Biomimetic Intelligence and Robotics 4 (2024) 100189

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

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

Research Article

Human autonomy teaming-based safety-aware navigation through bio-inspired and graph-based algorithms

Timothy Sellers [a], Tingjun Lei [a], Chaomin Luo [a,*,1], Zhuming Bi [b], Gene Eu Jan [c]

[a] Department of Electrical and Computer Engineering, Mississippi State University, Starkville 39762, USA
[b] Department of Civil and Mechanical Engineering, Purdue University Fort Wayne, Fort Wayne 46805, USA
[c] Department of Computer Science, Asia University, Taichung 413305, Taiwan, China

ARTICLE INFO

Article history:
Received 30 June 2024
Revised 3 September 2024
Accepted 26 September 2024
Available online 18 October 2024

Keywords:
Human autonomy teaming (HAT)
Robot path planning
Generalized Voronoi diagram (GVD)
Spatio-temporal graphs
Bio-inspired algorithms

ABSTRACT

In the field of autonomous robots, achieving complete precision is challenging, underscoring the need for human intervention, particularly in ensuring safety. Human Autonomy Teaming (HAT) is crucial for promoting safe and efficient human–robot collaboration in dynamic indoor environments. This paper introduces a framework designed to address these precision gaps, enhancing safety and robotic interactions within such settings. Central to our approach is a hybrid graph system that integrates the Generalized Voronoi Diagram (GVD) with spatio-temporal graphs, effectively combining human feedback, environmental factors, and key waypoints. An integral component of this system is the improved Node Selection Algorithm (iNSA), which utilizes the revised Grey Wolf Optimization (rGWO) for better adaptability and performance. Furthermore, an obstacle tracking model is employed to provide predictive data, enhancing the efficiency of the system. Human insights play a critical role, from supplying initial environmental data and determining key waypoints to intervening during unexpected challenges or dynamic environmental changes. Extensive simulation and comparison tests confirm the reliability and effectiveness of our proposed model, highlighting its unique advantages in the domain of HAT. This comprehensive approach ensures that the system remains robust and responsive to the complexities of real-world applications.

1. Introduction

Autonomous robots are increasingly being used in indoor spaces like airports, hospitals, and museums [1,1–4]. Indoor path planning [5–9] and [10] involves navigating through dynamic, crowded areas while avoiding obstacles. Effective navigation [11–14] in these areas requires *more than just* exploration; it needs to integrate smoothly with human presence. Relying solely on sensory data or human input can limit navigation efficiency. Human Autonomy Teaming (HAT) offers a balanced approach, cooperating human expertise and robotic autonomy without one overshadowing the other. This collaboration promises improved performance, particularly in uncertain conditions. However, developing effective motion planning strategies within HAT for indoor settings remains a key area of ongoing research.

In the rapidly evolving field of robotic path planning, researchers are pioneering diverse methodologies to enhance the efficiency and capability of robots in navigating complex environments. For instance, Srivastava et al. [15] delved into the realm of autonomous robotics, specifically targeting path planning robots essential for industrial and autonomous systems. Their innovative approach incorporates OpenCV for robust image processing alongside hardware components such as the Raspberry Pi 4, Arduino Uno, motor drivers, and a Pi-Cam. This combination aims to significantly enhance robot efficiency in industrial tasks through sophisticated image analysis and canny edge detection algorithms. In dynamic settings, the challenge of mobile robot path planning is adeptly addressed by Han and Seo [16], who proposed an efficient two-step methodology to generate collision-free paths. Initially, the surrounding point set (SPS) method strategically identifies key points around obstacles to form a viable path, which is subsequently refined by a path improvement algorithm (PIFLP) to optimize for distance and smoothness, adjusting each point relative to its neighbors for optimal navigation. Gandia et al. [17] explored the integration of design and fabrication processes in the realm of robotically assembled structures, presenting a software interface for automatic robot path planning. Their work focuses on refining path search parameters and integrating design and fabrication processes, broadening

* Corresponding author.
E-mail address: Chaomin.Luo@ece.msstate.edu (C. Luo).

1 Given his role as Associate Editor of this journal, Chaomin Luo had no involvement in the peer-review of this article and had no access to information regarding its peer-review. This article was handled by Dr. Long Wang.

https://doi.org/10.1016/j.birob.2024.100189

the possibilities for complex robotic assemblies. Adding a novel perspective to the discussion, Sidler et al. [18] introduced an emotion-influenced approach to robotic path planning. Their method enables robots to adjust their speed and learn preferred locations based on emotional states like happiness and fear, offering a significant deviation from traditional path planning methods devoid of emotional considerations. The efficiency and energy consumption of multi-robot systems are scrutinized by Nair and Supriya [19], who focused on cooperative behavior for path planning in known environments. Emphasizing the importance of robots assisting each other during malfunctions, their approach employs reinforcement learning, specifically the Temporal Difference (TD) algorithm, to enhance multi-robot efficiency. Theja and Naidu [20] developed a memory-leveraging robot path planning method for improved efficiency in autonomous navigation, significantly reducing computational demands for robots in diverse applications. This method showcases the potential for efficient path planning by storing successful paths, thus avoiding repeated path forecasting for familiar locations.

Addressing the pursuit scenarios, Drake et al. [21] proposed an incremental moving target path planning algorithm, prioritizing speed and simplifying re-planning with minimal impact on path length, crucial for unmanned vehicles. Furthermore, the critique by Dam et al. [22] on the limitations of Rapidly-exploring Random Tree (RRT) algorithms in complex environments leads to the introduction of the Monte-Carlo Path Planning (MCPP) approach. This innovative strategy applies the Monte-Carlo tree search to robot path planning, demonstrating enhanced performance in various scenarios, including fully and partially observable Markov decision processes. Lastly, Zhang et al. [23] tackled the inefficiencies of traditional RRT algorithms in complex industrial robot path planning by proposing an enhanced RRT algorithm. This algorithm introduces a regression mechanism to minimize over-searching and an adaptive expansion mechanism to refine boundary nodes, improving the spatial information in the robotic manipulator's joint space, marking another step forward in the continuous evolution of path planning methodologies.

Graph-based path planning offers a structured approach to navigating complex environments, and recent research has introduced various innovative algorithms to optimize this process for different robotic applications. These methods range from enhancing navigation in shared workspaces to exploring subterranean networks and improving efficiency in dynamic settings. Lyu et al. [24] have developed a novel graph-based path planning algorithm for mobile robots, incorporating the Floyd algorithm for dynamic weight assignment. This method optimizes collision-free navigation in shared workspaces, proving effective and robust through extensive testing on robot path datasets, promising significant advancements for industrial robot operations. Dang et al. [25] and in a subsequent work [26], proposed strategies for autonomous path planning in complex subterranean environments. Their approach uses a dual planner architecture, with a local planner employing a rapidly-exploring random graph for immediate surroundings and a global planner addressing broader navigation challenges. This bifurcated method is optimized for the intricate networks of tunnels found underground, suitable for both aerial and legged robots. Belanová et al. [27] explored autonomous path planning without human intervention, utilizing the D* Lite algorithm to enable a Pepper robot to find the shortest path in a known static environment. This research demonstrates the feasibility of graph-based algorithms for robotic movement, marking a step towards developing intelligent workspaces. Liu and Jiang [28] introduced a robot path planning algorithm using a triangular grid graph for navigating complex maps. By converting maps into triangular meshes, they create a weighted undirected graph that utilizes the Dijkstra algorithm for finding the shortest path, subsequently refined by the Douglas–Peucker algorithm for enhanced efficiency and navigation clarity.

Diao et al. [29] presented a learning-based path planning method employing Graph Neural Networks (GNN) to reduce collision detection operations. Their model efficiently processes environmental maps to guide obstacle avoidance, demonstrating significant reductions in collision checks and accelerated path planning in complex settings. Sabudin et al. [30] tackled the limitations of established path planning techniques by introducing a dynamic artificial potential field (DAPF) algorithm. This approach effectively overcomes the issue of local minima and incorporates path pruning for streamlined navigation. Clawson et al. [31] offered an algorithm for bi-criteria path planning, optimizing multiple criteria simultaneously. Their method augments the state space to accommodate secondary cost budgets, facilitating primary cost minimization through an innovative upward sweep. Lee and Kim [32] enhanced genetic algorithm-based robot path planning with an effective initialization method. This addresses a critical gap in existing research, underscoring the importance of initialization in genetic algorithms for path planning. Finally, Ganeshmurthy and Suresh [33] addressed mobile robot path planning in dynamic environments, proposing a novel approach that combines a heuristic method with simulated annealing. This method optimizes path length and demonstrates improvements in runtime and efficiency, showcasing its effectiveness against both stationary and moving obstacles.

The burgeoning field of human–robot interaction (HRI) is increasingly focused on creating synergistic relationships between humans and robots, aiming to leverage the unique strengths of both to achieve combined goals, such as Zhao et al. [34]. This area of research spans across various applications, including manufacturing, rescue operations, and even individual entertainment settings, emphasizing the optimization of shared control, autonomy, and mutual adaptation to enhance collaboration and task efficiency. Musić and Hirche [35] delved into the integration of human and robot capabilities within team settings, exploring how optimal shared control can leverage the human's strategic planning abilities and the robot's precision. Their work sets the stage for deeper exploration into collaborative frameworks that enhance team performance in critical applications. Ajmera [36] took this concept further by proposing a shared autonomy model implemented through a web interface. This model combines human input with robotic autonomy, aimed at boosting performance while minimizing cognitive load by filtering and displaying only essential information.

Additionally, this framework introduces the notion of human–robot mutual adaptation, fostering a more effective collaborative environment. Comparative analysis by Tokadlı and Dorneich [37] on interaction paradigms across human-autonomy teams highlights the importance of dynamic role allocation. By examining how these paradigms compare with human–human and human–computer interactions, they identify strategies that could lead to more efficient control and collaboration with autonomous systems. Nikolaidis et al. [38] specifically addressed the challenge of shared autonomy in enhancing human–robot teams. Their approach involves influencing humans towards more effective collaboration strategies, sometimes against their initial inclinations, while ensuring trust is maintained. They incorporate a mutual adaptation formalism within a decision-making framework that allows robots to adjust their behavior based on the human partner's actions. Beer et al. [39] proposed a framework to categorize Levels of Robot Autonomy (LORA), spanning from teleoperation to full autonomy. This framework aims to bridge insights from HRI and human-automation interaction, introducing a taxonomy to assess autonomy levels and their implications on user acceptance and reliability.

