GNN-Enabled Reinforcement Learning for Robust Task Admission and Routing in IoBT Environments

Athanasios Gkelias*, Kin K. Leung*, Patrick J. Baker†, Olwen Worthington‡
*EEE Department, Imperial College London, London, UK
†Rapid Capabilities Office, Royal Air Force, Farnborough, UK
‡Cyber & Information Systems, DSTL, Porton Down, UK
Email: a.gkelias@imperial.ac.uk, kin.leung@imperial.ac.uk, pbaker@dstl.gov.uk, olworthington@dstl.gov.uk

***Abstract*—Internet of Battlespace Things (IoBT) deployments in adversarial environments face critical challenges in network resource management, where rapid environmental changes and dynamic threats render traditional optimization approaches inadequate. The inherently volatile nature of these environments, where network conditions can change within seconds, necessitates algorithms that prioritize speed over perfect optimization to maintain operational effectiveness. This paper presents a novel Graph Neural Network (GNN) based Deep Reinforcement Learning (DRL) framework specifically designed for combinatorial task admission and routing optimization in dynamic IoBT networks. Our approach integrates Graph Attention Networks (GATs) for capturing network topology dependencies, Deep Sets encoders for permutation-invariant task processing, Adaptive Path GNNs for learning path representations, and statistical feature encoders that complement learned embeddings with interpretable routing heuristics. The system formulates network optimization as a Markov Decision Process, enabling real-time decision-making that maximizes task utility while respecting capacity constraints and adapting to topology changes. Comprehensive experimental evaluation across multiple network scenarios demonstrates that our Deep Q-Network (DQN) agent consistently outperforms greedy baselines by 5-58%, achieving near-optimal utility with sub-second inference times compared to Mixed Integer Programming solvers that require hundreds of seconds. The framework shows strong generalization capabilities, with agents trained on smaller task sets effectively scaling to larger workloads, and exhibits superior performance when trained under dynamic conditions rather than static environments.**

***Index Terms*—Graph Neural Networks, Reinforcement Learning, Combinatoric Optimisation, Networking**

I. Introduction

Modern IoBT deployments in adversarial and harsh environments present unprecedented challenges for network resource management and optimization. Mission-critical applications such as disaster response coordination, battlefield communications, emergency first responder networks, and infrastructure monitoring in extreme conditions demand robust, adaptive networking solutions that can maintain operational effectiveness despite dynamic threats, equipment failures, and rapidly changing environmental conditions. These scenarios are characterized by heterogeneous device capabilities, intermittent connectivity, resource constraints, and the constant threat of adversarial interference or physical damage [1].

This work was supported by the Dstl, UK, under the "SDS Continuation" Project. Gemini AI (Google) was used for editing and grammar enhancement.

Traditional network optimization approaches, which typically assume stable topologies and predictable traffic patterns, are inadequate for such dynamic adversarial environments. The combinatorial nature of resource allocation problems in these settings is further complicated by uncertainty in network state, unpredictable task arrivals, and the need for real-time decision-making under partial information. However, the challenge is amplified in IoBT environments by their inherently volatile nature, where network conditions, threat landscapes, and operational requirements can change within seconds or minutes, rendering previously optimal solutions obsolete before they can be fully implemented. For instance, defense networks must maintain operational capability while under active cyber or physical attack, requiring intelligent resource allocation that anticipates and mitigates potential disruptions, yet the fast-paced nature of combat scenarios demands immediate decision-making where a reasonably good solution implemented quickly often outperforms an optimal solution that arrives too late. This fundamental *speed-versus-optimality trade-off* necessitates algorithms capable of producing acceptable solutions within tight time constraints, prioritizing rapid adaptation over perfect optimization to maintain operational effectiveness in rapidly evolving IoBT environments.

This work addresses these challenges by developing a GNN based Reinforcement Learning (RL) framework specifically designed for combinatorial optimization problems in adversarial IoBT environments. Our approach leverages the inherent graph structure of communication networks to learn adaptive policies that maximize task utility while maintaining resilience against various forms of network perturbation. The integration of attention mechanisms, permutation-invariant task processing, and path-aware feature encoding enables the system to make informed decisions under uncertainty, balancing immediate operational needs with long-term network sustainability.

