Optimization of Urban Search Trajectories for Unmanned Ground Vehicles (UGVs) through Deep Reinforcement Learning and Graph-Based Techniques

by

Kina Kim

A thesis submitted in conformity with the requirements
for the degree of Bachelor of Applied Science
Graduate Department of Engineering Science
University of Toronto

Abstract

Optimization of Urban Search Trajectories
for Unmanned Ground Vehicles (UGVs) through
Deep Reinforcement Learning and Graph-Based Techniques

Kina Kim
Bachelor of Applied Science
Graduate Department of Engineering Science
University of Toronto
2024

This thesis explores the optimization of search trajectories for Unmanned Ground Vehicles (UGVs) in non-disastrous urban environments, focusing on locating vulnerable individuals such as lost children or dementia patients. Addressing the complexities of urban landscapes and the limitations of existing search methods, this study develops a more efficient and tailored solution by integrating graph theory with deep reinforcement learning (DRL) techniques specifically designed for urban search operations.

The research introduces a single-agent UGV model, employing a reinforcement learning approach to trajectory planning that strategically balances coverage and the likelihood of finding targets within a limited time frame. By transforming urban map data into a graph format, this model leverages the application of advanced graph-based techniques, such as Deep Graph Infomax (DGI) and Graph Neural Networks (GNNs), to refine search routes. Opting for a single-agent allows for in-depth exploration of system development without the complexities of coordinating multiple agents. Yet the framework is consciously designed with scalability in mind, setting the stage for future expansion to real-life applications. The proposed framework demonstrates potential to significantly enhance operational efficiency and effectiveness in intricate urban environments.

While the preliminary results are promising, further validation and testing in more varied and complex urban settings are essential to fully evaluate the effectiveness of the proposed methods. This research contributes to urban search operations by proposing a model that not only integrates advanced computational techniques but also suggests a scalable framework for future enhancements in public safety and search strategies.

Acknowledgements

I would like to express my gratitude to Professor Beno Benhabib for the invaluable research opportunities he has provided. I also send special thanks to Cameron Haigh for his close supervision and supportwhich guided me to the successful completion of this study.

Last but not least, my heartfelt appreciation goes to my family—Jinyoung, Haejung, and Kyeongeun. Their continuous encouragement and support have been my constant source of strength throughout my undergraduate journey.

Contents

List of Figures

1 Introduction

Annually, Canada reports an estimated 70,000 to 80,000 missing persons cases, emphasizing the importance of search operations for public safety and individual recovery [15][31]. Traditional search operations have predominantly relied on manual efforts by police and specialized rescue teams. However, the advent of technological advancements is transforming the search operation paradigm, exemplified by Police Scotland's Air Support Unit, which has incorporated drones into their search missions [1]. This shift highlights the growing necessity for research to facilitate and optimize such technological evolution in search operations.

This thesis investigates Urban Search operations, distinguishing them from the general Search and Rescue (SAR) framework, which includes Wilderness SAR (WiSAR) and Urban SAR (USAR). While WiSAR involves rescuing individuals in natural, often challenging terrains, USAR is concerned with post-disaster urban rescues, such as after earthquakes [30][39]. However, this study focuses on Urban Search in a non-disastrous urban setting, targeting the location of mobile, vulnerable individuals like lost children or dementia patients. This context, although urban, aligns more closely with WiSAR in its non-catastrophic nature and the assumption that the target remains mobile, yet it presents unique challenges due to the dense and complex nature of urban environments.

In robot-aided search operations, the choice between Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs) is pivotal. UGVs are favored in this urban context for their ability to perform proximity searches along potential paths of the missing individual, offering a higher likelihood of direct observation compared to the aerial perspective of UAVs. Additionally, UGVs boast greater endurance, crucial for continuous search efforts without the frequent interruptions for battery replacements typical of UAVs.

The thesis marks a departure from prevailing research trends that emphasize multi-agent UAV strategies for enhancing search efficiency. Instead, it delves in on the trajectory planning of a single UGV allowing for a deeper exploration into optimizing trajectory planning algorithms, avoiding the complexities inherent in coordinating multiple agents. This shift is underpinned by an innovative problem formulation, where the urbans search challenge is reinterpreted through a graph-theoretical lens. This reconceptualization, inspired by road networks that naturally mirror the structure of a graph, offers a logical and efficient model for developing search strategies.

Building upon this foundation, the thesis introduces a cutting-edge algorithm designed to optimize search trajectories by judiciously balancing the coverage and likelihood of locating missing individuals within the constraint of time in an urban context. This approach integrates Reinforcement Learning (RL) and advanced graph-based techniques like Deep Graph Infomax and Graph Neural Networks (GNNS), aiming to refine the optimization of search routes. This strategic emphasis is intended to boost the efficiency and effectiveness of urban search operations.

Ultimately, by adopting a single-agent UGV model coupled with a graph-theoretical strategy for trajectory planning, this thesis endeavors to fill a substantial gap in current urban search methodologies. It presents a novel problem formulation using graph theory and unfolds a groundbreaking, graphically informed trajectory planning approach. This innovative strategy is set to enhance operational performance and success rates in urban search missions, thereby promoting public safety and demonstrating the significant impact of sophisticated computational techniques in the field of search and rescue.

2 Literature Review

Given the scarcity of directly comparable studies in Urban Search, this literature review establishes foundational knowledge by evaluating strategies from other SAR methodologies, particularly WiSAR, albeit in contrasting settings. WiSAR and Urban Search share operational dynamics, especially in non-disaster scenarios with mobile targets, demonstrating significant similarities with Urban Search. The review aims to leverage these parallels to enrich understanding and identify research gaps. Next, it will examine graph theory's applicability to Urban Search, assessing potential algorithm adaptations or limitations. Lastly, the review will discuss Reinforcement Learning (RL) fundamentals and alongside techniques like Deep Graph Infomax and Graph Neural Networks that enhance the proposed search strategies. This comprehensive analysis addresses the research gap and prepares the groundwork for the thesis's proposed innovations.

Section 2.1 will focus on trajectory planning within WiSAR environments, exploring the specific methodologies used. Section 2.2 will examine graph theory applications within the framework, detailing how these theories are utilized. In Section 2.3, the discussion will shift to Reinforcement Learning, the central methodology of this research, and will also cover the integration of Deep Graph Infomax and Graph Neural Networks. This section will highlight how these technologies synergize with Reinforcement Learning to enhance its effectiveness for the research's particular context.

2.1 WiSAR

2.1.1 Navigational Strategies in WiSAR

WiSAR has historically received a good amount of research focus, owing to its varied applications in critical scenarios from locating lost hikers to avalanche rescue [35][32]. This extensive research provides a rich repository of strategies and technological advancements that could be adapted.

WiSAR favors UAV over UGV for its operation in uneven terrain and for its unstructured nature necessitating in a more exploratory approach. This frequently leads to systematic coverage methods. The idea is to cover as much area as possible and while covering the plane will be able to locate the missing person [29].

A common baseline approach is called the parallel swath, also known as a lawnmower strategy [11]. This follows a zig-zag pattern for comprehensive area coverage, as depicted in Figure 1. Again UAVs are typically preferred over UGVs due to their extensive coverage capabilities. While effective for thorough exploration, this method may not be

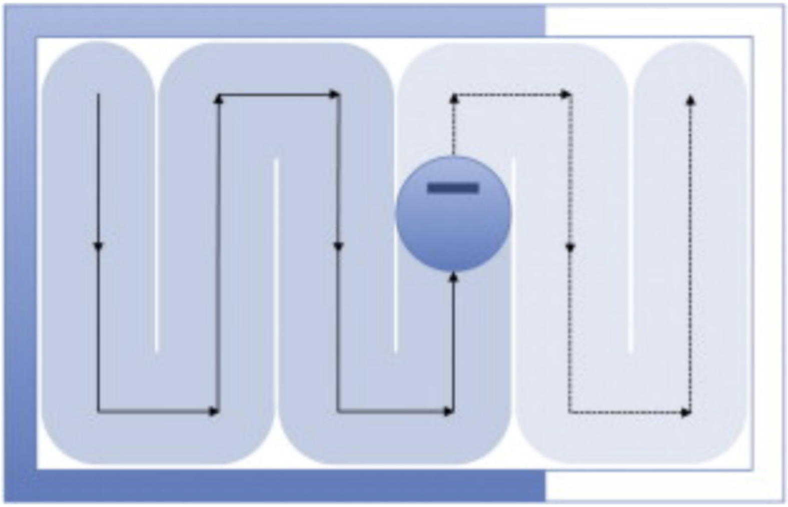


Figure 1: Parallel Swath Illustration [13]

the most time efficient for urgent missing person searches.

