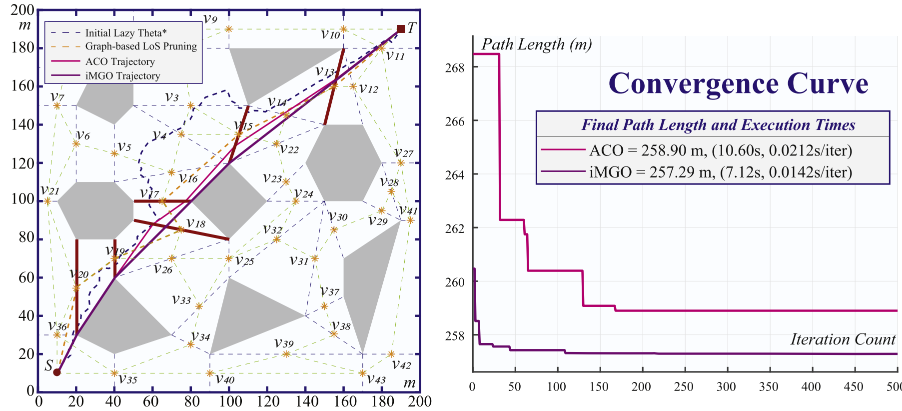


Fig. 14. Lazy Theta* initialization with refinements by ACO (pink) and iMGO (purple). Convergence profiles confirm consistently faster iMGO optimization.

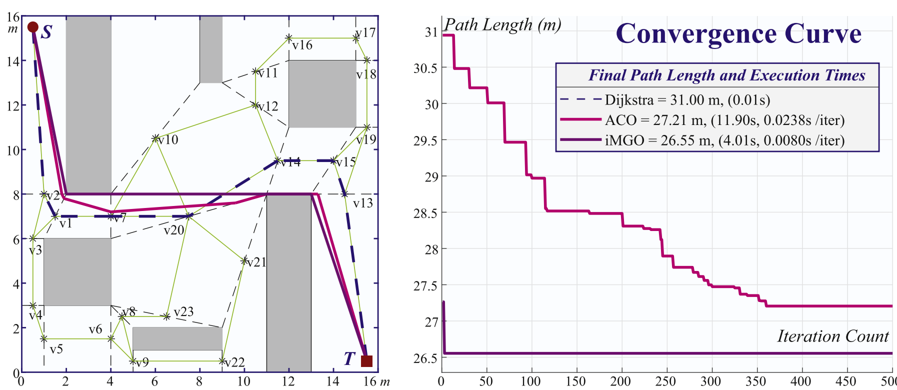


Fig. 15. Static scenario: MGT-Lazy Theta* initial path and iMGO refinement.

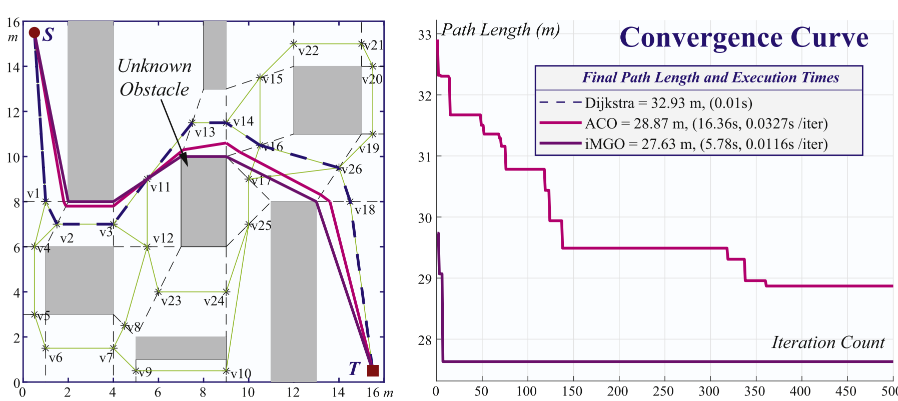


Fig. 16. Dynamic scenario: a new obstacle invalidates a MAKLINK connection; iMGO rapidly re-optimizes the detour.

in iMGO contributes to this efficiency gain by accelerating convergence under rapid improvement and preventing premature cooling during stagnation, compared to the fixed exponential decay of baseline MGO.