The application of GNNs to combinatorial optimization problems has emerged as a promising research direction, with significant contributions spanning multiple domains. [2] provides a comprehensive conceptual review of key advancements in applying machine learning techniques to combinatorial optimization, establishing the theoretical foundations for neural approaches to NP-hard problems, though their work focuses primarily on offline optimization rather than the real-

time, sequential decision-making framework we propose for dynamic network environments. In the context of network resource management, several studies have shown the potential of GNN-based approaches. [3] proposes Flex-Net, a novel GNN architecture for joint optimization of communication direction and transmission power in flexible duplex networks, achieving near-optimal performance while maintaining low computational complexity. However, their approach addresses physical layer optimization in wireless networks, whereas our work tackles the higher-level problem of task admission and routing in multi-hop networks with explicit utility maximization objectives. Similarly, [4] addresses dynamic resource slicing in 6G multi-access edge computing networks using message passing GNNs combined with online ADMM, demonstrating superior performance in highly dynamic and unreliable network scenarios. While this work shares our focus on dynamic environments, it concentrates on resource slicing rather than the combined task admission and routing problem we address, and lacks the reinforcement learning framework that enables our agent to learn from experience and adapt to diverse network conditions. The integration of attention mechanisms in graph-based optimization has been explored by [5], which proposes graph pointer networks with attention-based branching rules for combinatorial optimization, significantly outperforming traditional expert-designed heuristics. In contrast to their offline branch-and-bound approach, our work leverages attention mechanisms within a real-time RL framework that makes sequential decisions under uncertainty without requiring complete problem knowledge a priori. For sequential decision-making in dynamic environments, [6] develops a GNN-DRL framework for dynamic job-shop scheduling problems, formulating the optimization as a Markov decision process with disjunctive graph representations. While this work demonstrates the potential of GNN-RL combinations, it focuses on manufacturing scheduling rather than network optimization, and does not address the unique challenges of path selection and capacity constraints inherent in communication networks. The challenge of handling variable-sized inputs in optimization contexts has been addressed through Deep Sets architectures, as demonstrated by [7] on set prediction networks that maintain permutation invariance while enabling effective set-to-set learning. Our work extends this concept by integrating Deep Sets with GATs and path-aware encoders, creating a novel hybrid architecture specifically designed for the task admission problem where the set of pending tasks varies dynamically and must be processed alongside network topology information.

The key contributions of this research include: (1) a novel GNN-RL architecture that captures complex network dependencies while adapting to topology changes and capacity variations, (2) a comprehensive problem formulation that addresses task admission and routing under adversarial conditions, and (3) empirical validation demonstrating superior performance compared to traditional heuristics while maintaining computational efficiency suitable for real-time deployment in resource-constrained environments.

The remainder of this paper is organized as follows. Section II defines the network optimization problem and its computational complexity. Section III presents our GNN-RL system design, including the neural architecture components, state representation, and reward function. Section IV provides experimental evaluation comparing our approach against baselines across multiple network scenarios. Section V concludes with key insights and future research directions.

II. Network Optimization Problem: Maximizing Task Utility

In this work, we address a network optimization problem focused on maximizing the total utility of served tasks within a multi-hop communication network. The network comprises N nodes, including sink or switch nodes, where each directed link possesses a finite communication capacity. A pool of tasks, representing data flows, must be accommodated in the network. These tasks originate from random source nodes and are destined for distinct sink nodes, traversing multi-hop paths to reach their destinations. Each task is characterized by a unique data rate requirement, a utility value in the interval $(0, 1]$, and a randomly selected destination sink node. Importantly, routing is restricted to single paths, prohibiting the splitting of any task's flow across multiple routes. The primary objective is to select a subset of these tasks that maximizes the aggregate utility while adhering to the network's capacity constraints. This selection process must ensure that the chosen tasks can be routed without violating link capacities, thereby optimizing resource utilization in capacity-limited environments. Such problems arise in various other applications, including wireless sensor networks, data center traffic management, and telecommunication systems, where efficient allocation of bandwidth is critical to performance.

To formalize the problem, we define the following sets and parameters. Let $\mathcal{I}$ denote the set of tasks, indexed by i, and $\mathcal{L}$ the set of directed links, indexed by l, each with capacity C_l. For each task i, r_i represents its data rate requirement, u_i its utility value where $0 < u_i \leq 1$, o_i its origin source node, and s_i its destination sink node.

The decision variables include binary indicators $x_i \in \{0, 1\}$, which determine whether task i is selected, and $f_{i,l} \in \{0, 1\}$, indicating whether link l is utilized by task i. The optimization seeks to maximize the total utility:

$$\max \sum_{i \in \mathcal{I}} u_i x_i. \tag{1}$$

This objective is subject to several constraints. First, single-path routing is enforced: if a task is selected ($x_i = 1$), the selected links must form a valid path from o_i to s_i. This is achieved through flow conservation at each intermediate node n (neither o_i nor s_i):

$$\sum_{l \in \mathcal{L}_{\text{in}}(n)} f_{i,l} = \sum_{l \in \mathcal{L}_{\text{out}}(n)} f_{i,l}, \tag{2}$$

where $\mathcal{L}_{\text{in}}(n)$ and $\mathcal{L}_{\text{out}}(n)$ are the sets of incoming and outgoing links at node n, respectively.