Refining this method, WiSAR introduced a more sophisticated strategy called the Modified Lawn-Mower. This strategy weights regions differently based on uncertainty, letting it focus on the areas that require more attention due to higher uncertainty. This strategy then balances coverage and time efficiency [28].

Further algorithmic advancements like LHC_GW_CONV (Local Hill Climb, Global Warming, Convolution) [22] and the Greedy algorithm [36] use grid-based local search methods to navigate [10]. These algorithms focus on adjacent cells with the highest fitness value, employing tie-breaking strategies when necessary. Reference 22 uses convolution kernels to determine the direction forward, while Reference 40 incorporates various strategies including LA_MaxMax (a variant of a greedy l-step look-ahead algorithm) which proved to be the most effective [40]. These methods, despite their efficiency, face the limitation brought by the nature of greedy algorithms of being too focused on local maxima, neglecting broader exploration.

Addressing this, a 'global warming' feature enhances broader exploration by reducing repetitive visits to the same areas [23]. Moreover, a unique heuristic based on the Mode Goodness (MG) ratio, utilizing Gaussian Mixture Models (GMM), is introduced for hierarchical path planning. This allows for prioritization of specific search subregions, guiding UAVs effectively to areas of higher priority.

2.1.2 Determining Coverage Areas

The previously discussed coverage-based algorithms calls for more details on "*how* is the area of interest defined?" There are various ways but one of the more established approaches is using the target-behavior prediction. They rely on probabilistic approaches which involve predicting the location of a moving target using probability theory. The target's location is represented through a continuous or discrete probability distribution, which is propagated over time based on a stochastic process [7][21]. For example, a study in Alberta, Canada, analyzed missing-person incidents in wilderness areas using the Wakeby distribution to categorize wilderness users and planned the search boundaries [19]. In marine SAR scenarios, a floating target influenced by winds and waves was examined, using a Markov motion model to propagate the target's location probability density function [7][18]. More advanced models like iso-probability introduced in Reference 25, incorporate dynamically changing conditions like terrain, target physiology, and psychology. It uses curves that adjust over time with new information, offering a more adaptable and accurate method for target prediction and defining the area of interest. It starts from the Last Known Position (LKP) of the target. From this point, it is assumed that the target can move in any direction, within a full $[0^o, 360^o]$. The process involves estimating the maximum distance a person could travel in a straight line from the LKP over time. To construct the iso-probability curves, the method involves determining the worst-case motion scenarios and utilizes probabilistic data about the target's peer group, such as average walking speeds. The approach uses rays emanating from the LKP to map potential paths of the target, overlaying a probability distribution for the target's location at each point in time onto these rays. This results in the creation of iso-probability curves, which are 2-D closed contours that represent different levels of likelihood for the target's location at various times. This overcomes the limitations of previous simple static models by considering a broader range of influencing factors and adapting to real-time data. These curves help in guiding search efforts by providing a probabilistic framework for estimating the most probable areas where the target might be found, taking into account factors like terrain and movement patterns. This method enhances the efficiency of coverage search strategies by delineating the areas that need to be searched, supporting less exhaustive coverage strategies.

It is apparent that WiSAR strategies captialize on UAVs' ability to traverse expansive, uneven terrains, as seen in the Modified Lawn-Mower or the LHC_GW_CONV. The incorporation of 'global warming' features and the Mode Goodness (MG) ratio further refine these approaches, optimizing search patterns to prioritize areas of high probability. Yet, the complexity of urban environments diminishes UAV effectiveness due to a greater likelihood of overlooking targets. Consequently, Urban Search prefers UGVs for their capability of conducting close-range searches, less chance of missing the target, at the cost of the speed and wide-range peripheral that UAVs offer. This discrepancy indicates the limitation in directly applying WiSAR methodologies to Urban Search. This emphasize the necessity for research in efficient, rapid UGV-based search methods in urban settings.

2.2 Theoretical Graph Approaches

Reformulating the framework as a graph problem calls for a detailed examination of graph theory to uncover adaptable methodologies or identify limitations that could highlight research gaps. Graph theory encompasses a wide array of problems including connectivity analysis and network max flow. This study centers on the domain between graph traversal and path planning, aiming to ensure coverage and increase the likelihood of locating a lost individual by navigating towards regions with higher probabilities - directly aligning with the primary goal of the research. Therefore, both graph traversal and graph path planning algorithms will be examined.

2.2.1 Graph Traversal

Graph traversal involves systematically visiting each vertex in a graph to achieve computational objectives like node searching, graph structure mapping, or node relationship analysis. Two prominent methods used are Breadth First Search (BFS) and Depth First Search (DFS). BFS systematically explores the graph's vertices by visiting all neighbors of a starting vertex before moving to the next level of neighbors, ensuring a level-by-level exploration. On the other hand, DFS dives deeper into the graph, exploring as far as possible along each branch before backtracking.

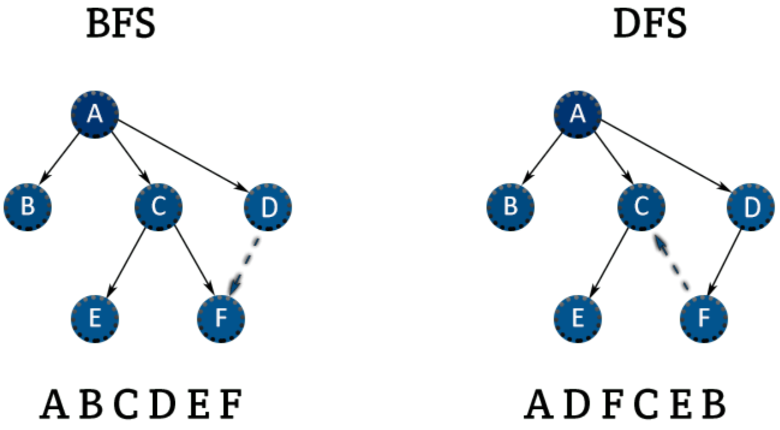


Figure 2: BFS-DFS Algorithm [20]

In the context of path for Unmanned Ground Vehicles (UGVs), the method must facilitate efficient and smooth and navigation. However, the inherent mechanisms of BFS and DFS prioritize complete coverage over efficiency, meaning it will design a suboptimal trajectory. Also, their algorithms hold discontinuity, creating abrupt transitions between nodes. For example, notice in Figure 2, for BFS there is jump between node D to node E. Similarly, for DFS, there is discontinuity between node E to node B. This lack of smooth trajectory implies the necessity to backtrack to previous junctions to continue the path, which is again not ideal - revisiting to reach other node it wants to travel to. These limitations highlights the need for tailored strategies that align with the unique navigational requirements of UGVs, ensuring seamless and coherent pathfinding.

2.2.2 Graph Path Planning

Graph path planning is a computational problem that involves finding the shortest path between two nodes in a graph. Among the most renowned algorithms for addressing this challenge are A* and D* [26]. The A* algorithm combines the strengths of Dijkstra's algorithm, known for its efficacy in identifying the shortest path, with the speed of Greedy Best-First Search in reaching the target node. On the other hand, D*, or Dynamic A*, enhances A* by incorporating incremental graph search, enabling the optimization of paths in dynamic environments.

Despite their advantages, these methods encounter specific limitations when applied to urban search operations. The primary goal in such contexts extends beyond simple point-to-point navigation; it encompasses maximizing the search area coverage to enhance the probability of locating missing individuals. This objective demands more than just finding the shortest path; it requires a certain level of scanning of the area, which these algorithms might not efficiently support. The focus of A* and D* on the shortest path can inadvertently lead to overlooking extensive areas, thus potentially reducing the effectiveness of search operations in urban environments.

Given the unique challenges of Urban Search, traditional WiSAR strategies and classic graph algorithms reveal limitations when directly applied to urban environments. WiSAR methodologies, although adept at covering extensive and uneven terrains using UAVs, do not seamlessly translate to the densely structured and unpredictable nature of urban settings where UGVs are preferable. On the otherhand, while classic graph algorithms like A* and D* are efficient in pathfinding, they primarily focus on point-to-point navigation, often overlooking the broader requirement of area coverage essential for designing optimal path.