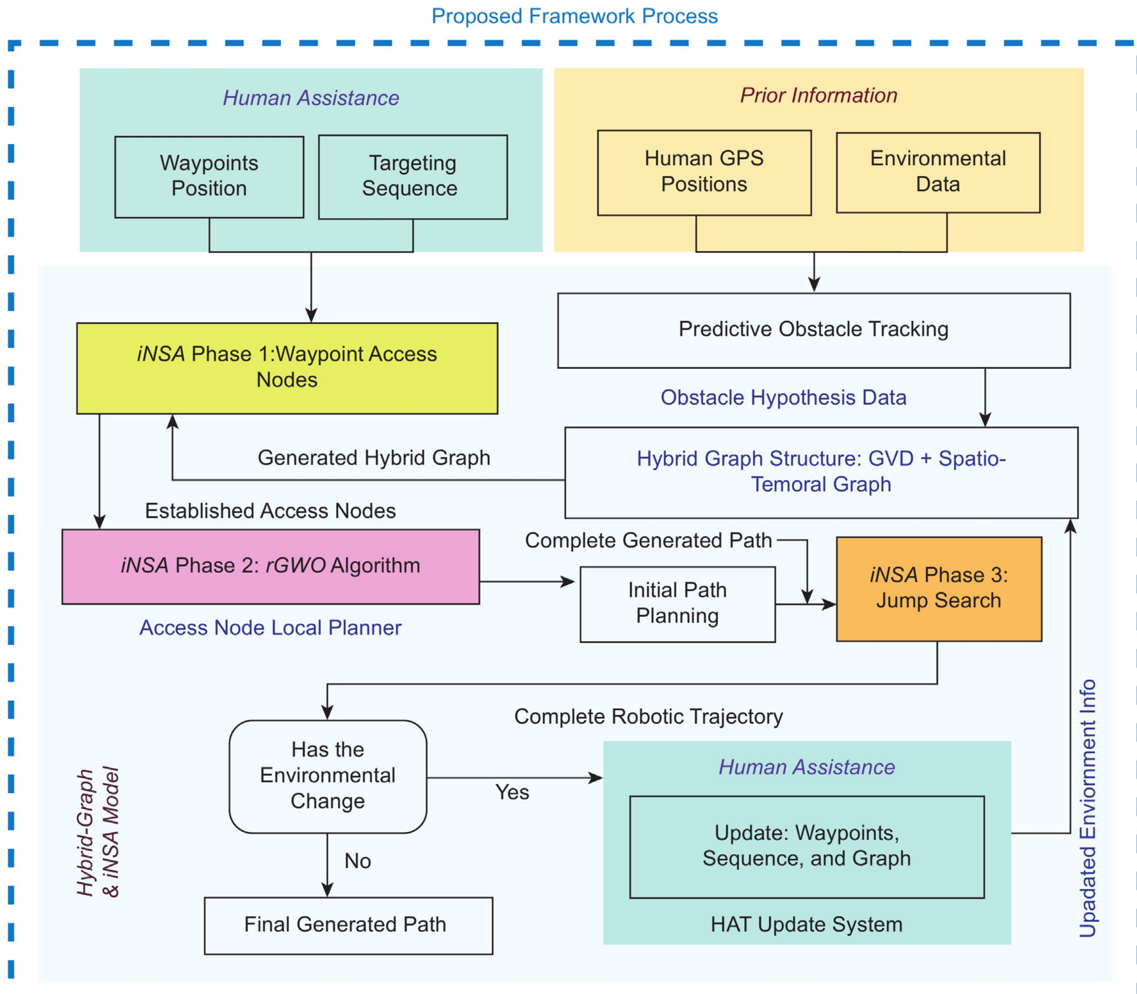

Fig. 1. Depicts a proposed framework for path generation. It integrates human positions, environmental data, and prior information to create a hybrid graph structure (GVD + spatio temporal graph). This structure undergoes three *iNSA* phases: waypoint access node establishment, path generation using the *rGWO* algorithm, and a jump search phase.

Scalise et al. [40] investigated the delicate balance between efficiency and cognitive load in human–robot communication. They emphasize the incorporation of human knowledge and preferences into robotic task execution, exploring how language analysis in shared autonomy settings can illuminate the trade-offs considered by individuals to maintain cooperative yet manageable interactions. Pérula-Martínez et al. [41] focused on enhancing social robot autonomy with a bio-inspired decision-making system for individual entertainment settings. This system utilizes interactive tools to autonomously make decisions that align with user preferences, pushing the boundaries of intelligent, responsive HRI. Miller et al. [42] described a system that integrates human intentions into robotic autonomy through vision-based human pose estimation. This approach translates human gestures into commands for robots, detailing solutions for autonomy in mobile robots and showcasing successful implementations in human-following behaviors and automated decision-making for parking. Selvaggio et al. [43] highlighted the role of shared control (SC) and shared autonomy (SA) in reducing human workload in robotic operations. They discuss how advancements in inference and learning techniques have broadened the applications of SC, allowing robots to dynamically adapt their autonomy levels, particularly in physical HRI contexts. Lastly, Cantucci et al. [44] advocated for robots capable of intelligently adapting to human needs by understanding and acting upon human mental states. They proposed a cognitive architecture that enables robots to attribute beliefs and intentions to humans, fostering an enhanced level of trust and collaboration in HRI.

This paper introduces a methodology grounded in HAT to enhance the navigation and interaction between autonomous robots and humans in complex indoor environments, with a strong emphasis on safety. It underscores the essential role of human input, from providing initial environmental data and setting key waypoints to adapting to unforeseen challenges. The autonomous system is designed to respond dynamically to both environmental information and human instructions. Additionally, the methodology incorporates an advanced obstacle tracking method, enabling the system to detect and respond to obstacles in real-time. By leveraging the strengths of both humans and robots, the approach aims to create a more adaptable and resilient navigation system. *Human* inputs are critical for mapping the environment and defining navigation goals, while the autonomous system processes these inputs and adjusts its behavior based on evolving conditions. This collaborative and dynamic approach ensures that the system can handle the complexities and unpredictability of indoor spaces, ultimately enhancing safety and efficiency in human–robot interactions. The proposed framework is illustrated in Fig. 1. The contributions of this paper are as follows:

- A hybrid graph-based structure in correlation with an improved node selection algorithm is proposed to enhance

the interaction between humans and autonomous robots in indoor environments. This approach combines traditional graph-based methods with advanced algorithms, focusing on improving communication and cooperation in dynamic and confined spaces.
- The study is performed for custom-tailored HAT methodologies to suit the dynamic and evolving nature of indoor scenarios, emphasizing adaptability and real-time node selection decision-making. The developed methodologies are specifically designed for indoor environments, ensuring that the system can adapt to changes and make decisions in real-time.
- A groundbreaking hybrid graph structure is developed that merges a Generalized Voronoi Diagram (GVD) with spatio-temporal graphs, offering a unique solution for complex spatial representation. This structure integrates GVD for accurate spatial partitioning and uses spatio-temporal graphs to manage time-dependent data and dynamic changes.
- An improved node selection algorithm (iNSA) is developed to enhance path efficiency within the hybrid graph structure, addressing the critical challenge of optimizing traversal in intricate environments. The algorithm focuses on selecting optimal nodes for efficient pathfinding and reduces the computational complexity of traversing complex environments.

The structure of this paper is as follows: An overview of the proposed framework is presented in Section 2. In Section 3, a hybrid graph structure is proposed to improve the mapping and initial path planning of the model. A iNSA algorithm is introduce in Section 4, to improve the generation of access nodes into the graph as well as improve path planning capabilities. In Section 5, a obstacle traction methodology is utilized to improve the HAT-based path planning and decision making. Simulation and Comparative analysis are presented in Section 6. Lastly, our conclusion of the proposed frame work can be found in Section 7.

2. Human autonomy teaming used for navigation

In modern robotic systems, particularly in dynamic environments such as hospitals and supermarkets, it is essential to understand and adapt to the surrounding context. The model introduced here exemplifies an innovative blend of algorithmic intelligence and human expertise, enhancing the overall safety of the framework. At its core is a hybrid graph structure that integrates the GVD with spatio-temporal graphs. This model employs an iNSA in conjunction with HAT to optimize the selection of nodes and edge nodes, thereby forming a near-optimal path. Additionally, the structure is refined using the revised Grey Wolf Optimization (rGWO) algorithm, ensuring efficient path determination and navigation. The effectiveness of the proposed framework is demonstrated in Fig. 2 and Table 1. The core strength of this model lies in its HAT approach. Instead of the robot solely depending on pre-programmed algorithms and sensors, human expertise is integrated at every necessary stage, adapting to updates in the dynamic environment. Human insights are essential for providing initial environmental data, setting key waypoints, and intervening during unexpected challenges or changes. This integration enhances the system's adaptability and safety, ensuring the robot's navigation is optimal and synchronized with the ever-changing indoor dynamics.

3. HAT hybrid graph structure

Graph-based mapping is crucial in various academic and practical fields due to its ability to represent and analyze complex systems through nodes and edges, capturing the intricate relationships between entities. This methodology is essential for visualizing and understanding the structure and dynamics of systems such as social networks, biological pathways, transportation infrastructures, and technological ecosystems. The utilization of hybrid methods in graph-based mapping significantly enhances analytical capabilities by combining diverse algorithms and techniques. These hybrid approaches leverage the strengths of multiple methods, such as integrating clustering algorithms with distance-based edge connections, to produce more accurate and efficient analyses. This is particularly advantageous in managing large, heterogeneous datasets, where single-method approaches often fall short. By employing hybrid methods, researchers can achieve more robust and scalable solutions, facilitating deeper insights and advancements in knowledge. Thus, hybrid graph-based mapping serves as a versatile and powerful tool in academic research, enabling the exploration and resolution of complex problems across various disciplines.

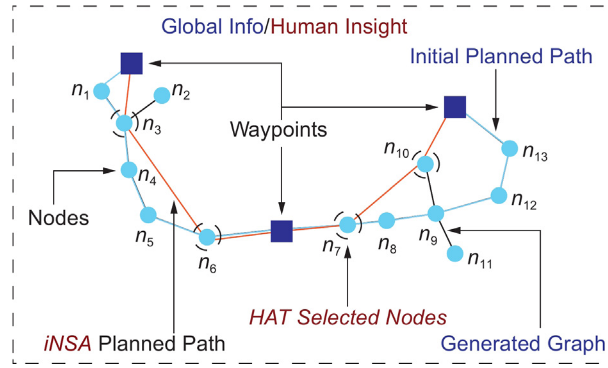

Fig. 2. Illustration of a robot's initial path in an indoor setting, created by linking a hybrid graph and an *iNSA* algorithm. Human insights are then applied to select crucial nodes (n_7 to n_{10}) from an HAT *Permission Token List (PTL)*, optimizing the path, as illustrated by the black dashed circles. The initial path (light-blue line) and the waypoints (olive colored squares) show how human adjustments improve the iNSA's planned route (tangerine line).

Table 1
The table depicts the nodes and edge nodes utilized in the HAT node selection process. Here "✗" refers to nodes being rejected by the human operator, and "✓" means that the node was selected.

Feature	1	2	3	4	5	6	7	8	9	10
Nodes	n_1	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{12}	n_{13}
Decision	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗

3.1. Generalized Voronoi diagram (GVD)

The GVD is an advanced adaptation of the classical Voronoi diagram, which is foundational in the realm of computational geometry for partitioning spaces, such as in Wang and Meng [45],Chen et al. [46] proposed frameworks. Traditional Voronoi diagrams segment a space into regions based on the closest proximity to a set of predefined points, known as seeds or sites. The GVD extends this concept beyond simple points and Euclidean distances, accommodating a broader array of site types such as lines, curves, or even polygons, and can utilize different distance metrics to define the nearest site. The GVD graph can be seen in Fig. 3. The GVD is formulated within a space X and involves a set of entities $S = \{s_1, s_2, \ldots, s_n\}$, where each entity s_i represents a site that could be of any geometric shape. The diagram partitions the space into regions $V(s_i)$, where each region corresponds to one site and consists of all points in X that are closest to that particular site compared to all other sites, under a chosen distance

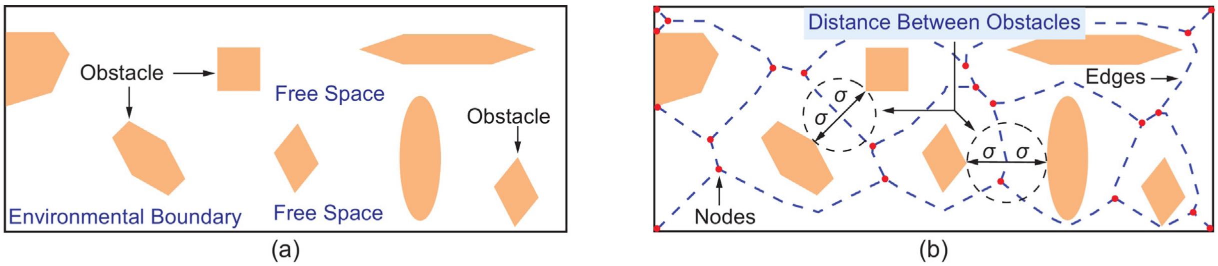

Fig. 3. Illustration of the GVD algorithm: (a) shows a simple robot environment, while (b) demonstrates the algorithm generating nodes and edges. Blue lines and red circles represent nodes and edges in the GVD graph, used for initial path planning. The parameter σ indicates the equal distance from nearby obstacles to form the graph's edges.