6.3.2. *Dynamic obstacle scenario*

iMGO's responsiveness to mid-mission environmental changes is evaluated on a 16×16 corridor map. A new obstacle is inserted during execution, invalidating a previously valid MAKLINK connection and requiring immediate route re-optimization. The initial route is recomputed by MGT-Lazy Theta* and then refined by Dijkstra, IGA, ACO, or iMGO. Figs. 15 and 16 show the static and dynamic refinement results respectively.

Table 4 presents minimum path length and average computation time over 500 trials in both static and dynamic conditions. iMGO achieves the shortest paths in both cases (26.55 static, 27.63 dynamic) and completes refinement in 4.01 s and 3.72 s respectively, outperforming ACO by a wide margin in computation time and matching or exceeding IGA in path quality. The fleet-level warm-start initialization contributes particularly to re-optimization speed in the dynamic case: when a task assignment changes or an obstacle is inserted, iMGO restarts from the previous $\gamma^{\star}$, placing the initial population close to the prior solution and reaching convergence faster than a cold-started search.

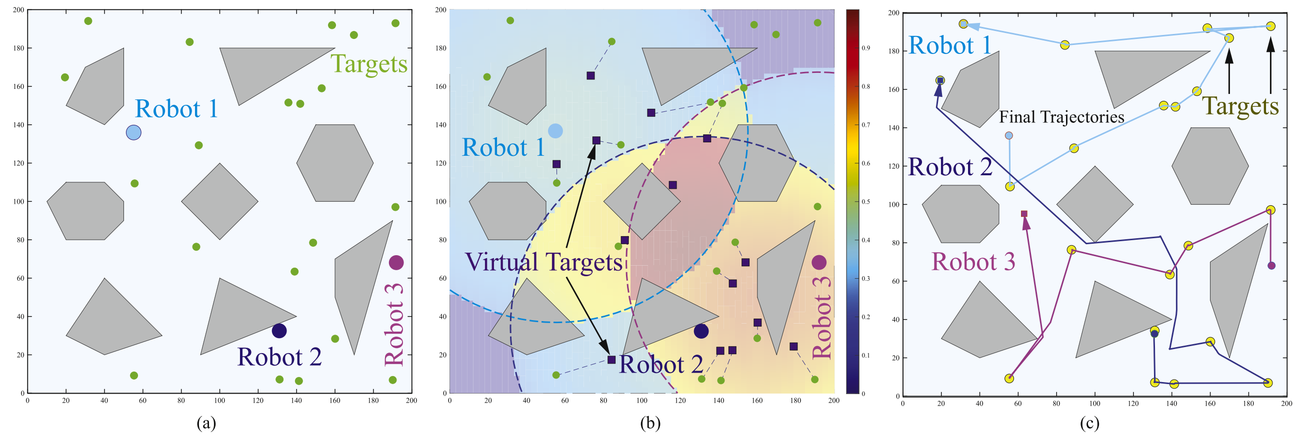


Fig. 17. Integrated multi-robot simulation in a hospital floor plan. (a) Initial configuration: robot start positions and service task locations. (b) Priority-weighted SOM influence map with evolving task regions and virtual targets. (c) Final optimized trajectories after Lazy Theta* routing and iMGO refinement; robots maintain distinct regions and collision-free motion.