These constraints underscore the need for a nuanced approach that leads to more efficient trajectory. Here, Reinforcement Learning (RL) emerges as a potential paradigm-shifter, offering a dynamic and adaptable framework suitable for the complex demands of Urban Search. Capitalizing on RL's ability to balance exploration and exploitation, a more tailored and effective search strategy can be developed. This proposed RL framework envisions a system that is both responsive to the urban environment's intricacies and adept at making precise, strategic decisions to facilitate timely and efficient searches.

The subsequent section will examine the foundational principles behind the innovative approach designed to achieve an optimal balance between extensive area coverage and enhancing the probability of locating the target. It will detail the architectural building blocks that underpin this advanced search strategy, shedding light on how these elements synergize to form an effective solution.

2.3 Foundations for the Proposed Framework

Reinforcement Learning (RL) is a branch of machine learning where an agent learns to make decisions by interacting with its environment, aiming to achieve optimal behavior through trial and error rather than by following explicit instructions. In RL, the agent assesses its situation, executes actions, and receives feedback in the form of rewards, which gauge the effectiveness of its actions. The essence of RL lies in developing a policy—a strategy dictating the agent's actions in various states to maximize cumulative rewards over time. This process necessitates a balance between exploring new strategies and exploiting known ones, distinguishing RL from other learning paradigms by its focus on learning from the consequences of actions, making it particularly suited for dynamic and time-sensitive decision environments.

2.3.1 Introduction to Reinforcement Learning (RL)

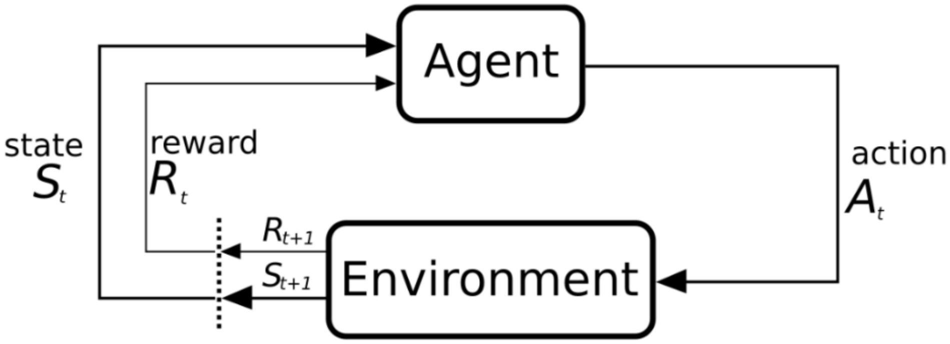


Figure 3: Base Structure of Reinforcement Learning [3]

Understanding the RL framework involves delineating its core components: the Agent and the Environment. The Agent observes its state within the Environment and makes decisions (actions) based on this state, aiming to achieve predefined goals and receiving rewards in response. The central challenge in RL is identifying the actions that lead to rewards, a task managed by formulating a policy $\pi(s, a) = \Pr(a = a|s = s)$, which dictates the actions a to take in state s to optimize future rewards in a probabilistic setting [8].

To devise an effective policy, it's crucial to comprehend the value of each state s under that policy. The value function $V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} r_t \mid S_0 = s\right]$ represents the expected reward from each state, analogous to strategizing in a chess game where not all board states are known, but certain positions offer strategic advantages. The value function is central to refining the policy, directing the agent towards beneficial states. The goal of the RL framework is to optimize this policy to maximize future rewards, employing various strategies such as differential programming [42], Monte Carlo methods [41], temporal difference (TD) learning [38], and the Bellman equation [6] for optimization.

Expanding upon the fundamental principles of reinforcement learning, we observe how agents assess and adapt to their environments. It is imperative to explore the strategic frameworks behind these interactions, as they significantly influence the effectiveness of learning outcomes.

Reinforcement learning methodologies are generally divided into model-based and model-free approaches, as shown in figure 4. Each rely on different degrees of environmental understanding. This categorization is critical, as it reflects the adaptability of RL strategies to the varying complexities and uncertainties encountered in practical applications [4]. In the Model-based approach, the agent possesses some understanding or 'model' of the environment's behavior. In other words, it has a simplified representation of the world that can aid the agent in decision-making. In contrast, Model-free Reinforcement Learning is used when the agent lacks a definitive model of the environment. Here, the agent learns directly from its own experiences, operating through trial and

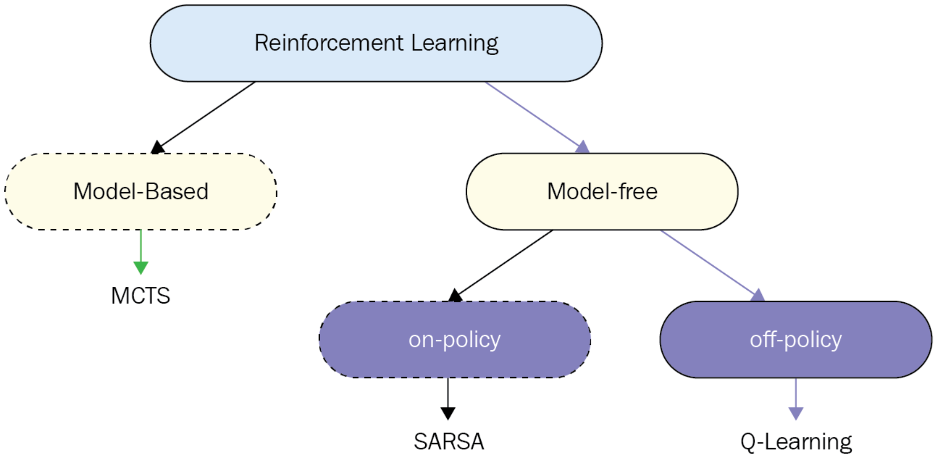


Figure 4: Branches of Reinforcement Learning [3]

error rather than a pre-established understanding of environmental dynamics. It can be further split into on-policy and off-policy algorithms:

- **On-Policy Algorithm**: The agent strictly adheres to its current policy during exploration. This approach ensures that the agent acts according to what it deems best based on its current knowledge, even though the policy may not be optimal. State-Action-Reward-State-Action (i.e., SARSA) is a typical on-policy method, known for its conservative learning approach.

- **Off-Policy Algorithm**: These allow the agent to explore actions outside of its current policy, akin to testing out new strategies that may initially appear suboptimal. A prominent algorithm in this category is Q-learning, which aims to ascertain the quality (Q) of particular actions in given states. Off-policy methods often provide more efficient learning and typically achieve faster convergence.

Given the need for the agent to perform trajectory planning across various maps without a definitive map, the problem naturally aligns with the use of an off-policy algorithm. One of the primary goals is to ensure a sufficient level of exploration (coverage). Since off-policy algorithms facilitate this type of exploration, they are deemed more appropriate for this application [17]. As an off-policy method, Deep Q-Learning guides the policy for behavior generation—typically an epsilon-greedy strategy—from the evaluated and improved greedy policy. This distinction allows for a balance between the exploitation of the best-known strategy and the exploration necessary for extensive learning. Consequently, Q-learning has been selected as the algorithm of choice for this problem.

Q-learning is a comprehensive approach that concurrently learns both the policy and the value functions. The Q-function, assessing the utility of specific state-action pairs, is updated as follows:

$$Q_{\text{update}}(s_t, a_t) = Q_{\text{old}}(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q_{\text{old}}(s_{t+1}, a_t) - Q_{\text{old}}(s_t, a_t) \right)$$

In this equation, $r_t + \gamma \max_a Q_{\text{old}}(s_{t+1}, a_t)$ represents the agent's observed cumulative reward as it interacts with the environment. The term $(r_t + \gamma \max_a Q_{\text{old}}(s_{t+1}, a_t) - Q_{\text{old}}(s_t, a_t))$ denotes the Temporal Difference (TD) Error. This error measure guides the update of the Q-values, which are crucial for learning optimal actions. The learning rate, denoted by α, and the discount factor, denoted by γ, are parameters that influence how significantly future rewards affect current decisions, thereby guiding the agent to optimize actions for the greatest long-term benefit. Through iterative process, governed by the TD learning equation, convergences to the optimal Q-function $Q_i \rightarrow Q^*$. In these iterations, the agent continually refines its strategy, enhancing the policy to secure the most favorable outcomes. From these Q-values, the value function and policy function can be deduced as follows:

$$V(S) = \max_a Q(s, a) \tag{1}$$

$$\pi(s, a) = \arg\max_a Q(s, a) \tag{2}$$

These equations direct the agent's choices toward the best possible actions in any given state [9].