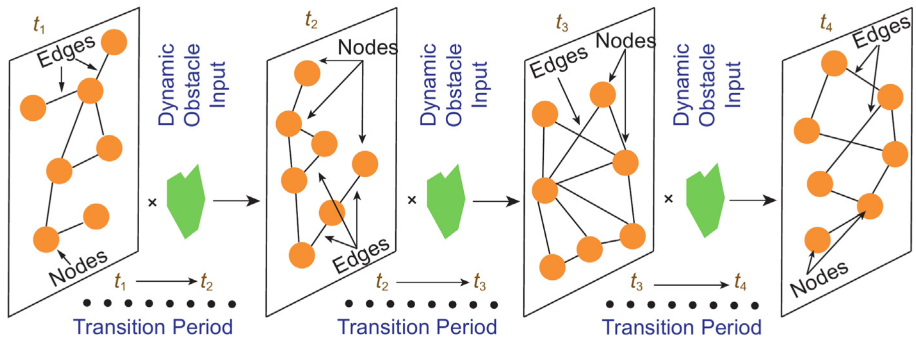

Fig. 4. Illustration of the spatio-temporal graph process for time based graph reconstruction. As the dynamic obstacle travels through the workspace the graph updates to move around the obstacle.

metric d. Mathematically, this is represented as

$$V(s_i) = \{x \in X : d(x, s_i) \leq d(x, s_j) \forall s_j \in S, j \neq i\} \quad (1)$$

This definition underscores the GVD's adaptability to various contexts by allowing for different types of sites and distance functions, thereby broadening the scope of applications significantly beyond those of traditional Voronoi diagrams. The properties of GVDs are an extension of those found in traditional Voronoi diagrams, but with greater complexity due to their generalized nature. The regions formed by a GVD are connected, ensuring a continuous partitioning of the space that is integral for applications requiring spatial coherence. The boundaries between these regions are defined by the bisectors that are inherently determined by the chosen distance metric and the geometrical properties of the sites. This characteristic is pivotal for the GVD's application in fields that require an understanding of spatial relationships and territorial boundaries. Moreover, the GVD's flexibility in defining what constitutes a site, along with the choice of distance metric, allows for customized solutions tailored to specific problems in computational geometry, making it an indispensable tool in a wide array of scientific and engineering disciplines.

3.2. Spatio-temporal graphs

Spatio-temporal graphs represent a sophisticated fusion of spatial and temporal dimensions, building upon the foundational principles of traditional graph theory, such as in Rowold et al. [47],Meliou et al. [48] proposed framework. These graphs extend beyond the static representation of relationships found in standard graphs by incorporating the element of time, thus offering a dynamic perspective on how spatial interactions evolve. In a spatio-temporal graph, the integration of time with spatial structures allows for the modeling of complex systems where both the location of entities and their temporal interactions are of paramount importance. This dual consideration facilitates a comprehensive analysis of systems where change over time is a crucial factor, enabling the examination of patterns, dynamics, and evolution in spatially defined networks. The formal definition of a spatio-temporal graph encapsulates the essence of its structure, represented as

$$G = (V, E, T) \quad (2)$$

where V denotes the set of vertices or nodes, embodying the spatial components of the graph. The edges E connect these nodes, representing the spatial relationships or interactions between them. The temporal dimension T is incorporated to model the evolution of these interactions over time, which can be specified as discrete timestamps, continuous intervals, or any form of temporal metric that suits the system being modeled, as seen in Fig. 4. This temporal aspect transforms the graph into a dynamic entity, capable of capturing the transient and evolving nature of the underlying spatial phenomena. The temporal component T in a spatio-temporal graph is pivotal, as it introduces a dynamic aspect to the graph's topology, allowing for the representation of changes in the network structure over time. This can include the addition or removal of nodes and edges, changes in node or edge attributes, or the occurrence of events, all of which are temporally bound. The granularity of T can vary significantly, from high-resolution milliseconds to broader time spans like years, depending on the application's requirements and the nature of the data being analyzed. In essence, spatio-temporal graphs provide a powerful and versatile framework for analyzing and visualizing dynamic systems where spatial and temporal components are interlinked. Their ability to encapsulate changes over time in the context of spatial relationships makes them invaluable in various fields such as transportation networks, social network analysis, epidemiology, and environmental modeling.

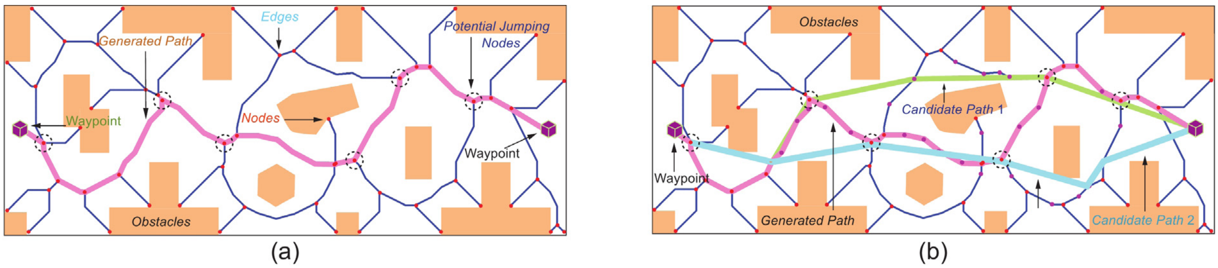

Fig. 5. Illustration of the proposed methodology path planning technique both pre and post of the overall process. (a) depicts the initial path planning method to generate a basic planned path. The green line illustrates the initial planned path via the hybrid graph. (b) illustrates the proposed methodology generated planned path. The blue and orange lines illustrate candidate paths found utilizing the initial planned path. The red circles with black dashed lines around it are potential jumping nodes shown tot he human operator. .

3.3. Unified framework for hybrid analysis

The unified framework for hybrid analysis leverages the strengths of the GVD and spatio-temporal graphs to create a robust and dynamic mapping system capable of adapting to complex and evolving environments. The framework begins by dividing the environment into distinct sections using the GVD, a method that efficiently segments space based on proximity to significant features, or landmarks, within the environment. Let the space be represented by X, containing a set of landmarks $S = \{s_1, s_2, \ldots, s_n\}$, where each landmark s_i represents a critical point, such as walls, obstacles, or key areas of interest. The GVD partitions the space into non-overlapping regions $V(s_i)$, with each region corresponding to a specific landmark and defined mathematically as

$$V(s_i) = \{x \in X : d(x, s_i) \leq d(x, s_j)\, \forall s_j \in S, j \neq i\} \tag{3}$$

where $d(x, s_i)$ represents the distance from point x to the landmark s_i, calculated using a chosen distance metric d, such as Euclidean or Manhattan distance. The resulting regions $V(s_i)$ form distinct sections within the environment, each of which is conceptualized as a "super node" $N_{\text{super}} = \{N_1, N_2, \ldots, N_n\}$ in the graph $G = (V, E)$. Within each section created by the GVD, nodes are further categorized based on their specific roles and importance in the graph structure. The super nodes N_{super} represent the entire sections derived from the GVD and serve as the foundational components of the graph. Edge nodes N_{edge} are generated at the boundaries of these sections, specifically at the intersection of bisectors between adjacent regions. These edge nodes are crucial for capturing the interactions between neighboring sections and are mathematically defined as

$$N_{\text{edge}} = \{x \in X : d(x, s_i) = d(x, s_j)\, \text{for some } i \neq j\} \tag{4}$$

where x represents the points equidistant from the adjacent landmarks s_i and s_j. Additionally, internal nodes N_{int} are identified within each section, representing significant points such as areas of high traffic or locations of critical environmental features. The generation of internal nodes is governed by a function f, which evaluates the internal importance of points within the section. Mathematically, these nodes are expressed as

$$N_{\text{int}} = \{x \in V(s_i) : f(x) > \tau\} \tag{5}$$

where τ is a threshold parameter determined by the specific application, such as traffic density, sensor data, or other criteria relevant to the system being modeled. The hybrid graph $G(t) = (V(t), E(t))$ is inherently dynamic, evolving over time as the environment changes. The temporal component t introduces the ability to adapt the graph structure in real-time, ensuring that it accurately reflects the current state of the environment. The positions of nodes, whether they are super nodes, edge nodes, or internal nodes, may shift in response to dynamic changes such as the movement of obstacles, changes in environmental conditions, or alterations in the importance of certain areas. For any node $n \in V(t)$, its position is updated by a function $u(n, \Delta t)$ that models the environmental changes over a time interval Δt, leading to the updated position

$$n(t + \Delta t) = n(t) + u(n, \Delta t) \tag{6}$$

This function u encapsulates the effects of various factors, such as the velocity of moving obstacles, the influence of environmental forces, or user interactions, making the framework highly adaptable to different scenarios. Furthermore, the edges $E(t)$ connecting the nodes in the graph are dynamically adjusted based on proximity and interaction criteria, ensuring that the connectivity within the graph remains relevant to the current state of the environment. An edge e_{ij} between nodes n_i and n_j is created or removed depending on whether the nodes remain within a specified interaction distance d_{int}. The edge $e_{ij}(t)$ is defined as

$$e_{ij}(t) = 1 \text{ if } d(n_i(t), n_j(t)) \leq d_{\text{int}}, \text{ and } e_{ij}(t) = 0 \text{ otherwise} \tag{7}$$

indicating that the nodes are close enough to interact, and the edge does not exist if they are not within the interaction distance. This mechanism allows the graph to reflect the dynamic nature of the environment, with edges appearing or disappearing as nodes move and interact differently over time. By integrating these components—GVD-based sectioning, dynamic node classification, and spatio-temporal graph updates—the unified framework provides a powerful and flexible tool for analyzing and managing complex environments. The framework's adaptability ensures that it can handle the intricacies of real-world scenarios, such as indoor navigation in large, busy spaces, where both spatial divisions and temporal dynamics play critical roles. This mathematically grounded approach enables the continuous refinement of the graph structure, supporting effective decision-making and optimization in various applications.

4. HAT-based improved node selection algorithm (iNSA) in a hybrid graph structure

The iNSA advances autonomous systems by blending human expertise with computer-based methods, enhancing decision-making. It identifies relevant nodes for Human Autonomy Teaming (HAT) processes and facilitates efficient path planning. The synergy of human contextual understanding and the rGWO algorithm optimizes problem-solving, significantly improving navigation in complex environments. This approach highlights the benefits of combining HAT insights with sophisticated algorithms, as depicted in Fig. 5.

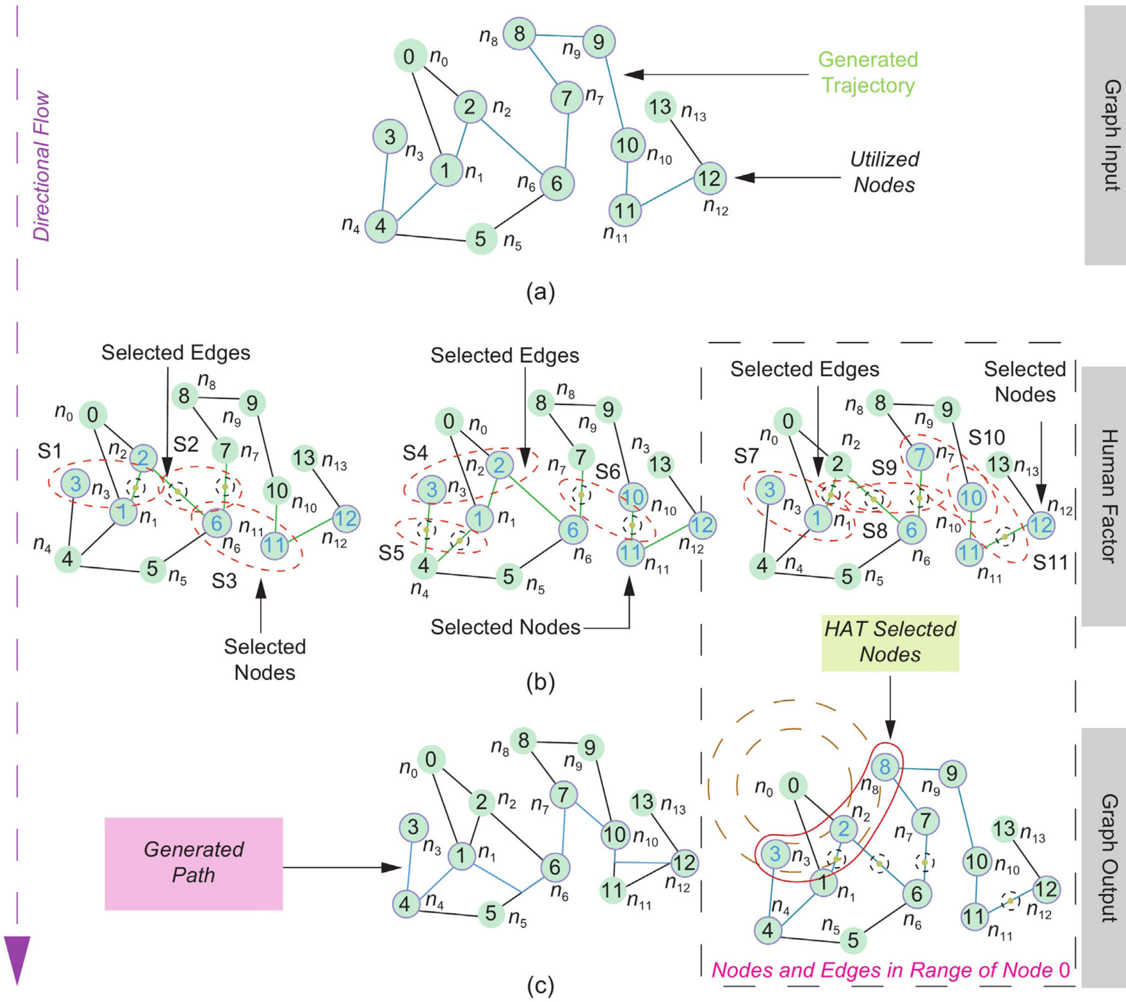

Fig. 6. Illustrates the influence of the human factor on hybrid graph-based path planning. (a) displays the initial path generated by the HAT algorithm, with nodes and edges selected based on proximity and connectivity. (b) highlights adjustments made through human intervention, where nodes and edges are re-selected to address considerations like safety or environmental conditions. (c) presents the final path, integrating both algorithmic and human selections to produce an optimized path that balances computational efficiency with practical requirements.