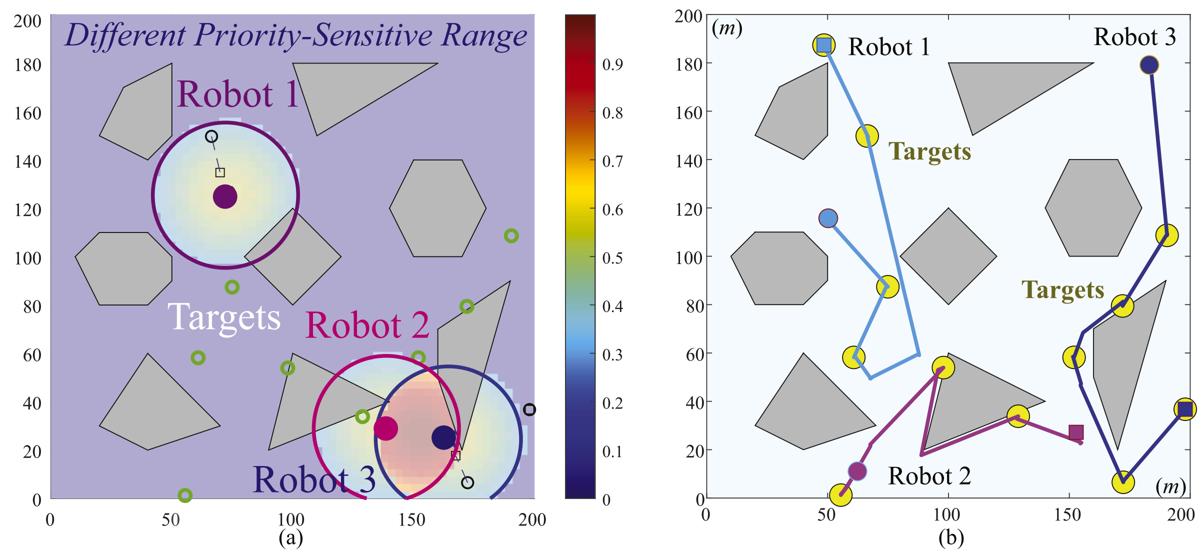


Fig. 18. (a) Priority-weighted SOM influence map during task learning. Color intensity reflects allocation confidence derived from priority weights $A_j(t)$. (b) Final optimized multi-robot trajectories after SOM allocation, Lazy Theta* routing, and iMGO refinement. Robots maintain distinct task regions and collision-free motion.

Table 3
ACO vs. iMGO under different initializers, with/without LoS pruning (10 runs, 500 iterations).

Initializer	Min length		Avg length		Avg time (s)	
	ACO	*iMGO*	ACO	*iMGO*	ACO	*iMGO*
Dijkstra	329.34	326.07	329.90	326.14	8.20	2.80
Dijkstra + LoS	328.91	326.07	329.28	326.07	4.93	1.72
Lazy Theta*	264.11	258.66	273.95	264.45	12.87	9.37
Lazy Theta* + LoS	258.47	**257.29**	262.43	**260.57**	5.65	4.33

Table 4
Path refinement in static and dynamic obstacle scenarios: minimum path length and average computation time (500 trials).

Algorithm	Static		Dynamic	
	Length	Time (s)	Length	Time (s)
Dijkstra	31.00	0.50	32.93	0.50
IGA	27.23	4.05	28.57	4.90
ACO	27.21	11.90	28.81	8.89
iMGO (proposed)	**26.55**	4.01	**27.63**	3.72

6.4. *Integrated multi-robot simulations of the full perception-allocation-refinement pipeline*

To demonstrate the end-to-end capability of the proposed framework, an integrated simulation combines all three modules: CNN-LiDAR situational awareness, priority-weighted SOM task allocation, and iMGO path refinement. The simulated environment (Fig. 17) represents a 200×200 hospital floor plan with polygonal obstacles modeling wards, corridors, and service areas. Four robots $\{R_1, \dots, R_4\}$ are deployed to complete twenty service tasks $\{T_1, \dots, T_{20}\}$ distributed throughout the free space.