2.3.2 Deep RL and Enhancement Techniques

Reinforcement Learning (RL) has historically faced significant challenges in path planning due to the "curse of dimensionality," necessitating handcrafted features for robot state representation. This limitation confined its application to fully observable, low-dimensional environments. However, the advent of Deep Reinforcement Learning (DRL) creates a substantial shift. DRL enhances RL's capabilities by automatically learning state features through iterative interaction with the environment, significantly reducing the dimensionality challenge [24].This advancement is exemplified in the study 'DRL Robot for SAR in Unknown Cluttered Environments,' which integrates frontier-based exploration with a DRL network to enable adaptive learning and efficient information gain in robotic navigation [27]. This approach allows robots to autonomously adapt their exploration strategies based on the environment's layout, demonstrating enhanced effectiveness in identifying optimal exploration frontiers and robustness across diverse environmental conditions.

Similarly, DRL tackles the limitations of traditional Q-learning, which traditionally relies on a tabular approach to record Q-values for each state-action pair—a method that becomes untenable as environments increase in complexity and state-action spaces enlarge. Deep Q-Learning, an advanced iteration of traditional Q-learning, utilizes neural networks as function approximators to estimate Q-values, denoted by $Q(s, a; \theta)$. These networks, parameterized by θ, aim to emulate the optimal Q-function $Q^*(s, a)$, and are trained by minimizing the following loss function at each iteration i:

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s' \sim p(\cdot)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i) \right)^2 \right] \tag{3}$$

The Temporal Difference (TD) target is defined as $r + \gamma \max_{a'} Q(s', a'; \theta_{i-1})$, and the expression $r+\gamma \max_{a'} Q(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)$ represents the TD error. The probability distribution $p(\cdot)$ over transitions $\{s, a, r, s'\}$ captures the likelihood of occurrences as the agent interacts with the environment [37].

Despite their advanced function approximation capabilities, neural networks can introduce volatility during training phases, attributable to the complex interdependence of states and actions. Fluctuations in network parameters can lead to instability and do not always assure convergence to the optimal value function.

To mitigate these challenges, the Deep Q-Network (DQN) framework integrates two pivotal enhancements: Experience Replay and the Target Network. Experience Replay boosts training efficiency and stability by leveraging a replay buffer. It draws random mini-batches of transitions to calculate the loss and gradient instead of depending on sequential experiences. This technique not only extends the utility of individual transitions but also breaks potential correlations in the training data sequence. The result is a stronger and more varied dataset for updating the network [33][37]. Target Network, central idea to the Double DQN variant, provides stability by updating its parameters θ_{i-1} less frequently and using them to generate the Temporal Difference (TD) targets. It decouples the target value generation from the parameter updates, offering steady TD targets for the primary network's approximation and enhancing the algorithm's stability [16]. These adaptations are essential for deep reinforcement learning, enabling stable and reliable convergence towards optimal action values.

2.4 Integration of Graph Theory with DRL

2.4.1 Graphical Representation

Building on the advancements in DRL, the application of graphical representations can further alleviate the dimensionality problem. By transforming Urban Search into a graph problem, the map data is restructured into a graph format, effectively condensing noise and extracting crucial information such as roads and intersections into edges and nodes. This transformation makes the data more compact and manageable, aligning it more closely with RL's preference for structured, Euclidean data [12].

Despite concerns that restructuring may lead to significant information loss, graphical representation actually enhances the architectural framework by facilitating the application of advanced graph-based techniques. One such technique is Deep Graph Infomax (DGI), which leverages the structured data to amplify the utility of the information within the graph. DGI applies unsupervised learning on graph-structured data, aiming to maximize the mutual information between local node representations and a global graph summary. This process involves generating node embeddings through a Graph Convolutional Network (GCN), which captures nuanced information such as feature-based context of each node. A readout function then aggregates these embeddings to form a comprehensive graph summary. By maximizing mutual information, DGI encourages the model to retain essential information in the node embeddings, effectively representing the entire graph. This transformation not only condenses the data but also enriches it, ensuring that only the most pertinent information is emphasized.

2.4.2 Graphical Neural Network

The translation of data from map into a graphical format allows the application of Graph Neural Networks (GNNS) as well. Many DRL methods using standard neural networks, such as Recurrent Neural Networks (RNNs), struggle with generalization in dynamic environments. This poses a challenge for their deployment in networks with changing topologies. However, GNNs are adept at generalizing across various graph structures and sizes. This characteristic enables GNN-based DRL agents to effectively learn and adapt to diverse environmental network topologies. Consequently, GNNs are excellent for transferring knowledge efficiently, even when dataset sizes in input networks vary from those used in training [12]. This ability to generalize to untrained maps is particularly valuable, ensuring that the agents remain effective in unfamiliar scenarios.

2.4.3 GNN and DRL Applications

GNN-based RL has shown promising performance across various domains. The study, *Deep Reinforcement Learning meets Graph Neural Networks: Exploring a Routing Optimization Use Case*, presents a method to optimize routing in Optical Transport Networks (OTNs) by integrating GNNs with DRL [5]. The authors address the challenge of generalizing DRL models to unseen network topologies and demonstrate that the DRL and GNN model significantly improves performance by effectively learning and generalizing over arbitrary network topologies. The paper does extensive testing on both synthetic and real-world network topologies, showing the model's ability to outperform state-of-the-art solutions in topologies it was not trained on, highlighting its potential for practical deployment in dynamic network environments. Another notable application of GNN-based DRL is in Autonomous Mobility-on-Demand Systems. A recent study demonstrates that this technology enhances policy transferability, scalability, and generalization, effectively handling diverse urban scenarios [14]. Through detailed simulations, the framework showcased superior performance across various environments, adeptly adapting to different, complex urban topologies. The findings indicate that GNN-based DRL provides substantial advantages for Autonomous Mobility-on-Demand systems, especially in its ability to generalize across varied network structures. This capability is particularly crucial for deploying UGVs in densely populated urban areas, highlighting the robustness and adaptability of GNN-based DRL in dynamic and challenging settings.

The successful deployment of GNN-based DRL across diverse domains demonstrates its robustness and lays a solid foundation for its application in Urban Search. The case studies above underscore the versatility and adaptability of GNNs, particularly their capacity to generalize effectively to previously unseen environments. This adaptability, combined with enhanced scalability and transferability, is indispensable for the deployment of UGVs in search. The ability of GNN-based DRL to manage and interpret dynamic network topologies ensures that UGVs can operate efficiently and responsively. This sets the stage for discussion in the following sections on how these structures are tailored and implemented.

3 Methodology

The methodology of this study is divided into two main phases: 1) generating and preprocessing data into a graph, and 2) designing the architecture by integrating all components.