4.1. Advanced access node and waypoint integration in node selection algorithm

Our advanced navigation system emphasizes the crucial role of HAT, integrating sophisticated algorithmic recommendations with human judgment. The system supports decision-making by providing a detailed table, known as the *Permission Token List (PTL)*, which includes potential traditional nodes and generalized edge points, as shown in Table 2 and Fig. 6. These candidates are rigorously assessed using a probabilistic approach, considering key metrics such as their distance from the robot, proximity to obstacles, and overall environmental context. The probabilistic relationship that determines the suitability of each node or edge is represented in the following equation:

$$p(n_i) = w_d \times p_d(n_i) + w_o \times p_o(n_i) \tag{8}$$

where $p_d(n_i)$ and $p_o(n_i)$ represent desirability and feasibility based on distance and obstacle proximity, with w_d and w_o serving as contextually adjustable weights. This intricate process forms the core of the node selection criteria, elaborated visually in the corresponding Fig. 7. Empowered with this information, the HAT undertakes the strategic selection of optimal nodes. This decision is informed by the algorithm's logical suggestions, tempered with the operator's experiential knowledge and strategic understanding. Following path selection, the system utilizes a jump search strategy, which prioritizes direct routes that are free from obstacles. If environmental conditions make this method unfeasible, the system switches to node path planning using a rGWO Algorithm. This alternative method, an improved version of the lion optimizer, dynamically adjusts the navigation path to effectively circumvent obstacles, ensuring smooth robot operation. The complete process is outlined in Algorithm 1. By combining algorithmically processed data with a visual decision-making framework, the system integrates computational methods with HAT strategies.

4.2. Augmented straight-line optimization: jump search

Initially, the methodology focuses on efficiency. If a direct route, indicated as P_{ij}, can be created between two non-adjacent nodes n_i and n_j, the robot can navigate this path without detouring through intermediate nodes. This straight vector is defined as

$$\vec{P_{ij}} = \vec{n_j} - \vec{n_i} \tag{9}$$

By maximizing the Euclidean distance $\|\vec{P_{ij}}\|$, the robot can quickly reach its destination, provided that this straight-line path is unobstructed, as shown in Fig. 8. First, consider the direct path P_{ij}

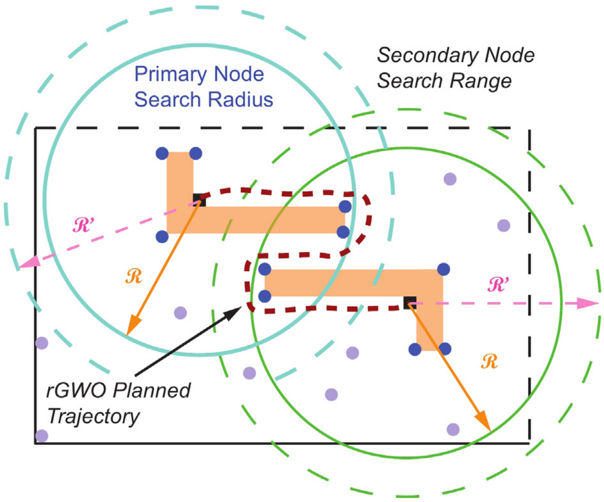

Fig. 7. Illustration of the iNSA algorithm determining near-optimal nodes suggestion to give to the human operator. Within the figure a path is constructed utilizing the rGWO algorithm to connect the targeted nodes.

between nodes n_i and n_j. Normally, P_{ij} would be assessed for obstructions to verify its feasibility. However, in dynamic environments, a deterministic evaluation might fall short. Therefore, we incorporate a probabilistic approach. Let O_k be a potential obstruction, and $p(O_k)$ be the likelihood of O_k intersecting with P_{ij} during traversal. The overall probability P_o of P_{ij} being obstructed is formulated as

$$P_o = 1 - \prod_k (1 - p(O_k)) \tag{10}$$

If P_o is less than a predetermined risk threshold T_r, the path is deemed viable. This probabilistic evaluation helps the robot assess risks and make well-informed decisions. To further refine decision-making, a learning mechanism is necessary for continuous improvement. After each journey, the robot records its experiences, noting the decisions made, the actual outcomes, and any deviations. Let $L(P_{ij})$ denote the recorded experience for path P_{ij}, capturing differences between the predicted probabilities and the actual results. Enhancing robotic navigation through a combination of straight-line optimization, probabilistic decision-making, and iterative learning significantly increases both efficiency and adaptability. This approach allows robots to continuously refine their routes in dynamic environments. When jump search proves insufficient in complex settings, a revised nature-inspired algorithm is utilized to effectively navigate these challenges.

Algorithm 1: HAT based robotic navigation optimization

$\Gamma \leftarrow$ all nodes and edges # `store potential nodes and edges`
$\mathcal{N} \leftarrow$ Number of nodes and edges $\in \Gamma$
$\zeta \leftarrow$ initialize as empty # `store evaluated nodes edges`
for $i = 1{:}\mathcal{N}$ **do**
  # `Evaluate desirability and feasibility` $p(n_i) = w_d \times p_d(n_i) + w_o \times p_o(n_i)$
  $\zeta \leftarrow n_i$ # `store selected nodes score`
end
$SN_i \leftarrow$ operator's choice # `store selected nodes`
if *Straight-line applicable* **then**
  Jump search # `straight-line optimization`
else
  rGWO # `path planning`
end

Table 2
The Permission Token List (PTL) from Fig. 6. This table summarizes the scenarios used in Fig. 6 to illustrate how nodes are selected by the initial path construction and human intervention within the hybrid graph. "N" denotes nodes selected for the initial path, and "E" represents nodes chosen based on human factors. The "Action" column indicates whether the iNSA algorithm performed a "Re-Route Locally" to adjust the path or took "No Action." "Re-Route Locally" means that the algorithm creates a new connection or path segment to optimize the robot's route in response to the changes introduced by human selections or environmental factors.

Scenario	Initial path	Human factor	Action
S1	N	E	Re-Route Locally
S2	E	N	Re-Route Locally
S3	N	N	No Action
S4	N	N	No Action
S5	E	E	Re-Route Locally
S6	E	E	Re-Route Locally
S7	N	N	No Action
S8	E	E	Re-Route Locally
S9	E	E	Re-Route Locally
S10	N	N	No Action
S11	N	E	Re-Route Locally

4.3. Node path planning via an rGWO algorithm

Advanced robotic navigation benefits immensely from the synergy between nature-inspired optimization algorithms. The rGWO algorithm demonstrates significant efficacy in exploring complex solution spaces, as seen in Liu et al. [49] proposed framework. However, its tendency towards premature convergence, primarily due to an over-reliance on the alpha (α) wolf's decisions, necessitates a strategic intervention to enhance its exploration capabilities. To address this limitation, Liu et al. integrate the lion optimizer with the traditional GWO algorithm. This hybridization introduces a disturbance factor, enriching the algorithm's ability to diversify its search and avoid local optima, as shown in Algorithm 2 and the equations below:

$$X_{11} = (X_\alpha - A1 \times D_\alpha) \times \alpha_f \tag{11}$$

$$X_{22} = (X_\beta - A2 \times D_\beta) \times \beta_1 \tag{12}$$

$$X_{33} = (X_\delta - A3 \times D_\delta) \times \alpha_c \tag{13}$$

The integration of the lion optimizer not only mitigates the rGWO's premature convergence issue but also enhances its ability to navigate through the multi-dimensional search space more effectively. The lion optimizer, inspired by the territorial defense mechanism and patrolling behavior of lions, introduces a stochastic perturbation into the wolf pack's leadership dynamics. This stochastic component is critical for ensuring that the search process does not stagnate and is represented in the enhanced exploration equation

$$X_{new} = X_{old} + \Delta X_{lion} \times (L_{best} - X_{current}) \tag{14}$$

where ΔX_{lion} signifies the perturbation vector generated by the lion optimizer, L_{best} denotes the best position found by the lion optimizer, and $X_{current}$ represents the current position of the wolf. This equation ensures that the search agents are not only guided by the pack leaders but also incorporate random explorative steps influenced by the lion optimizer's strategy. This nuanced approach allows the rGWO algorithm to dynamically balance exploration and exploitation throughout the optimization process, significantly enhancing its effectiveness in complex environments. The algorithm's adaptability is further underscored by its application in multi-objective scenarios where competing objectives, such as safety, time efficiency, and energy conservation, necessitate a delicate balance between various optimization

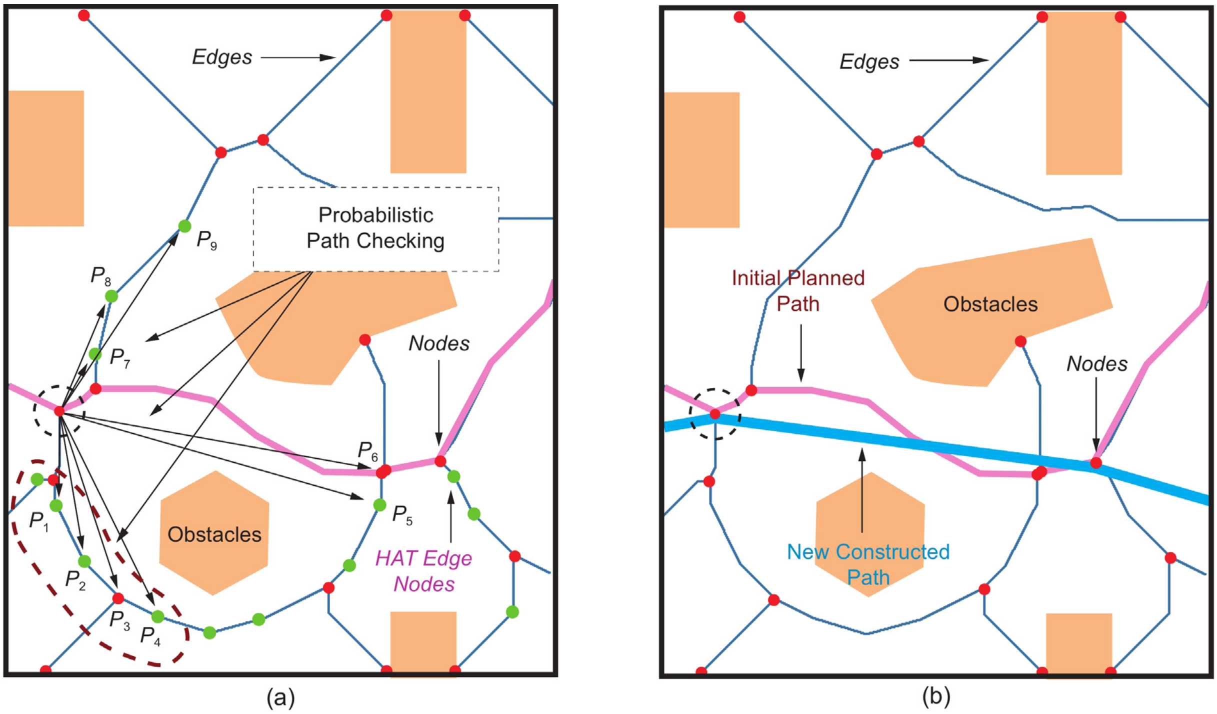

Fig. 8. Illustration of the augmented straight-line optimization approach. (a) depicts the initial path planning generated path through the GVD graph. (b) shows how the augmented straight-line optimization approach optimizes the initial generated path.