At the start of each coordination cycle, the CNN-LiDAR module provides each robot with peer position estimates $\{\hat{\mathbf{p}}^{(i)}_{\text{det}}\}$ and confidence scores $\{c^{(i)}\}$. These feed directly into the priority-weighted SOM, updating the proximity factor ϕ_{prox} and fleet workload estimates $v_i(t)$ used in the competitive cost Eq. (24). Through priority-weighted competitive learning, each robot forms an adaptive influence region that evolves dynamically and converges to a stable allocation configuration. As shown in Fig. 17(b), overlapping influence fields are resolved through winner-take-all competition, ensuring exclusive task ownership and balanced workload distribution across the fleet.

Once allocation converges, Lazy Theta* generates initial collision-free routes on the MGT graph $\mathcal{G}$, which iMGO then refines using fleet-level warm-start initialization and elite-guided adaptive annealing. The resulting optimized trajectories are shown in Fig. 17(c). Robots maintain spatially distinct paths, navigating narrow hospital corridors without collision or congestion.

A second scenario with three robots and randomly generated task distributions is shown in Fig. 18(a) and (b), illustrating the framework's adaptability across different spatial configurations. Fig. 18(a) highlights the evolving influence regions during task learning, with color intensity reflecting allocation confidence. The influence range is adjustable based on environment complexity and operational requirements. Fig. 18(b) presents the final optimized trajectories after convergence, confirming stable allocation even under stochastic task distributions.

To assess robustness and generalization, a Monte Carlo study of 100 independent runs is conducted, with robot count, start positions, task locations, obstacle sizes, and corridor topology all randomized in each run. The integrated framework achieves an average mission time reduction of 18.6% and a total travel distance reduction of 12.4% compared to random task allocation, confirming that the bio-inspired coordination architecture scales effectively across diverse and dynamically changing hospital service environments.

7. Conclusion

This paper presented a bio-inspired, graph-driven coordination framework for multi-robot hospital service systems, integrating three coordination modules, built on a shared MGT environment representation, into a complete perception-to-execution pipeline. A CNN-LiDAR situational awareness module provides fail-safe peer robot detection via binary segmentation of 2D LiDAR scans, maintaining fleet awareness during GPS and communication outages. A priority-weighted SOM dynamically allocates service tasks based on contextual urgency, workload balance, and real-time peer positions from the CNN-LiDAR layer, with Lazy Theta* generating the initial any-angle routes inside the allocation loop. Initial Lazy Theta* routes on the MGT graph are subsequently refined by the iMGO, which introduces elite-guided adaptive step annealing and fleet-level warm-start initialization to accelerate convergence in multi-robot scenarios. The pipeline-level coupling of perception, task assignment, and path execution enables robust coordination even when primary communication infrastructure is unavailable. Simulation and comparison studies validated each module and their integrated operation.

Future work will focus on extending the CNN-LiDAR module to 3D sensing and multi-floor environments and scaling the coordination pipeline to larger hospital fleets through fully decentralized protocols. Real-world deployment in clinical settings, where safety certification and human-robot coexistence must be rigorously evaluated, remains the most critical next step.

CRediT authorship contribution statement

Tingjun Lei: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition; **Samuel Steen:** Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing – original draft, Writing – review & editing; **Chaomin Luo:** Conceptualization, Methodology, Formal analysis, Supervision, Resources, Writing – review & editing; **Jun Chen:** Methodology, Validation, Supervision, Writing – review & editing; **Yaqun Yuan:** Conceptualization, Validation, Investigation, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tingjun Lei reports financial support was provided by North Dakota NASA EPSCoR. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the North Dakota NASA EPSCoR Research Grant and the University of North Dakota Early Career Scholars Program.

Data availability

Data will be made available on request.