3.1 Data: Lost Person Simulator

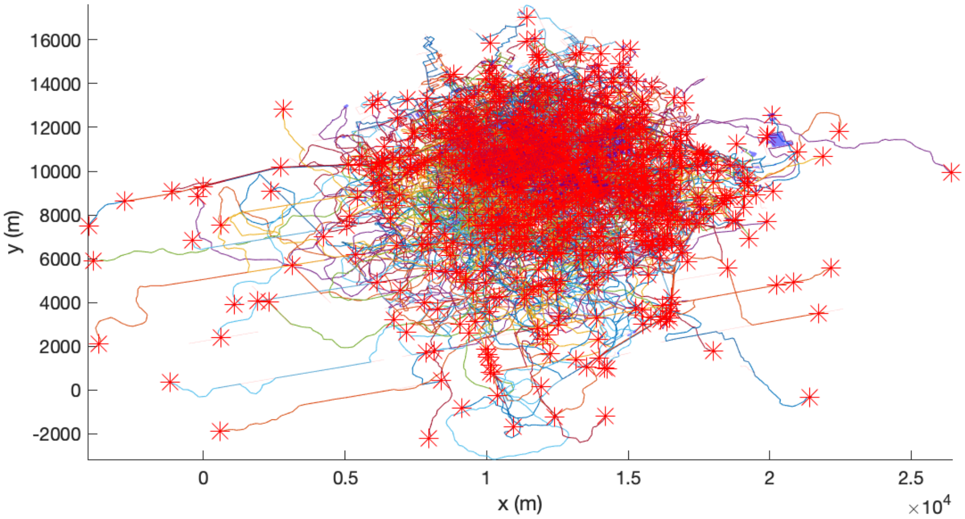
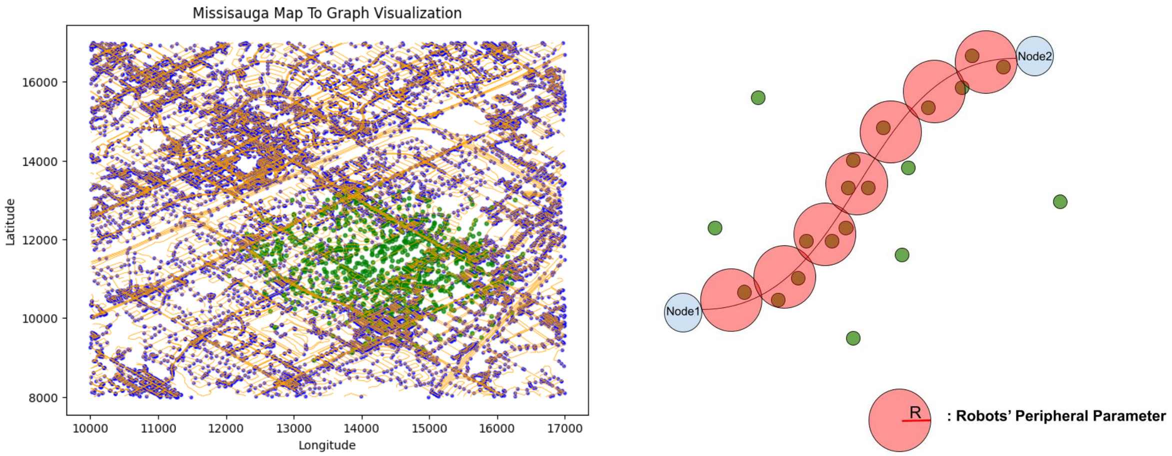


Figure 5: Toronto Potential Target Position Visualization [2]

The procedure initiates with data generation using the Lost Person Simulator, developed by Cameron Haigh [2]. This simulator utilizes data from Open Street Maps (OSM) to create detailed environmental maps and employs a parametric behavior-based stochastic model to simulate possible routes of a lost person over a specific period. The model's parameters are fine-tuned using stochastic gradient descent, calibrated against demographic data of historically lost persons. The simulator outputs a range of probable locations for the lost person, depicted in Figure 5, by conducting numerous Monte Carlo simulations to capture the variability of potential paths.

3.2 RL Environment: Map to Graph

Following, data collection, the environmental map is transformed into a graph to construct the RL environment. From map to graph, as shown in 6a, road intersections (blue) turn into nodes and connecting roads (orange) turn into edges. Each edge is weighted based on the simulation, reflecting the number of potential lost person sightings within the robot's adjustable peripheral range, as shown in figure 6b. This graph-based environment sets up the groundwork to apply graph-theoretical methods to optimize



(a) Graph features on a Map

(b) Simulation Data to Weight

Figure 6: Transformation from Map to Graph

search strategies.

3.3 Overall Architecture

At the core of this novel architecture lies the integration of advanced graph methods - Deep Graph Infomax (DGI) and Grpah Neural Neteworks (GNNs) - within the Deep Reinforcement Learning (DRL) framework using Deep Q-Network (DQN), as in Figure 7. DGI enhances the learning process by maximizing mutual information between local patch representations and global graph summaries, aiding in pinpointing significant nodes within the search graph. Concurrently, GNNs exploit these node representations to delineate probable paths for locating lost persons swiftly and accurately.

The DQN framework is further enhanced with a Prioritized Replay Buffer and 5-step temporal difference learning. The Prioritized Replay Buffer prioritizes experiences that offer greater learning potential, thus optimizing the learning process by frequently revisiting crucial experiences. Meanwhile, the 5-step temporal difference learning extends the look-ahead capability of the agent, enabling it to evaluate the outcomes of its actions over a longer sequence of steps. This extended evaluation helps the model to better understand the consequences of its actions, supporting more strategic decision-making that accounts for future states instead of only immediate rewards.

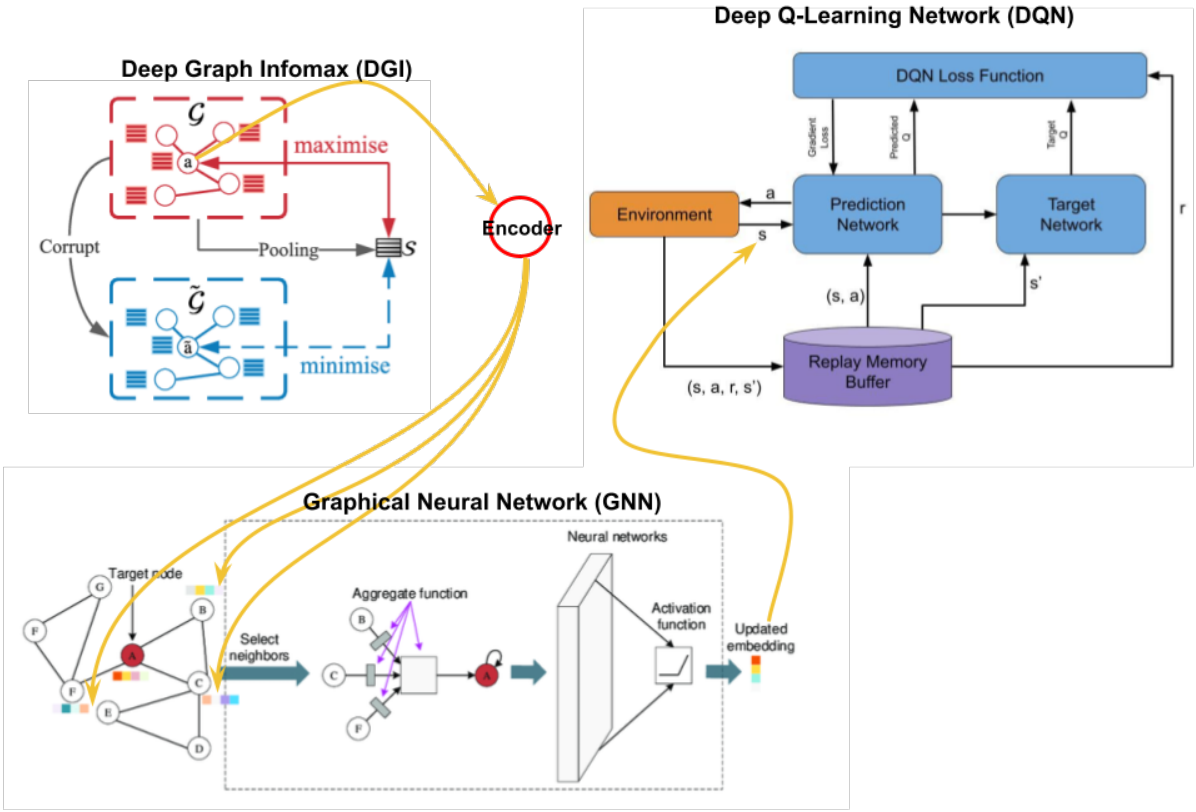


Figure 7: Overall Architecture Visualization [34][43]

3.4 Experiment

The computational demands of Reinforcement Learning (RL) pose significant challenges when applied to extensive environments, such as entire city maps, especially with limited resources. To manage this, a scaled-down graph environment was constructed to demonstrate the functionality and effectiveness of the proposed approach.