Algorithm 2: rGWO via an lion optimizer

Initialize: Wolf pack (α, β, δ), lion pride, position vectors, weights w_l, w_g
while *Not end condition* **do**
  Calculate X_{11}, X_{22}, X_{33}
  `# with lion's disturbance`
  $X_{11} = (X_\alpha - A1 \times D_\alpha) \times \alpha_f$
  $X_{22} = (X_\beta - A2 \times D_\beta) \times \beta_1$
  $X_{33} = (X_\delta - A3 \times D_\delta) \times \alpha_c$
  `# Evaluate solutions via Pareto criteria`
  $X_h \leftarrow w_l \times X_{lion} + w_g \times X_{rGWO}$
  Adjust w_l, w_g for balance
  `# Update wolf positions and hierarchy`
  if *Solution meets all criteria* **then**
    break
  end
end
return Optimal path

---

criteria. To quantitatively assess solution quality in such multi-objective contexts, the algorithm employs a Pareto dominance criterion, enabling it to identify a set of optimal solutions (Pareto front) that represent the best trade-offs among the objectives

$$\forall i \in \{1, \dots, n\} : f_i(s_1) \leq f_i(s_2) \quad \text{and at least for one } j : f_j(s_1) < f_j(s_2) \tag{15}$$

By harmonizing the strengths of both the lion optimizer and GWO, we forge a hybrid rGWO approach that ensures a balanced and effective search mechanism:

$$X_h = w_l \times X_{lion} + w_g \times X_{rGWO} \tag{16}$$

Dynamically adjusting the weights w_l and w_g facilitates a seamless transition between broad, global exploration and focused, local exploitation. This flexibility is paramount for navigating the multi-faceted challenges of robotic navigation, where the algorithm must efficiently discern near-optimal paths amidst a plethora of viable routes, all while minimizing computational overhead. The integration of nature-inspired algorithms, particularly the rGWO augmented with the lion optimizer, represents a significant stride towards achieving this goal, offering a robust, adaptive, and highly effective solution for advanced robotic navigation tasks. The rGWO algorithm is utilized to efficiently determine paths from waypoints to access nodes, thereby establishing a cohesive route through the hybrid graph structure. This algorithm plays a crucial role in optimizing the pathfinding process by leveraging its capability to handle complex graph topologies and dynamically adjust to varying conditions. Beyond its primary function, the rGWO algorithm is also employed to identify optimal paths from nodes to edge nodes. This additional application is pivotal in minimizing the overall path lengths within the graph, enhancing the efficiency and performance of the network. By integrating user instructions, the rGWO algorithm can further refine the pathfinding process, ensuring that the resultant paths are not only shorter but also aligned with specific user-defined criteria. Consequently, the use of the rGWO algorithm in hybrid graph structures significantly improves both the accuracy and efficiency of pathfinding, making it an invaluable tool in the analysis and optimization of complex systems.

4.4. Human factor interaction

Human interaction is a vital component in enhancing the path planning capabilities of the proposed model. Integrating human input into the model serves multiple purposes, foremost among them being the improvement of safety and usability in areas with high concentrations of people, such as public spaces or busy environments. The complexity and unpredictability of such areas often present challenges that purely algorithmic approaches may struggle to address adequately. By involving human judgment, the model can better account for nuanced factors, such as temporary obstacles, crowd movement patterns, or specific

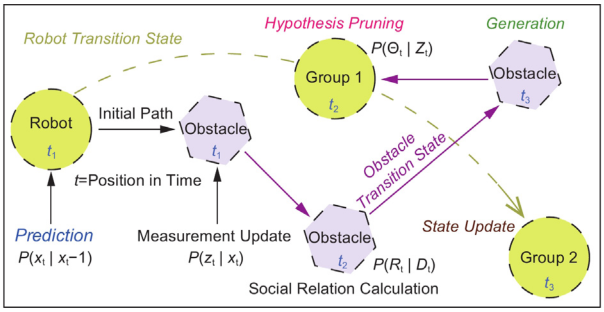

Fig. 9. Illustration of the Multi-Hypothesis Social Grouping and Tracking (MHSGT) model and how its predicts the obstacle movements within the environment.

safety concerns, that automated systems might overlook. Additionally, human interaction enriches the model by introducing a level of creativity and adaptability that extends beyond the limitations of pre-programmed algorithms. Humans can perceive and interpret context in ways that machines cannot, enabling the identification of new and innovative paths that the model might not have considered. This is particularly evident when comparing the initial paths generated by the model to those refined through human intervention. As demonstrated in Fig. 5, the candidate paths developed after incorporating human input were not only significantly improved but also diverged in approach, often taking entirely different routes to reach the goal more effectively. The human factor thus serves as a critical enhancement to the model, allowing it to achieve higher efficiency and greater adaptability. It ensures that the model is not just reacting to the current environment but is also capable of anticipating and responding to potential future scenarios. This proactive capability is essential for creating robust and reliable path planning solutions that are effective in a wide range of real-world situations. In sum, the inclusion of human interaction in the path planning process significantly elevates the model's performance, ensuring that it can deliver safer, more innovative, and more efficient routes in complex environments.

5. Predictive obstacle tracking

Obstacle tracking is crucial in a human–robot intertwined environment as it enables the robot to navigate safely and efficiently while interacting with humans. In dynamic settings where human movement is unpredictable, obstacle tracking ensures that the robot can detect and respond to changes in real-time, avoiding collisions and maintaining smooth operations. This capability enhances the robot's ability to understand and anticipate human actions, leading to more natural and intuitive interactions. Furthermore, effective obstacle tracking contributes to the overall safety and reliability of the system, as it allows the robot to adapt to varying conditions and unexpected obstacles, thereby fostering a more seamless and collaborative human–robot partnership. Incorporating the Multi-Hypothesis Social Grouping and Tracking (MHSGT) model developed by Luber and Arras [50] Arras significantly enhances the understanding of dynamic human environments in the context of human–robot teaming. This model, which focuses on detecting and tracking both individuals and their social groupings in real-time, provides critical insights into human behavior and interactions within a given scenario. This capability is particularly vital in human–robot collaboration settings, where anticipating human movement and interactions can lead to more informed and effective decision-making by robotic systems. The MHSGT model leverages a multi-hypothesis tracking (MHT) approach, allowing the system to hypothesize multiple possible scenarios regarding group formations and individual movements, as seen in Fig. 9. This recursive, real-time model utilizes 2D range data to continuously refine these hypotheses based on new information, making it robust in environments where occlusions and sensor mobility are frequent challenges. To predict human motion more accurately, the model integrates the probability of social relations and group formations into its tracking framework. At a given time t, it calculates the conditional probability of a social relation $P(R_t|D_t)$ based on observed feature sequences D_t, using a Bayesian filtering approach:

$$P(R_t|D_t) = \alpha P(D_t|R_t)P(R_t|R_{t-1})P(R_{t-1}|D_{t-1}) \tag{17}$$

where α is a normalizing constant, $P(D_t|R_t)$ is the likelihood of observing the features given the social relation, and $P(R_t|R_{t-1})$ encodes the transition probabilities of social relations. The likelihood $P(D_t|R_t)$ can be decomposed into

$$P(D_t|R_t) = \prod_{i=1}^{n} P(d_{i,t}|R_t) \tag{18}$$

where $d_{i,t}$ represents individual observations at time t. The transition probability $P(R_t|R_{t-1})$ can be modeled using a Markov process

$$P(R_t|R_{t-1}) = \int P(R_t|R_{t-1}, u_t)P(u_t|R_{t-1})\,du_t \tag{19}$$

where u_t represents the control inputs or actions taken between time $t-1$ and t. The model also tracks multiple hypotheses by introducing an intermediate tree level at each time step, allowing models to branch off from parent hypotheses and form their own data association trees. The final expression for the probability of a hypothesis Θ_t at time t, considering both models and data associations, is given by

$$\begin{aligned} P(\Theta_t|Z_t) = {} & \beta P(Z_t|\phi_t, M_t, \Theta_{t-1})P(\phi_t|M_t, \Theta_{t-1}, Z_{t-1}) \\ & \times P(M_t|\Theta_{t-1})P(\Theta_{t-1}|Z_{t-1}) \end{aligned} \tag{20}$$

Here, Z_t represents the set of measurements, ϕ_t the assignment of measurements to tracks, and M_t the model at time t. The likelihood term $P(Z_t|\phi_t, M_t, \Theta_{t-1})$ can be expressed as

$$P(Z_t|\phi_t, M_t, \Theta_{t-1}) = \prod_{j=1}^{m} P(z_{j,t}|\phi_{j,t}, M_t, \Theta_{t-1}) \tag{21}$$

where $z_{j,t}$ represents individual measurements. To ensure the most probable hypotheses are retained while less likely ones are pruned, the model employs a recursive update mechanism. The probability update for a social relation R_t is given by

$$P(R_t|D_{0:t}) = \frac{P(D_t|R_t)P(R_t|D_{0:t-1})}{P(D_t|D_{0:t-1})} \tag{22}$$

where $D_{0:t}$ represents all observations up to time t. The denominator $P(D_t|D_{0:t-1})$ serves as a normalizing constant ensuring the probabilities sum to one. By integrating the MHSGT model, human–robot teaming is significantly enhanced. The model's ability to track and predict human and group behaviors in real-time aligns perfectly with the goal of leveraging human factors to inform and optimize robotic decision-making processes in dynamic environments. This approach allows robots to better navigate, interact, and collaborate with humans by understanding and anticipating their movements and social dynamics. The MHSGT model represents a comprehensive approach to enhancing human–robot teaming. Its real-time capabilities and robust handling of dynamic environments make it a valuable tool for improving the efficiency and effectiveness of human–robot interactions.

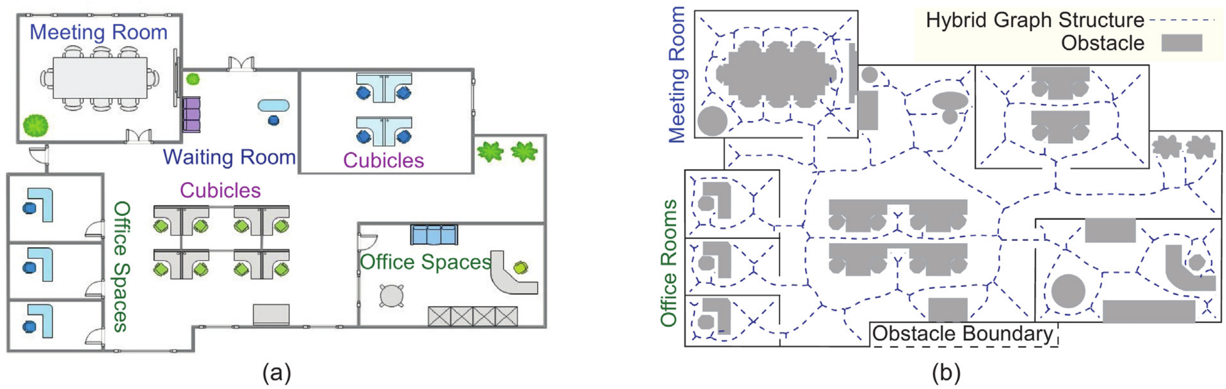

Fig. 10. Illustration of the obstacle environment and hybrid graph structure. (a) showcases the indoor environment with detailed obstacles and boundaries [51]. (b) depicts the hybrid graph structure presented in the proposed framework.

6. Simulation and comparison studies

In this section, how the proposed framework behaves in various indoor situations is addressed. This analysis provides valuable insights into the framework's reliability, effectiveness, and suitability in real-world applications.