References

Alarabi, S., Lei, T., Santora, M., Luo, C., & Sellers, T. (2024). Multi-robot path planning using potential field-based simulated annealing approach. In *Spie conference: Unmanned systems technology xxvi* (pp. 102–117). SPIE (*vol. 13055*).

Bernhard, L., Schwingenschlögl, P., Hofmann, J., Wilhelm, D., & Knoll, A. (2024). Boosting the hospital by integrating mobile robotic assistance systems: A comprehensive classification of the risks to be addressed. *Autonomous Robots*, *48* (1), 1. https://doi.org/10.1007/s10514-023-10154-0

Black, B., Sellers, T., Lei, T., Luo, C., & Carruth, D. W. (2024). Optimal multi-target navigation via graph-based algorithms in complex environments. In *2024 IEEE 33rd International symposium on industrial electronics (ISIE)* (pp. 1–6). IEEE.

Chakraa, H., Guérin, F., Lefebvre, D., & Rahmoune, E. (2023). Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems*, *168*, 104492. https://doi.org/10.1016/j.robot.2023.104492

Chintam, P., Lei, T., Osmanoglu, B., Wang, Y., & Luo, C. (2024). Informed sampling space driven robot informative path planning. *Robotics and Autonomous Systems*, *175*, 104656.

Choubey, P., Shi, Q., Yang, Z., Terry, B., Shao, S., & Lei, T. (2025). Deep isolation forest-based anomaly behavior analysis for internet of medical things. In *2025 Cyber awareness and research symposium (CARS)* (pp. 1–7). IEEE.

Cruz, L. D., Lei, T., Luo, C., Bi, Z., & Jan, G. E. (2025). Advanced multi-target navigation using ADD-RRVS with double Q-learning. In *2025 International conference on information and automation (ICIA)* (pp. 448–453). IEEE.

Cui, Y., Hu, W., & Rahmani, A. (2024). Multi-robot path planning using learning-based artificial bee colony algorithm. *Engineering Applications of Artificial Intelligence*, *129*, 107092.

Devanga, A., Lei, T., Hu, J., Luo, C., & Shao, S. (2025). Deep reinforcement learning for USV navigation in dynamic ocean environments. In *2025 IEEE International conference on robotics and biomimetics (ROBIO)* (pp. 1024–1029). IEEE.

Faigl, J., & Hollinger, G. A. (2018). Unified multi-goal path planning for a data collection tour with multiple traveling salesman problem. *IEEE Robotics and Automation Letters*, *3* (3), 2016–2023.

Hicks, M., Lei, T., Luo, C., Carruth, D. W., & Bi, Z. (2025a). A bio-inspired goal-directed cognitive map model for robot navigation and exploration. *IEEE Transactions on Cognitive and Developmental Systems*, *17* (5), 1125–1140.

Hicks, M., Lei, T., Sellers, T., Luo, C., & Bi, Z. (2025b). Secure cognitive mapping and neural informed RRT* for robust path planning. In *2025 Cyber awareness and research symposium (CARS)* (pp. 1–8). IEEE.

Hicks, M., Lei, T., Sellers, T., Luo, C., & Yang, E. (2025c). A neural field approach to robot navigation with brain-inspired goal-directed cognitive maps. In *2025 IEEE International conference on robotics and biomimetics (ROBIO)* (pp. 269–276). IEEE.

Jan, G. E., Lei, T., Sun, C.-C., You, Z.-Y., & Luo, C. (2024). On the problems of drone formation and light shows. *IEEE Transactions on Consumer Electronics*, *70* (3), 5259–5268.

Janati, F., Abdollahi, F., Ghidary, S. S., Jannatifar, M., Baltes, J., & Sadeghnejad, S. (2017). Multi-robot task allocation using clustering method. In *Robot intelligence technology and applications 4: Results from the 4th International conference on robot intelligence technology and applications* (pp. 233–247). Springer.