For comparative analysis, a Greedy strategy was implemented, which selects the neighboring node offering the highest immediate reward. This strategy is relevant in the context of Wilderness Search and Rescue (WiSAR) operations, where greedy strategies are commonly employed on various factors such as fitness levels. This approach also mirrors classic graph path planning strategies, exemplified by the state-of-the-art algorithm, A*, which combines elements of Dijkstra's and Greedy Best-First Search methods. To

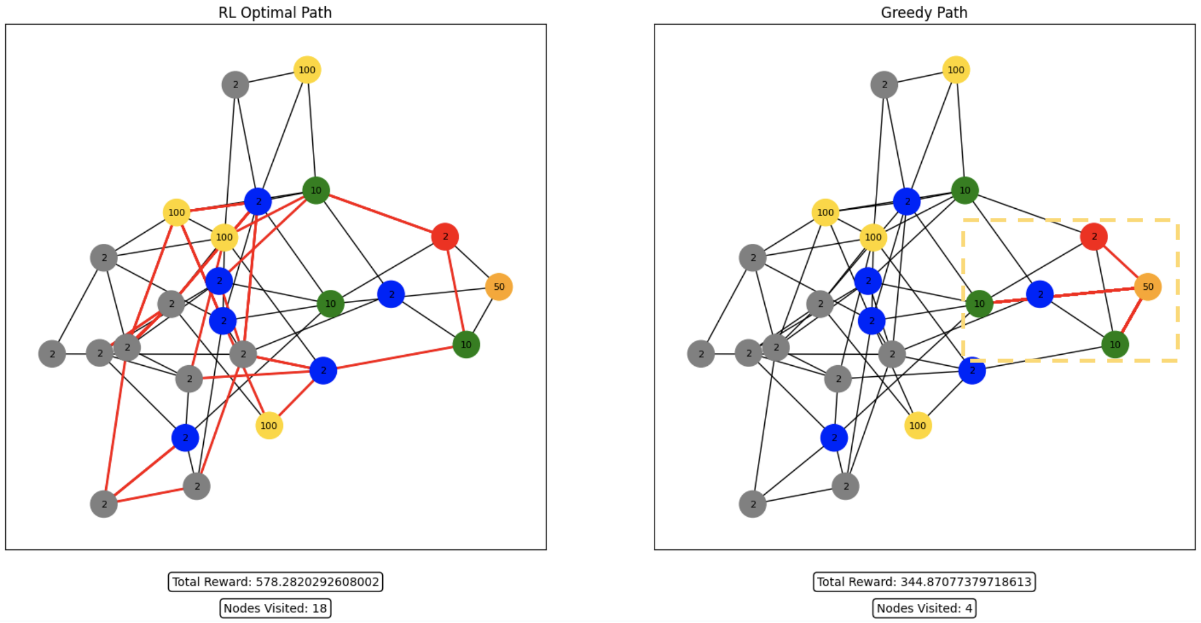


Figure 8: Sample Illustration of Scenario with Suboptimal Solution

discourage revisiting the same node—a strategy that yields no new information in search contexts—the Greedy solution's reward is reduced by a factor of 0.5 with each visit.

The first objective of these experiments was to evaluate the strategic decision-making capabilities of the RL agent in scenarios characterized by suboptimal solutions. The experimental environment was specifically designed to challenge the agent's ability to prioritize long-term benefits over immediate rewards, requiring navigation through areas with lower immediate reward to achieve higher cumulative gains. Figure 8 provides a visualization of the paths generated by each the RL and Greedy strategies in this synthetic environment. It illustrates how the Greedy strategy gets stuck in a suboptimal path, highlighted in yellow box on Figure 8 for the Greedy solution, whereas the RL strategy successfully identifies and follows more beneficial trajectories, thereby demonstrating superior strategic decision-making.

To ensure a robust evaluation, the experiments were conducted across 50 synthetic graphs specifically designed to simulate the unique challenges of the study. Each agent's journey was capped at 30 steps to standardize testing conditions.

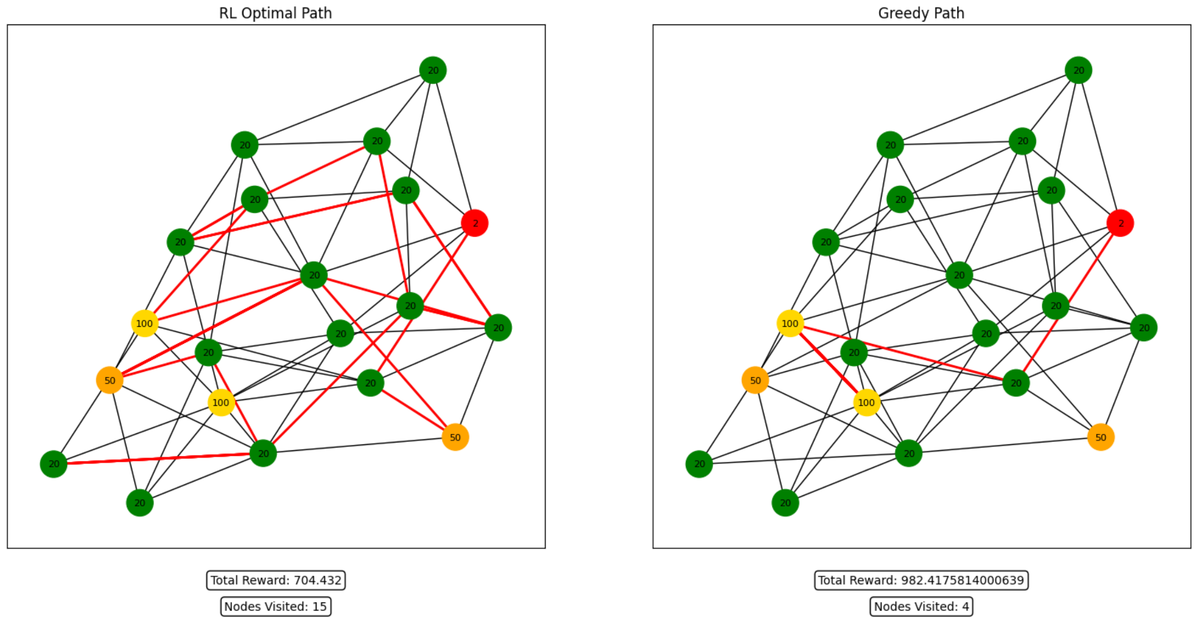


Figure 9: Sample Illustration of General Scenario

In a subsequent phase of the study, see Figure 9, both the RL and Greedy agents were assessed under more generalized conditions with randomly distributed rewards, differing from the strategically tailored scenarios initially tested. This phase again involved simulations on 50 different graphs, with each agent limited to a maximum of 30 steps. The goal was to observe their performance under varied and unpredictable conditions, further validating the flexibility and adaptability of the RL strategy compared to the Greedy approach.

4 Results

4.1 Evaluating Search Strategies

From the initial experiments, the results offered some interesting insights. The Greedy strategy, surprisingly, yielded a higher average in total rewards as shown in Figure 10. Recall that the rewards metric represents the potential lost person locations (simulation data) identified by the robot. At first glance, this could suggest the superiority of the Greedy approach over the RL method. However, a closer look at the nodes visited tells a

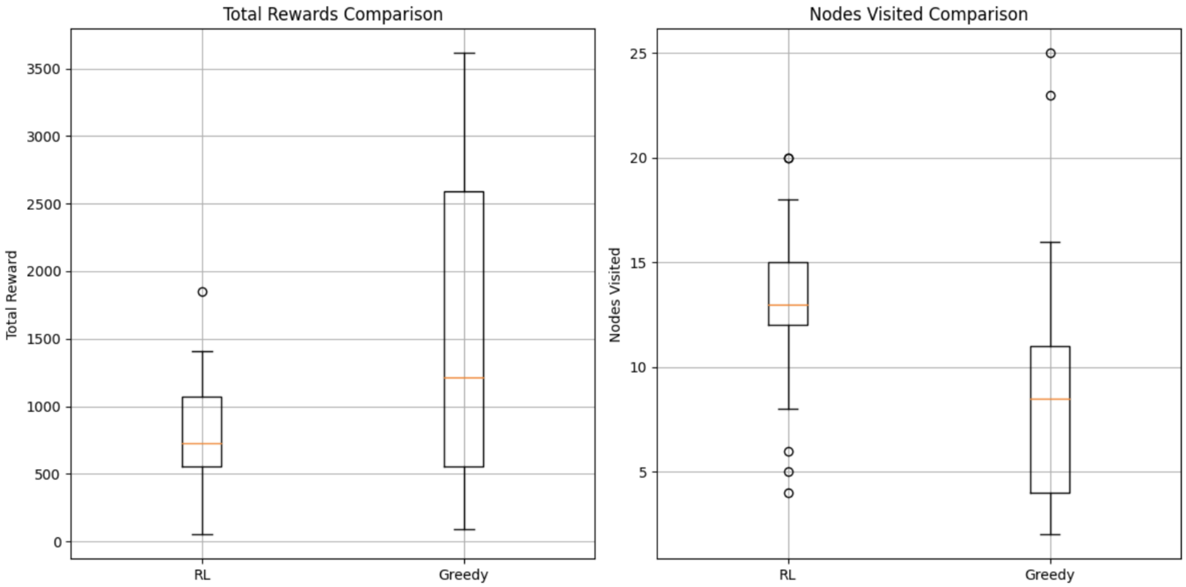


Figure 10: Result Comparison in Designed Scenarios

more nuanced story. While the Greedy solution might have accumulated greater immediate rewards, it lacked in breadth of exploration, tending to revisit the same high-reward nodes repeatedly. In contrast, the RL method exhibited a broader area of coverage, crucial for search operations where the objective is to maximize search coverage while maintaining a high likelihood of locating the lost person. Therefore, the apparent advantage of the Greedy solution in total rewards does not translate into effectiveness in a search context, where it fails to achieve comprehensive coverage.