6.1. Indoor robot path planning for real-world simulations

The real world simulation, depicted in Fig. 10, rigorously tested the effectiveness of the HAT framework and our enhanced Node Selection Algorithm, using a reconstructed indoor environment to represent typical workplace challenges. This setting included various obstacles—both static and dynamic—as seen in the transition map. Through a series of tasks, from patrols to emergency drills, the robot demonstrated its ability to navigate using the grey-scale obstacle map for distinguishing navigable spaces. Key findings from the simulation included the robot's efficient use of human-supplied waypoints and its adaptability to real-time environmental changes, confirming the practicality of our framework for safe and flexible autonomous navigation in dynamic settings.

Fig. 10(b) provides a detailed layout of an indoor environment designed for the validation of our autonomous navigation system. Fig. 10(b) vividly illustrates the range of obstacles within the environment, each surrounded by a boundary, and presented in grey-scale to distinguish between areas the robot can and cannot navigate. These boundaries are integral to the functionality of our Node Selection Algorithm, which is enhanced by the revised Grey Wolf Optimization, ensuring the robot maintains a safe clearance during operation. The dashed lines suggest feasible trajectories the robot can undertake, adaptable to the ever-changing indoor dynamics, such as moving obstacles or shifts in layout. This ability to dynamically adjust is critical to assessing the robot's real-world performance, reflecting the system's navigational precision, adaptability, and the effectiveness of human intervention when required. These diagrams encapsulate the complexity and effectiveness of our Human-Autonomy Teaming framework, offering a condensed yet rich analysis of our system's capabilities in managing the nuances of safe and efficient autonomous indoor navigation.

In our detailed simulation, presented in Fig. 11, the autonomous navigation system showcased exceptional proficiency in environment mapping and obstacle identification, which was crucial for real-time navigation in an indoor setting. The system's ability to maintain a safe operational distance from obstacles, as evident from the obstacle boundaries, highlights its sophisticated spatial awareness. Furthermore, the dynamic adaptability of the path planning, depicted through the dashed lines indicating multiple potential routes, demonstrates the system's responsiveness to environmental changes, which is essential for real-world application. Fig. 11 specifically underscores the successful navigation to the target point, marking a significant achievement of the simulation. This showcases not only the effectiveness of the iNSA and rGWO but also the seamless integration of human feedback into the autonomous system's decision-making process. Human operators were key in providing initial data and intervening when the robot faced unexpected environmental changes, underscoring the symbiotic nature of the HAT framework. Overall, the simulation results affirm the capability of our hybrid graph system to provide a safe, efficient, and adaptable solution for autonomous robots operating in dynamic indoor spaces. It suggests promising directions for future deployments of autonomous systems in similar environments, with implications for enhancing operational safety and interaction in Human-Autonomy Teaming scenarios.

7. Evaluating multi-waypoint navigation with real-time obstacle tracking

Multi-waypoint navigation is on of the fundamental areas in robotic navigation that allows for autonomous system to achieve multiple task and action to works synchronously within human robot environments. In this simulation study the task of multi-waypoint navigation is given to the proposed model to evaluate and assess its ability to reach each targeted waypoint. From Fig. 12 the result of the multi-waypoint navigation utilizing the proposed strategy can be found. Starting from the initial position, marked by a green hexagon, the model initiates the path planning process. The environment contains several waypoints, also denoted by green hexagons in the figure. The HAT-based optimized path is depicted by the orange dashed line. This path is the result of the optimization process applied by the HAT model, demonstrating the model's proficiency in handling multi-waypoint navigation. The optimized path is designed to navigate efficiently around obstacles, which are scattered throughout the environment. These obstacles and their boundaries pose significant challenges that the path planning algorithm must overcome to ensure a collision-free route. A critical aspect of the path planning process is the waypoint visiting sequence, which is determined by human factors. This sequence is illustrated by the connectivity between waypoints, showing the order in which the autonomous system visits each target. The human-influenced sequencing is essential for demonstrating the model's flexibility and responsiveness to human input. Within the proposed model its is assumed that the given waypoint sequence is based on importance by the user and not around optimality.

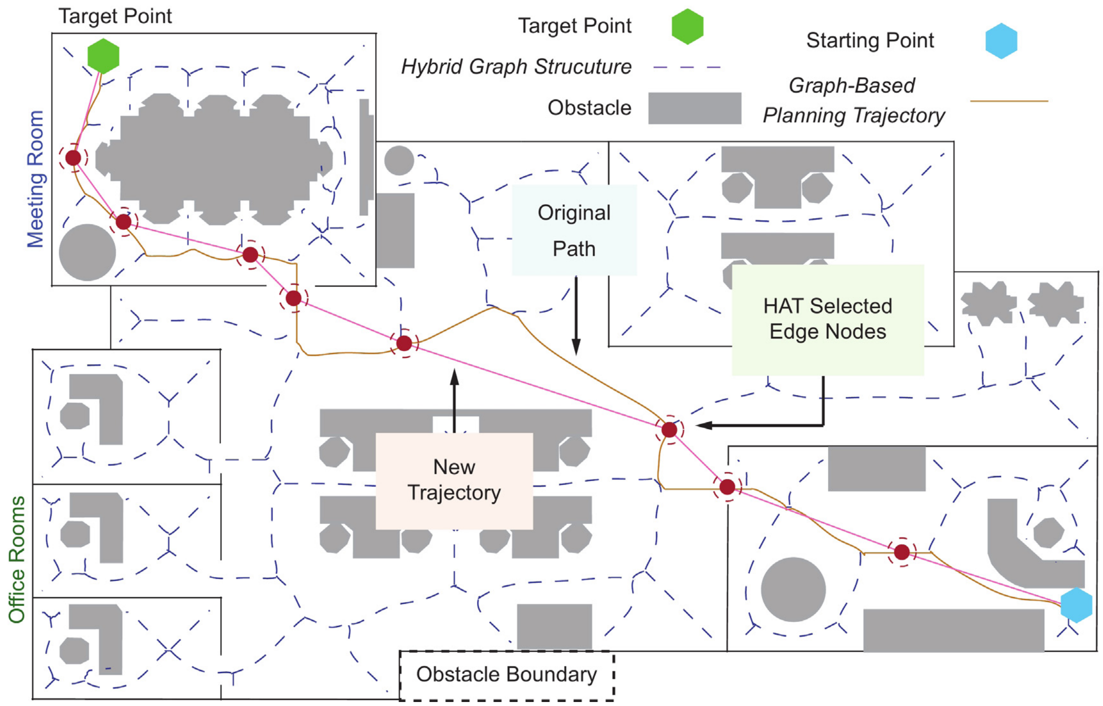

Fig. 11. Illustration of the real-world path planing result of the proposed framework. The figure depicts the initial planned trajectory of the proposed hybrid graph structure, as well as illustrates the improved path from the initial trajectory utilizing HAT based methodology.

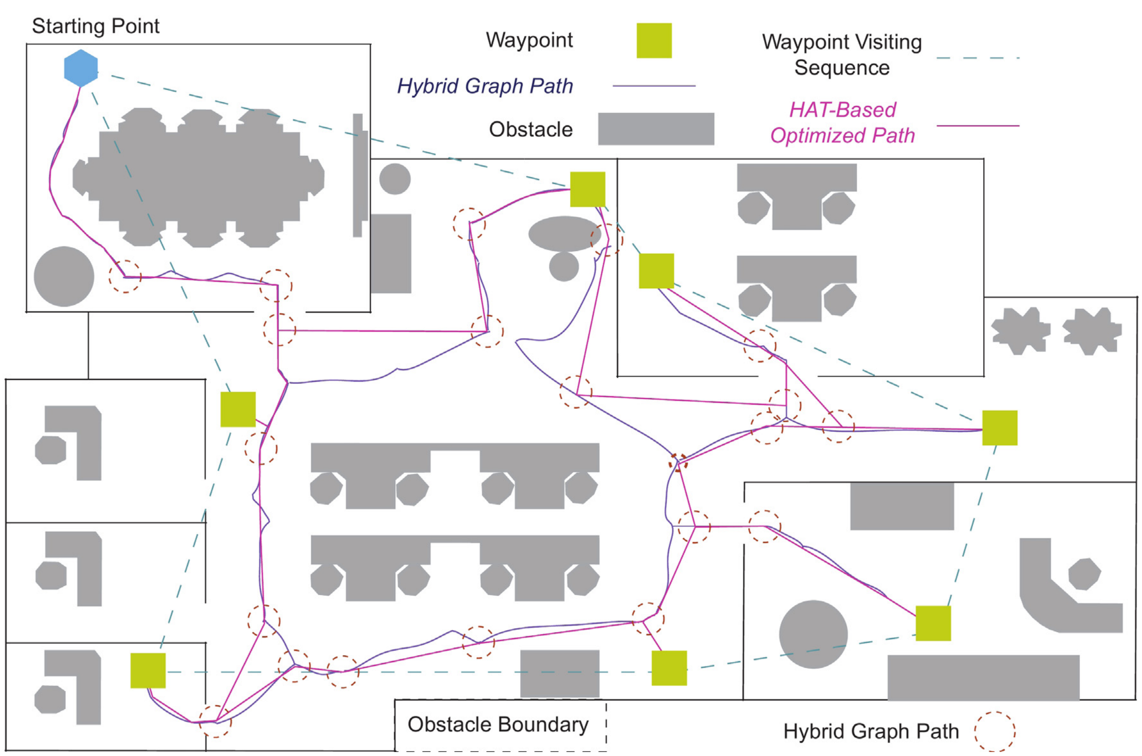

Fig. 12. The figure presents the results of our proposed HAT model for autonomous path planning, emphasizing its capability in multi-waypoint navigation. The proposed model integrates human factors to determine the waypoint sequencing order, and it utilizes a hybrid-graph structure combined with an iNSA to optimize the path planning process.

Initially, the model generates a preliminary path using a hybrid-graph structure, shown by the purple solid line. This initial path serves as a baseline route before any optimization is applied. The hybrid graph structure leverages both nodes and edges to construct a navigable path through the environment. Subsequently, the iNSA model refines this path by allowing the system to make strategic jumps between nodes or edge nodes. These crucial points within the hybrid graph, where transitions occur, are denoted by small red dashed circles. The iNSA model's optimization process is instrumental in enhancing the overall

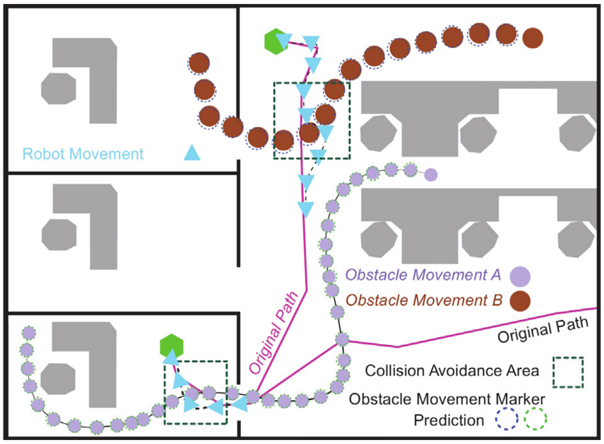

Fig. 13. The figure illustrates the capability of our proposed model in handling dynamic obstacle tracking and collision avoidance within an autonomous path planning framework. The model's ability to predict and respond to the movement of obstacles in a dynamic environment is demonstrated, ensuring the robot can still adhere to its pre-generated paths.

route, ensuring that the system can efficiently navigate from one waypoint to the next while avoiding obstacles. The comparison between the initial hybrid graph path and the HAT-based optimized path underscores the improvements achieved through our approach. The optimized path demonstrates superior efficiency and obstacle avoidance, highlighting the model's capability to handle complex multi-waypoint navigation tasks effectively. This capability is particularly significant in dynamic environments where obstacles and waypoints may change, requiring the autonomous system to adapt swiftly. In summary, the figure showcases the efficacy of the HAT-based path planning model in multi-waypoint navigation. By integrating human-determined waypoint sequencing and leveraging a hybrid-graph structure refined by the iNSA model, the system can navigate complex environments efficiently and safely. The demonstrated improvements in the optimized path compared to the initial hybrid graph path illustrate the model's potential for practical applications in various autonomous navigation scenarios.