Jayaraman, E., Lei, T., Rahimi, S., Cheng, S., & Luo, C. (2021). Immune system algorithms to environmental exploration of robot navigation and mapping. In *Advances in swarm intelligence: 12th international conference, ICSI 2021* (pp. 73–84). Springer.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

Kong, X., Zhou, Y., Li, Z., & Wang, S. (2024). Multi-UAV simultaneous target assignment and path planning based on deep reinforcement learning in dynamic multiple obstacles environments. *Frontiers in Neurorobotics*, *17*, 1302898.

Kyrarini, M., Lygerakis, F., Rajavenkatanarayanan, A., Sevastopoulos, C., Nambiappan, H. R., Chaitanya, K. K., Babu, A. R., Mathew, J., & Makedon, F. (2021). A survey of robots in healthcare. *Technologies*, *9* (1), 8. https://doi.org/10.3390/technologies9010008

Lei, T., Luo, C., Sellers, T., Wang, Y., & Liu, L. (2022). Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots. *IEEE Transactions on Aerospace and Electronic Systems*, *58* (6), 5129–5140.

Lei, T., Luo, C., Yang, S. X., Carruth, D. W., & Bi, Z. (2024). Bio-inspired intelligence-based multi-agent navigation with safety-aware considerations. *IEEE Transactions on Artificial Intelligence*, *5* (6), 2946–2961.

Lei, T., Sellers, T., Luo, C., Carruth, D. W., & Bi, Z. (2023). Graph-based robot optimal path planning with bio-inspired algorithms. *Biomimetic Intelligence and Robotics*, *3* (3), 100119.

Lei, T., Sellers, T., Rahimi, S., Cheng, S., & Luo, C. (2021). A nature-inspired algorithm to adaptively safe navigation of a Covid-19 disinfection robot. In *Intelligent robotics and applications: 14th international conference, ICIRA 2021* (pp. 123–134). Springer.

Li, H., Chu, Z., Fang, Y. et al. (2023). Source-seeking multi-robot team simulator as container of nature-inspired metaheuristic algorithms and A* algorithm. *Expert Systems with Applications*, *233*, 120932. https://doi.org/10.1016/j.eswa.2023.120932

Lin, S. et al. (2024). An improved fault-tolerant cultural-PSO with probability for multi-AGV path planning. *Expert Systems with Applications*, *236*, 121684.

Liu, D., Zhang, Y., Ren, Y., Pan, D., He, X., Cong, M., & Yu, G. (2025). Multi-directional attention: A lightweight attention module for slender structures. *Intelligence & Robotics*, *5* (4), 827–843.

Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, *227*, 120254. https://doi.org/10.1016/j.eswa.2023.120254

London, J., Lei, T., Luo, C., & Yuan, Y. (2025a). Human activity recognition and biomedical signal classification using multi-modal deep learning. In *2025 IEEE 68th International midwest symposium on circuits and systems (MWSCAS)* (pp. 676–680). IEEE.

London, J., Lei, T., Yuan, Y., Shao, S., & Hu, J. (2025b). Enhanced biomedical signal processing using time-frequency analysis and wavelet transforms. In *2025 IEEE 68th International midwest symposium on circuits and systems (MWSCAS)* (pp. 463–467). IEEE.

Nash, A., Daniel, K., Koenig, S., & Felner, A. (2010). Lazy theta*: Any-angle path planning and path length analysis in 3D. In *Proceedings of the AAAI conference on artificial intelligence*. (*vol. 24*).

Riser, E., Sellers, T., Lei, T., Chintam, P., & Luo, C. (2024). Multirobot navigation using improved RRT*-SMART with digital twin technology. In *Spie conference: Autonomous systems: Sensors, processing, and security for ground, air, sea, and space vehicles and infrastructure 2024* (pp. 114–125). SPIE (*vol. 13052*).