Subsequent experiments in more general scenarios also confirmed these patterns as illustrated in Figure 11. The Greedy strategy continued to achieve higher total rewards, but the RL agent surpassed it in the number of nodes visited. This consistent trend across various experimental conditions, highlights the inherent limitation of the Greedy approach: its sacrifice of coverage for immediate gains. The results robustly confirm the RL agent's capacity to navigate an optimal path that judiciously balances comprehensive search coverage with the probability of locating a lost person, thereby endorsing the RL strategy for more effective and wide-ranging search operations.

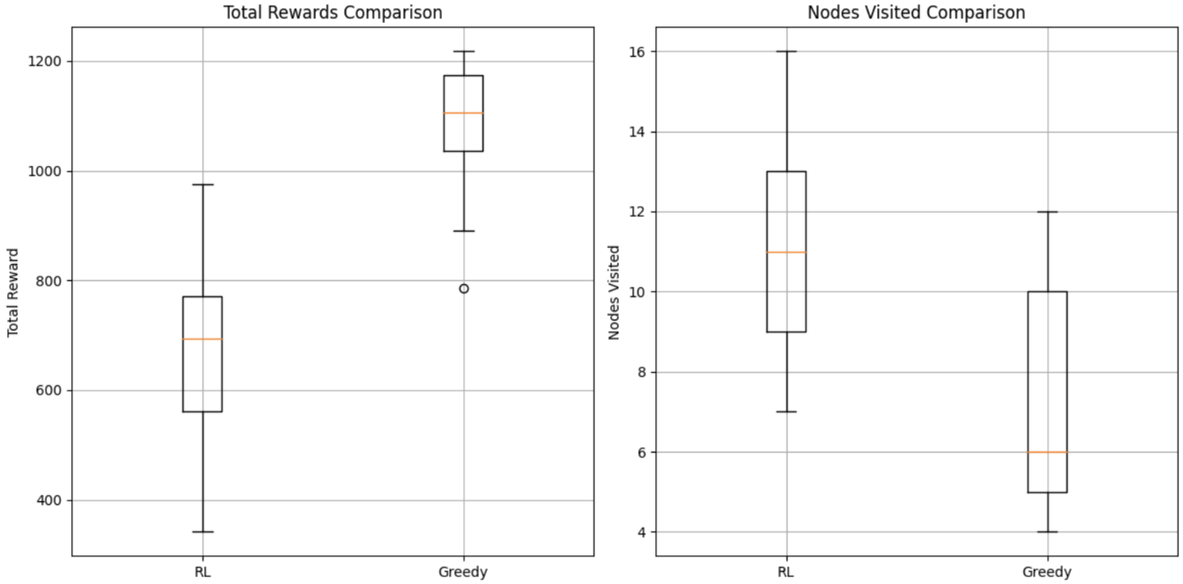


Figure 11: Result Comparison in General Scenarios

4.2 Real-World Urban Scenario Validation

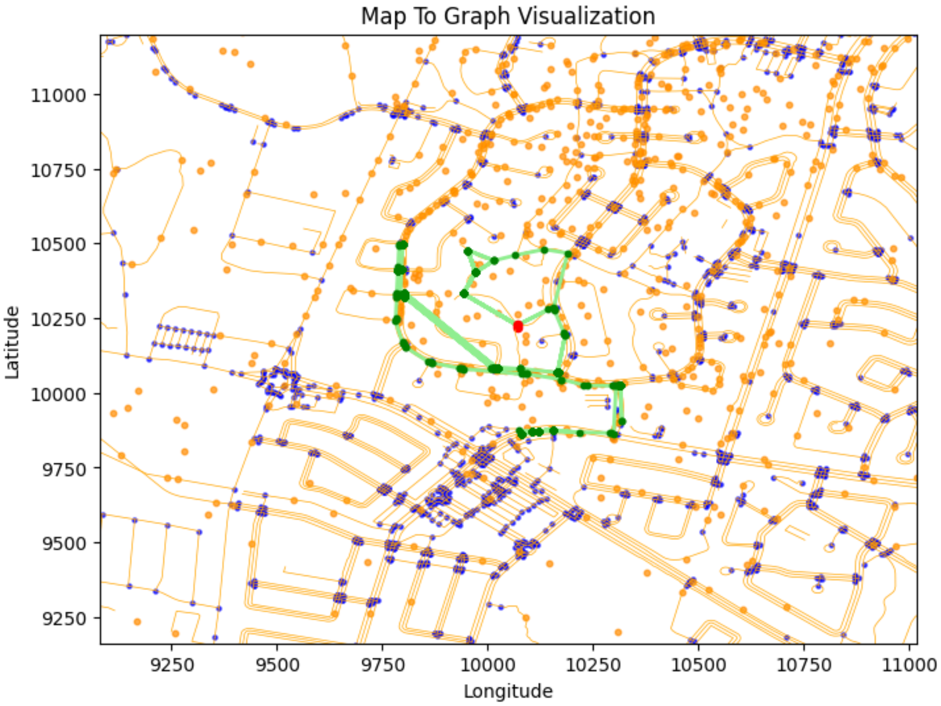


Figure 12: Agent's Search Trajectory on Mississauga Map

The method has been initially evaluated on a smaller scale due to computational constraints. Subsequently, this is validated on a real-world framework. Figure 12 showcases the method's scalability by applying it to the diverse and intricate cityscape of Mississauga, Ontario, Canada. In this depiction, intersections, denoted in blue, correlate to graph nodes, while the connecting orange road segments correspond to graph edges. Overlaying these, the orange dots are potential locations of a lost individual, informed by simulation data, which determine the weights on the graph. The paths marked in green are the result of the Reinforcement Learning algorithm's decision-making, post-training, in evaluation mode.

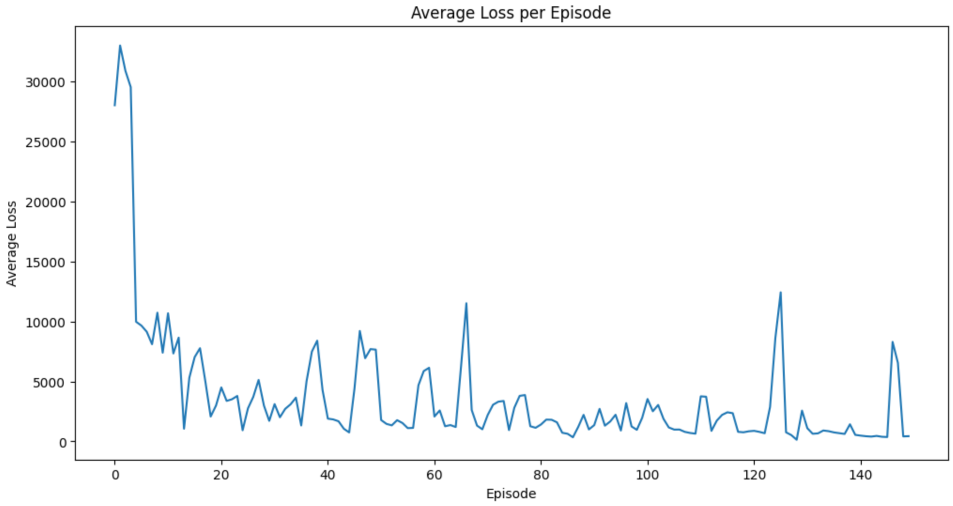


Figure 13: Train Loss from training RL Agent on a real Urban Environment

The agent underwent training in three different cities - Toronto, Mississauga, and Kitchener - across 200 epochs with 100 iterations each to prevent overfitting and to enhance its robustness across different urban landscapes. The downward trend in training loss, as depicted in Figure 13, validates the training's effectiveness. Overall, these visualizations demonstrates the method's capacity to adapt to more complex and extensive urban settings, indicating its potential for practical implementation in real-life search operations.