To illustrate the effectiveness of the obstacle tracking methodology, dynamic obstacles are employed in the indoor environment from Fig. 10 The result can be found in Fig. 13, here two dynamic obstacles are employed to show the effectiveness of the utilized model. Fig. 13 begins by illustrating the robot's movement along its original planned path (solid orange line). As the robot progresses, it encounters dynamic obstacles whose movements are denoted as obstacle movement A and obstacle movement B. These obstacles move within predefined trajectories, marked by red triangles. Obstacle Movement A follows a horizontal path, while Obstacle Movement B follows a vertical path. To ensure safe navigation, the model integrates a predictive mechanism to forecast the future positions of these moving obstacles. This prediction is visualized through the obstacle movement marker prediction, represented by light blue and pink circles. By anticipating the obstacles' future positions, the model can dynamically adjust the robot's path in real time. As the robot encounters these dynamic obstacles, it must deviate from its original path to avoid potential collisions. The areas where these adjustments occur are marked as collision avoidance areas (green dashed squares). Within these zones, the model recalculates the robot's trajectory to ensure it can navigate safely around the moving obstacles. The recalculation of the plan trajectory is done through the iGWO algorithm within the node selection algorithm. It is able to take in the scenario and avoid a collision with the obstacle, due to information given by the obstacle tracking mechanism. The adjusted robot movement showing the new path the robot follows to avoid collisions while still aiming to adhere as closely as possible to the original path. This demonstrates the model's robustness in dynamic environments, where obstacles are not static and can move unpredictably. Overall, this figure highlights the model's advanced capabilities in dynamic obstacle tracking and collision avoidance. By predicting obstacle movements and adjusting the robot's path accordingly, the model ensures safe and efficient navigation even in complex and unpredictable environments. This ability to adapt in real-time to dynamic changes is crucial for autonomous systems operating in real-world scenarios where static obstacle assumptions are often unrealistic.

Table 3
Comparative path lengths of the compared models.

Model	Path length (m)
D*-Lite	30.59
IAACO	28.52
Smart-RRT*	29.38
Dijkstra's	33.33
Proposed model	**28.06**

7.1. Comparative analysis of path planning methods

In our comprehensive path planning comparison analyses, we scrutinized the performance of our innovative HAT-based model alongside four established path planning algorithms: D*-Lite, IAACO, Smart-RRT*, and Dijkstra's. Distinct from these established methods, our model harnesses human insights to significantly enhance decision-making processes. This paradigm shift enables our system to adapt rapidly to complex environments, transcending the limitations of traditional pathfinding methods which are bound by predetermined rule sets and are less flexible. The inclusion of human inputs allows our iNSA to recalibrate paths in real-time, making it an intelligent counterpart to conventional algorithms that cannot swiftly accommodate unforeseen changes in the environment. The capability to not just find shorter paths, but also to navigate more efficiently and safely through unpredictable indoor scenarios, is a notable advantage of our HAT approach. The path lengths documented in Table 3 and visually captured in Fig. 14 affirm the iNSA's proficiency. Fig. 14(a), depicts the comparative path traversal by each algorithm, highlighting the IAACO algorithm as suggested by Miao et al. [52], and Fig. 14(b) delineates the optimized route formulated by our proposed model, marking clear distinctions in the efficacy of the varied algorithms. The results corroborate the superior performance of our model in terms of navigation efficiency, underscoring the valuable integration of human cognitive capabilities with algorithmic efficiency. The synergy created through this integration not only fosters a more intuitive navigation system but also signifies a substantial leap forward for autonomous robotics. It suggests vast potential for deploying such enhanced models in more intricate and dynamic scenarios, possibly revolutionizing the landscape of autonomous system applications.

7.2. Comparative analysis of HAT-based path planning method

This segment of our research presents an in-depth comparative analysis of the novel HAT-based path planning model against the established RRT-PCHIP algorithm. The proposed model leverages the iNSA and HAT input for initial path planning, introducing a human-interactive dimension to autonomous navigation. Our analysis was grounded on four pivotal metrics critical to the efficacy of indoor navigation algorithms: the succinctness of the path (path length), the algorithm's applicability to real-world

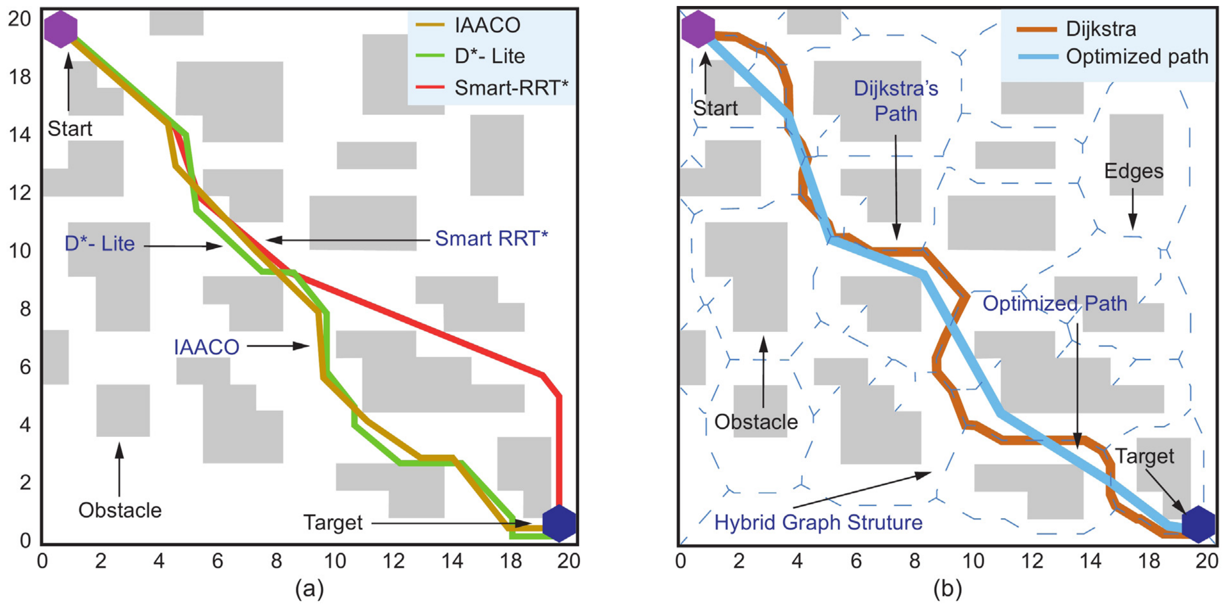

Fig. 14. Illustration of the path planning comparison result conducted with the proposed model. (a) illustrates the compared algorithms including the [52] proposed IAACO algorithm. (b) depicts the proposed methodology generated path.

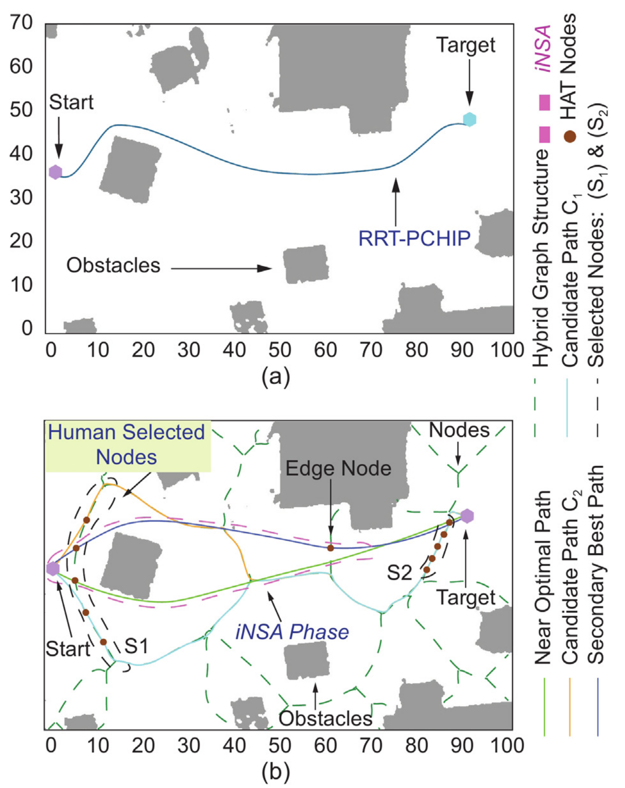

Fig. 15. This figure illustrates the comparative analysis between Rapidly-exploring Random Trees (RRT) with Piecewise Cubic Hermite Interpolating Polynomial (RRT-PCHIP) [53] and the proposed framework. (a) depicts the compared model result for indoor path planning. (b) shows the result gained from the proposed methodology.

scenarios (real-world feasibility), its flexibility in adapting to unpredictable changes within the environment (adaptability), and its computational speed and efficiency (computational efficiency). These criteria are indispensable in discerning the performance and practicality of path planning models in real-time indoor navigation. As detailed in Fig. 15 and Table 4, the proposed model exhibits a marked improvement in path efficiency. Within Table 4, the "✗" symbol indicates nodes dismissed by the human operator, representing an interactive filtration process to refine the proposed path. Conversely, the "✓" signifies nodes selected for forming the path, indicative of human-approved decisions in the path planning process. The initial paths, represented by lengths of 125.51 m and 124.15 m, were significantly optimized to 94.57 m and 93.98 m through strategic human-aided node selection and algorithmic calculation, illustrating the profound impact of the iNSA and HAT framework. In comparison, the conventional RRT-PCHIP algorithm yielded a longer path of 100.44 m, highlighting the enhanced spatial understanding and navigation acumen introduced by HAT principles. The evidence gleaned from our rigorous analysis unambiguously supports the superiority of the proposed HAT-based model. It showcases an exceptional blend of human-intuitive decision-making with algorithmic efficiency, which is manifest in its application to complex, real-world indoor environments. The proposed model outstrips conventional path planning algorithms like Dijkstra and demonstrates robustness against more recent models like RRT-PCHIP, establishing a new benchmark for computational and practical efficiency in the realm of autonomous indoor navigation systems.

8. Conclusion

We have introduced a framework that significantly enhances robotic cooperation in large indoor spaces by utilizing HAT principles alongside an obstacle tracking methodology. This methodology improves the model's understanding of human movement and dynamic obstacles within the environment. By coordinating the proposed graph structure with the iNSA algorithm, our framework ensures adaptability and robustness in HAT-based safety-aware navigation. A key aspect of our approach is the integration of human input, which includes providing environmental data, selecting crucial waypoints, and addressing unexpected challenges. This human–robot collaboration is designed to optimize performance in dynamic settings, making the system more resilient and responsive to environmental changes. Extensive testing has shown that this collaboration enhances the model's effectiveness, allowing it to navigate complex and evolving indoor spaces with greater efficiency.

Table 4
The table depicts the nodes and edge nodes utilized in the HAT node selection process.

Scenario	PTL	1	2	3	4	5	6	7
S1	Selection	✗	✓	✓	✗	✗	✗	✗
S2	Selection	✓	✗	✗	✗	✗	–	–

Additionally, simulation and comparative studies have validated the efficiency and reliability of our model. These studies demonstrate that our framework not only meets but often exceeds performance benchmarks in various dynamic scenarios. The enhanced cooperation facilitated by HAT principles and obstacle tracking methodology ensures that the model can handle current challenges and is adaptable to future advancements in human–robot interaction and dynamic environment navigation. In conclusion, our framework offers a robust solution for improving robotic cooperation in large indoor spaces. It achieves this by integrating advanced algorithms and human input, resulting in a highly adaptable, efficient, and reliable model capable of navigating complex and unpredictable environments.

For future work, we will focus on several key aspects to further enhance and validate our framework. First, a user study or feedback session will be conducted to evaluate the ease of interaction between humans and the robot, which is crucial for the HAT approach. Additionally, we plan to explore the scalability of the proposed model in larger and more complex environments, as well as its adaptability in rapidly changing scenarios. A more detailed examination of the study's limitations, including potential biases and technological constraints, will be undertaken to provide a balanced and comprehensive perspective. These efforts will help refine our framework, ensuring it remains robust and effective in diverse real-world applications.

CRediT authorship contribution statement

Timothy Sellers: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology. **Tingjun Lei:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Chaomin Luo:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Zhuming Bi:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Gene Eu Jan:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors 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 research was partially supported by the Mississippi Space Grant Consortium under NASA EPSCoR RID grant.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.birob.2024.100189.