Rogers, J. H., Sellers, T., Lei, T., Carruth, D. W., & Luo, C. (2023). Sensor-based multi-waypoint autonomous robot navigation with graph-based models. In *Spie conference: Autonomous systems: Sensors, processing and security for ground, air, sea, and space vehicles and infrastructure 2023* (pp. 215–224). SPIE (*vol. 12540*).

Rogers, J. H., Sellers, T., Lei, T., Hudson, C. R., & Luo, C. (2024). Centroid-based cell decomposition robot path planning algorithm integrated with a bio-inspired approach. In *Spie conference: Unmanned systems technology xxvi* (pp. 43–51). SPIE (*vol. 13055*).

Sellers, T., Lei, T., Luo, C., Bi, Z., & Jan, G. E. (2024). Human autonomy teaming-based safety-aware navigation through bio-inspired and graph-based algorithms. *Biomimetic Intelligence and Robotics*, *4* (4), 100189.

Sellers, T., Lei, T., Luo, C., Yang, E., & Bi, Z. (2026). Human-autonomy teaming in robot navigation using bioinspired approach for search and rescue. *IEEE Transactions on Human-Machine Systems*, *56* (2), 311–320.

Sellers, T., Lei, T., Rogers, J. H., Carruth, D. W., & Luo, C. (2023). Autonomous multi-robot allocation and formation control for remote sensing in environmental exploration. In *Spie conference: Autonomous systems: Sensors, processing and security for ground, air, sea, and space vehicles and infrastructure 2023* (pp. 225–241). SPIE (*vol. 12540*).

Short, D., Lei, T., Carruth, D. W., Luo, C., & Bi, Z. (2023). A bio-inspired algorithm in image-based path planning and localization using visual features and maps. *Intelligence & Robotics*, *3* (2), 222–241.

Short, D., Sellers, T., Lei, T., Gao, J., & Luo, C. (2025). Market-based multi-robot task allocation in limited-communication environments. In *2025 Cyber awareness and research symposium (CARS)* (pp. 1–6). IEEE.

Steen, S., Lei, T., Luo, C., Li, G., & Shao, S. (2025a). Fail-safe multi-robot awareness during GPS outages using CNN-based LDAR. In *2025 Cyber awareness and research symposium (CARS)* (pp. 1–7). IEEE.

Steen, S., Lei, T., Luo, C., Shao, S., & Gong, L. (2025b). A moss growth optimization approach to robot path planning. In *2025 IEEE Congress on evolutionary computation (CEC)* (pp. 1–4). IEEE.

Swapna, B., & Murali Mohan, V. (2024). Dynamic object detection using sparse LiDAR data for autonomous machine driving and road safety applications. *Expert Systems with Applications*, *255*, 124513. https://doi.org/10.1016/j.eswa.2024.124513

Wu, L., Xiao, W. et al. (2023). Dynamic path planning of mobile robot based on improved simulated annealing algorithm. *Expert Systems with Applications*, *215*, 119410. https://doi.org/10.1016/j.eswa.2022.119410

Xie, X., Wei, H., & Yang, Y. (2023). Real-time LiDAR point-cloud moving object segmentation for autonomous driving. *Sensors*, *23*(1), 547. https://doi.org/10.3390/s23010547

Yang, Y., Jiang, M., & Li, W. (2020). 2D path planning by lion swarm optimization. In *2020 4th International conference on robotics and automation sciences (ICRAS)* (pp. 117–121). IEEE.

Zheng, B., Chen, Y., Wang, C., Heidari, A. A., Liu, L., & Chen, H. (2024). The moss growth optimization (MGO): Concepts and performance. *Journal of Computational Design and Engineering*, *11* (5), 184–221.

Zhong, S., Chen, T., Huang, K., Liu, X., Liu, Y., & Liu, M. (2024). DCL-SLAM: A distributed collaborative LiDAR SLAM framework for a robotic swarm. *IEEE Sensors Journal*, *24* (4), 4786–4797. https://doi.org/10.1109/JSEN.2023.3345541