5 Discussion

The primary goal of this study was to design an algorithm that optimizes search trajectories for Unmanned Ground Vehicles (UGVs), ensuring a judicious balance between coverage and the likelihood of locating a target within a constrained time frame. Preliminary evaluations of this design show promise in meeting these objectives. The proposed integration of graph theory and Deep Reinforcement Learning (DRL) represents a contribution to the field of urban search, providing a new perspective through which complex urban environments may be navigated more effectively.

The research has brought a synergy between graph-theoretical models and DRL algorithms to light that is particularly well-suited to the dynamics of urban scenarios. By aligning search strategies with the new setup, the study advances the current understanding of search optimization in urban settings. This represents a meaningful enhancement over traditional search methods, which may lack the sophistication to account for urban complexity.

It is important to note, however, that this study is not without limitations. The reliance on simulations for preliminary evaluations, while necessary and informative, calls for evaluation in real-world trials for further validation. Moreover, the scope of single-agent operations, though insightful, is but a precursor to the more complex but realistic scenario of multi-agent coordination, which the future work should address.

Future research directions should include a thorough investigation into refinement of reward design. The current data processing framework captures and stores distances as node attributes but these metrics are not actively utilized in the reward system. The integration of distance and temporal factors within the reward system could yield a more nuanced approach to trajectory optimization, considering the variable speeds at which different UGVs may operate and the time-critical nature of search missions. An ablation study would complement this by quantifying the impact of each system component, providing guidance for targeted enhancements. Furthermore, expansion to more dynamic and detailed maps will add a layer of realism to the simulation environment, paving the way for a comprehensive methodology that could revolutionize urban search operations on a broader scale.

6 Conclusion

This study has pioneered a novel approach to urban search operations, leveraging the strengths of UGVs, graph theory, and DRL. The research outcomes have demonstrated that this method can potentially exceed the operational efficiency and precision of traditional search techniques, marking a significant leap forward in the pursuit of more effective urban search strategies.

In contributing to the existing body of knowledge, this thesis underscores the feasibility of applying advanced computational methods to public safety challenges. This suggests a new paradigm in which urban search operations could be conducted. Contemplating the future of urban search strategies, the insights obtained from this work offer a foundation upon which more complex, adaptive, and comprehensive systems can be built.

The explorations and findings of this study not only provide a theoretical framework for subsequent research but also suggest practical applications that may one day be implemented in real-world scenarios. The path laid out by this thesis is poised to influence the development of search technology, offering a cornerstone for future advancements in the domain of public safety and emergency response.

Bibliography

[1] Air support unit - remotely piloted aircraft systems. https://www.scotland.police.uk/advice-and-information/air-support-unit/. Accessed: 2024-01-19.

[2] Lost person simulation.

[3] Understanding the impact of misspecification in inverse reinforcement learning. https://aihub.org/2023/05/04/understanding-the-impact-of-misspecification-in-inverse-reinforcement-learning/. Accessed: 2024-01-19.

[4] Dec 2023.

[5] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications*, 196:184–194, 2022.

[6] R. Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239, 1958.

[7] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. *Optimal search for a lost target in a Bayesian World*, pages 209–222. 2006.

[8] S. L. Brunton. *Machine learning meets control theory.* 2021.

[9] Steven L. Brunton. *Q-learning: Model free reinforcement learning and temporal difference learning*, Jan 2022.

[10] J. Ewers, D. Anderson, and D. Thomson. Optimal path planning using psychological profiling in drone-assisted missing person search. *Advanced Control for Applications*, 5(4), 2023.

[11] J.-H. Ewers. Optimum path planning for search and rescue, 2021.

[12] F. Fathinezhad, P. Adibi, B. Shoushtarian, and J. Chanussot. *Graph Neural Networks and Reinforcement Learning: A Survey*. 2023.

[13] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[14] D. Gammelli et al. Graph neural network reinforcement learning for autonomous mobility-on-demand systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021.

[15] Government of Canada CanadaMissing. Background - 2022 fast fact sheet. https://canadasmissing.ca/pubs/2022/index-eng.htm. Accessed: 2023-11-12.

[16] Hado Hasselt. Double q-learning. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[17] Michael Herrmann. Rl 5: On-policy and off-policy algorithms.

[18] C. D. Heth and E. H. Cornell. Characteristics of travel by persons lost in albertan wilderness areas. *Journal of Environmental Psychology*, 18(3):223–235, 1998.

[19] J. Hosking and J. R. Wallis. *Regional Frequency Analysis: An Approach Based on L-Moments*. Cambridge University Press, 2008.

[20] Board Infinity. Bfs vs dfs algorithms, Jul 2023.

[21] B. Lavis, T. Furukawa, and H. F. Durrant-Whyte. Dynamic space reconfiguration for bayesian search and tracking with moving targets. *Autonomous Robots*, 24(4):387–399, 2008.

[22] L. Lin and M. A. Goodrich. Uav intelligent path planning for wilderness search and rescue. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[23] L. Lin and M. A. Goodrich. Hierarchical heuristic search using a gaussian mixture model for uav coverage planning. *IEEE Transactions on Cybernetics*, 44(12):2532–2544, 2014.

[24] Z. Liu, Q. Wang, and B. Yang. Reinforcement learning-based path planning algorithm for mobile robots. *Wireless Communications and Mobile Computing*, 2022:1–10, 2022.

[25] A. Macwan, G. Nejat, and B. Benhabib. Optimal deployment of robotic teams for autonomous wilderness search and rescue. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[26] Yu Ming, Yanqiang Li, Zihui Zhang, and Weiqi Yan. A survey of path planning algorithms for autonomous vehicles. *SAE International Journal of Commercial Vehicles*, 14, 01 2021.

[27] Farzad Niroui, Kaicheng Zhang, Zendai Kashino, and Goldie Nejat. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2019.

[28] J. Ousingsawat and M. G. Earl. Modified lawn-mower search pattern for areas comprised of weighted regions. In *2007 American Control Conference*, 2007.

[29] K. Phillips et al. Wilderness search strategy and tactics. *Wilderness & Environmental Medicine*, 25(2):166–176, 2014.

[30] Public Safety Canada. Urban search and rescue (usar). https://www.publicsafety.gc.ca/cnt/mrgnc-mngmnt/rspndng-mrgnc-vnts/rbn-srch-rsc-en.aspx. Accessed: 2024-01-19.

[31] R. C. M. P. Government of Canada. Missing persons. https://www.rcmp-grc.gc.ca/en/missing-persons. Accessed: 2023-11-12.

[32] D. K. Rossmo, L. Velarde, and T. Mahood. Optimizing wilderness search and rescue: Discovery and outcome. *The Journal of Search and Rescue*, 6(1):37–45, 2023.

[33] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

[34] Sahil Sheikh. Exploring sageconv: A powerful graph neural network architecture, May 2023.

[35] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge. Multipurpose uav for search and rescue operations in mountain avalanche events. *Geomatics, Natural Hazards and Risk*, 8(1):18–33, 2016.

[36] S. S. Skiiena. *Algorithm Design Manual.* Springer, 2021.

[37] TensorFlow. Introduction to reinforcement learning. https://www.tensorflow.org/agents/tutorials/0$_intro_rl.Accessed : 2024 - 01 - 19$.

[38] W. Uther. *Temporal difference learning.* 2011.

[39] C. Vincent-Lambert, A. Pretorius, and B. Van Tonder. Use of unmanned aerial vehicles in wilderness search and rescue operations: A scoping review. *Wilderness & Environmental Medicine*, 34(4):580–588, 2023.

[40] S. Waharte and N. Trigoni. Supporting search and rescue operations with uavs. In *2010 International Conference on Emerging Security Technologies*, 2010.

[41] J.-C. Walter and G. T. Barkema. An introduction to monte carlo methods. *Physica A: Statistical Mechanics and its Applications*, 418:78–87, 2015.

[42] S. Yakowitz. *Algorithms and computational techniques in Differential Dynamic Programming*, pages 75–91. 1989.

[43] Yizhen Zheng, Shirui Pan, Vincent Lee, Yu Zheng, and Philip Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination, 06 2022.