References

[1] T. Lei, T. Sellers, C. Luo, D.W. Carruth, Z. Bi, Graph-based robot optimal path planning with bio-inspired algorithms, Biomimetic Intell. Robot. 3 (3) (2023) 100119.
[2] G.E. Jan, T. Lei, C.C. Sun, Z.Y. You, C. Luo, On the problems of drone formation and light shows, IEEE Trans. Consum. Electron. (2024).
[3] D. Short, T. Lei, C. Luo, D.W. Carruth, Z. Bi, A bio-inspired algorithm in image-based path planning and localization using visual features and maps, Intell. Robot. 3 (2) (2023) http://dx.doi.org/10.20517/ir.2023.14.
[4] T. Sellers, T. Lei, D. Carruth, C. Luo, Deep learning-based heterogeneous system for autonomous navigation, in: Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping VIII, vol. 12539, SPIE, 2023, pp. 140–153.
[5] T. Sellers, T. Lei, G.E. Jan, Y. Wang, C. Luo, Multi-objective optimization robot navigation through a graph-driven PSO mechanism, in: International Conference on Sensing and Imaging, Springer, 2022, pp. 66–77.
[6] J.H. Rogers, T. Sellers, T. Lei, D.W. Carruth, C. Luo, Sensor-based multi-waypoint autonomous robot navigation with graph-based models, in: Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea, and Space Vehicles and Infrastructure 2023, vol. 12540, SPIE, 2023, pp. 215–224.
[7] D. Zhu, C. Tian, X. Jiang, C. Luo, Multi-AUVs cooperative complete coverage path planning based on GBNN algorithm, in: 2017 29th Chinese Control and Decision Conference, CCDC, IEEE, 2017, pp. 6761–6766.
[8] S. Alarabi, C. Luo, M. Santora, A PRM approach to path planning with obstacle avoidance of an autonomous robot, in: 2022 8th International Conference on Automation, Robotics and Applications, ICARA, IEEE, 2022, pp. 76–80.
[9] B. Black, T. Sellers, T. Lei, C. Luo, D.W. Carruth, Optimal multi-target navigation via graph-based algorithms in complex environments, in: 2024 IEEE 33rd International Symposium on Industrial Electronics, ISIE, IEEE, 2024, pp. 1–6.
[10] T. Sellers, T. Lei, C. Luo, G.E. Jan, J. Ma, A node selection algorithm to graph-based multi-waypoint optimization navigation and mapping, Intell. Robot. 2 (4) (2022) 333–354.
[11] T. Lei, C. Luo, T. Sellers, S. Rahimi, A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping, Intell. Syst. Appl. 12 (2021) 200053.
[12] P. Chintam, T. Lei, B. Osmanoglu, Y. Wang, C. Luo, Informed sampling space driven robot informative path planning, Robot. Auton. Syst. 175 (2024) 104656.
[13] E. Riser, T. Sellers, T. Lei, C. Pradeep, C. Luo, Multi-robot navigation using improved RRT*-smart with digital twin technology, in: Autonomous Systems: Sensors, Processing, and Security for Ground, Air, Sea, and Space Vehicles and Infrastructure 2024, vol. 13052, SPIE, 2024, pp. 114–125.
[14] T. Lei, C. Luo, S.X. Yang, D.W. Carruth, Z. Bi, Bio-inspired intelligence-based multi-agent navigation with safety-aware considerations, IEEE Trans. Artif. Intell. (2023).
[15] A. Srivastava, P. Singh, S.K. Verma, K. Kartikey, P.S. Shukla, Path planning robot using PI-CAM, Int. J. Res. Appl. Sci. Eng. Technol. (2022) http://dx.doi.org/10.22214/ijraset.2022.41133.
[16] J. Han, Y. Seo, Mobile robot path planning with surrounding point set and path improvement, Appl. Soft Comput. 57 (2017) 35–47, http://dx.doi.org/10.1016/j.asoc.2017.03.035.
[17] A. Gandia, S. Parascho, R. Rust, G. Casas, F. Gramazio, M. Kohler, Towards automatic path planning for robotically assembled spatial structures, in: Robotic Fabrication in Architecture, Art and Design 2018, 2018, http://dx.doi.org/10.1007/978-3-319-92294-2_5.
[18] M. Sidler, C.R. von Rohr, R. Dornberger, T. Hanne, Emotion influenced robotic path planning, in: Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, ISMSI '17, Association for Computing Machinery, 2017, pp. 130–136, http://dx.doi.org/10.1145/3059336.3059346.
[19] S. Nair, P. Supriya, Multi robotic path planning in a static environment, in: 2022 4th International Conference on Inventive Research in Computing Applications, ICIRCA, 2022, pp. 211–217, http://dx.doi.org/10.1109/ICIRCA54612.2022.9985576.
[20] G.R. Theja, S. Naidu, Robot path planning using memory, Lect. Notes Netw. Syst. (2019) http://dx.doi.org/10.1007/978-981-13-3765-9_45.
[21] D. Drake, S. Koziol, E. Chabot, Mobile robot path planning with a moving goal, IEEE Access 6 (2018) 12800–12814, http://dx.doi.org/10.1109/ACCESS.2018.2797070.
[22] T. Dam, G. Chalvatzaki, J. Peters, J. Pajarinen, Monte-Carlo robot path planning, IEEE Robot. Autom. Lett. 7 (2022) 11213–11220, http://dx.doi.org/10.1109/LRA.2022.3199674.
[23] H. Zhang, Y. Wang, J. Zheng, J. Yu, Path planning of industrial robot based on improved RRT algorithm in complex environments, IEEE Access 6 (2018) 53296–53306, http://dx.doi.org/10.1109/ACCESS.2018.2871222.
[24] D. Lyu, Z. Chen, Z. Cai, S. Piao, Robot path planning by leveraging the graph-encoded Floyd algorithm, Future Gener. Comput. Syst. 122 (2021) 204–208, http://dx.doi.org/10.1016/J.FUTURE.2021.03.007.
[25] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, K. Alexis, Graph-based path planning for autonomous robotic exploration in subterranean environments, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2019, pp. 3105–3112, http://dx.doi.org/10.1109/IROS40897.2019.8968151.

[26] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, M. Hutter, Graph-based subterranean exploration path planning using aerial and legged robots, J. Field Robotics 37 (8) (2020) 1363–1388.

[27] D. Belanová, M. Mach, P. Sinčák, K. Yoshida, Path planning on robot based on D* lite algorithm, in: 2018 World Symposium on Digital Intelligence for Systems and Machines, DISA, IEEE, 2018, pp. 125–130.

[28] Y. Liu, Y. Jiang, Robotic path planning based on a triangular mesh map, Int. J. Control Autom. Syst. (2020) 1–9, http://dx.doi.org/10.1007/s12555-019-0396-z.

[29] X.X. Diao, W. Chi, J. Wang, Graph neural network based method for path planning problem, 2023, http://dx.doi.org/10.48550/arXiv.2309.14845, ArXiv abs/2309.14845.

[30] E.N. Sabudin, R. Omar, S.K. Debnath, M.S. Sulong, Efficient robotic path planning algorithm based on artificial potential field, Int. J. Electr. Comput. Eng. 11 (2021) 4840–4849, http://dx.doi.org/10.11591/IJECE.V11I6.PP4840-4849.

[31] Z. Clawson, X. Ding, B. Englot, T.A. Frewen, W.M. Sisson, A. Vladimirsky, A bi-criteria path planning algorithm for robotics applications, 2015, ArXiv abs/1511.01166.

[32] J.S. Lee, D.W. Kim, An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph, Inform. Sci. 332 (2016) 1–18, http://dx.doi.org/10.1016/j.ins.2015.11.004.

[33] M.S. Ganeshmurthy, G. Suresh, Path planning algorithm for autonomous mobile robot in dynamic environment, in: 2015 3rd International Conference on Signal Processing, Communication and Networking, ICSCN, 2015, pp. 1–6, http://dx.doi.org/10.1109/ICSCN.2015.7219901.

[34] W. Zhao, R. Lun, C. Gordon, A.-B.M. Fofana, D.D. Espy, A. Reinthal, B. Ekelman, G.D. Goodman, J.E. Niederriter, C. Luo, et al., Liftingdoneright: A privacy-aware human motion tracking system for healthcare professionals, Int. J. Handheld Comput. Res. 7 (3) (2016) 1–15.

[35] S. Music, S. Hirche, Control sharing in human-robot team interaction, Annu. Rev. Control 44 (2017) 342–354.

[36] Y. Ajmera, Shared autonomy in web-based human robot interaction, 2020, http://dx.doi.org/10.1007/978-3-030-55190-2_55, ArXiv.

[37] G. Tokadlı, M.C. Dorneich, Interaction paradigms: From human-human teaming to human-autonomy teaming, in: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference, DASC, IEEE, 2019, pp. 1–8.

[38] S. Nikolaidis, Y. Zhu, D. Hsu, S. Srinivasa, Human-robot mutual adaptation in shared autonomy, in: 2017 12th ACM/IEEE International Conference on Human-Robot Interaction, HRI, 2017, pp. 294–302, http://dx.doi.org/10.1145/2909824.3020252.

[39] J.M. Beer, A.D. Fisk, W. Rogers, Toward a framework for levels of robot autonomy in human-robot interaction, J. Hum.-Robot Interact. 3 (2014) 74–99, http://dx.doi.org/10.5898/JHRI.3.2.Beer.

[40] R. Scalise, Y. Bisk, M. Forbes, D. Yi, Y. Choi, S. Srinivasa, Balancing shared autonomy with human-robot communication, 2018, ArXiv abs/1805.07719.

[41] R. Pérula-Martínez, Á. Castro-González, M. Malfaz, M.A. Salichs, Autonomy in human-robot interaction scenarios for entertainment, in: Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, 2017, pp. 259–260.

[42] J. Miller, S. Hong, J. Lu, Self-driving mobile robots using human-robot interactions, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2018, pp. 1251–1256, http://dx.doi.org/10.1109/SMC.2018.00219.

[43] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi, B. Siciliano, Autonomy in physical human-robot interaction: a brief survey, IEEE Robot. Autom. Lett. 6 (2021) 7989–7996, http://dx.doi.org/10.1109/LRA.2021.3100603.

[44] F. Cantucci, R. Falcone, C. Castelfranchi, Human-robot interaction through adjustable social autonomy, Intell. Artif. 16 (2022) 69–79, http://dx.doi.org/10.3233/ia-210124.

[45] J. Wang, M.Q.H. Meng, Optimal path planning using generalized voronoi graph and multiple potential functions, IEEE Trans. Ind. Electron. 67 (12) (2020) 10621–10630.

[46] D. Chen, A. Xiao, M. Zou, W. Chi, J. Wang, L. Sun, GVD-exploration: an efficient autonomous robot exploration framework based on fast generalized voronoi diagram extraction, 2023, arXiv preprint arXiv:2309.06041.

[47] M. Rowold, L. Ögretmen, T. Kerbl, B. Lohmann, Efficient spatiotemporal graph search for local trajectory planning on oval race tracks, in: Actuators, vol. 11, (11) MDPI, 2022, p. 319.

[48] A. Meliou, A. Krause, C. Guestrin, J.M. Hellerstein, Nonmyopic informative path planning in spatio-temporal models, in: AAAI, vol. 10, (4) 2007, pp. 16–17.

[49] J. Liu, X. Wei, H. Huang, An improved grey wolf optimization algorithm and its application in path planning, IEEE Access 9 (2021) 121944–121956.

[50] M. Luber, K.O. Arras, Multi-hypothesis social grouping and tracking for mobile robots, in: Robotics: Science and Systems, 2013.

[51] W. EdrawMax, A small home office layout, 2020, [Online]; URL: https://www.edrawmax.com/templates/1001382/. (Accessed 15 April 2024).

[52] C. Miao, G. Chen, C. Yan, Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, Comput. Ind. Eng. 156 (2021) 107230.

[53] A. Hentout, A. Maoudj, D. Guir, S. Saighi, M.A. Harkat, M.Z. Hammouche, A. Bakdi, Collision-free path planning for indoor mobile robots based on rapidly-exploring random trees and piecewise cubic hermite interpolating polynomial, Int. J. Imaging Robot. 19 (03) (2019).