Additionally, flow balance at the origin and sink nodes is required. For the origin node o_i:

$$\sum_{l \in \mathcal{L}_{\text{out}}(o_i)} f_{i,l} = x_i, \qquad \sum_{l \in \mathcal{L}_{\text{in}}(o_i)} f_{i,l} = 0. \tag{3}$$

For the sink node s_i:

$$\sum_{l \in \mathcal{L}_{\text{in}}(s_i)} f_{i,l} = x_i, \qquad \sum_{l \in \mathcal{L}_{\text{out}}(s_i)} f_{i,l} = 0. \tag{4}$$

Finally, link capacity constraints ensure that the total data rate on each link l does not exceed C_l:

$$\sum_{i \in \mathcal{I}} r_i f_{i,l} \leq C_l \quad \forall l \in \mathcal{L}. \tag{5}$$

The problem is inherently a combinatorial optimization problem (COP) due to the binary nature of the decision variables: x_i governs the discrete selection of tasks, while $f_{i,l}$ dictates the binary assignment of paths. This discreteness introduces combinatorial explosion, as the solution space grows exponentially with the number of tasks and links. The complexity is underscored by reductions to known NP-hard problems. When utilities and data rates are uniform, the problem simplifies to the unsplittable flow problem (UFP), which is established as NP-hard. In the presence of heterogeneous utilities, it generalizes to a weighted utility maximization variant of UFP, retaining NP-hardness. Even in simplified scenarios, such as those with single-link capacity constraints, the problem resembles the 0-1 knapsack problem, another NP-hard classic. These reductions highlight the computational intractability of finding optimal solutions for large instances, motivating the development of approximation algorithms or heuristics for practical deployment.

III. System Design

The aforementioned combinatorial optimization problem can be naturally reformulated within the reinforcement learning (RL) framework to enable efficient decision-making in dynamic environments. In RL, the problem is modeled as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ represents the transition probabilities, $\mathcal{R}$ is the reward function, and γ is the discount factor. The objective is to maximize the expected cumulative discounted reward: $\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t \mid \pi\right]$, where π is the policy and T is the episode length. In DQN, this is achieved by minimizing the temporal difference error in Q-value approximations, leading to an optimal policy $\pi^*(s) = \arg\max_a Q^*(s, a)$. For the task admission and routing problem, each task arrival corresponds to a time step, where the agent decides whether to admit the task and, if so, which path to assign, aiming to maximize long-term total utility while respecting capacity constraints.

Our implementation proposes a DQN agent that leverages a synergistic fusion of heterogeneous neural modules to produce robust, context-aware decision-making for network combinatorial optimization. At its core, the model integrates deep relational reasoning from Graph Attention Network (GAT) layers, permutation-invariant task-set embeddings via the Deep Sets encoder, and expressive sequential representations from the Adaptive Path GNN, complemented by hand-crafted statistical path feature encodings. In the final layers, these diverse feature spaces—graph-level, global network state, aggregated task-set, and fused path descriptions—are concatenated and processed via a sequence of fully connected layers, enabling the agent to holistically reason about the trade-offs inherent in each decision: immediate utility, future congestion risks, and opportunity costs. The explicit fusion of learned and statistical encodings allows the model to dynamically balance generic heuristics with nuanced, experience-driven strategies that adapt to changing network and traffic patterns.

A. Graph Attention Network (GAT)

In learning effective routing and resource allocation policies in variable network environments, a key challenge lies in capturing complex dependencies between nodes and links, especially when these relationships are highly dynamic and non-uniform. Traditional GNNs such as Graph Convolutional Networks (GCNs) treat all neighboring nodes equally, which limits their expressiveness in scenarios where certain links or nodes play a disproportionately important role for specific tasks or topologies. The Graph Attention Network (GAT) directly addresses this limitation through a trainable attention mechanism that enables the model to focus on the most informative neighbors during feature aggregation.

The GAT layer computes attention coefficients α_{ij} between nodes i and j using a shared attention mechanism a:

$$e_{ij} = a(\mathbf{W}\mathbf{h}_i, \mathbf{W}\mathbf{h}_j) \tag{6}$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))} \tag{7}$$

where $\mathbf{W}$ is a learned weight matrix, $\mathbf{h}_i$ are node features, and $\mathcal{N}_i$ denotes the neighborhood of node i. The updated node representation is then computed as:

$$\mathbf{h}_i' = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{h}_j\right) \tag{8}$$

Multi-head attention further enriches the representation by enabling the model to attend to multiple aspects of the topology in parallel. This is particularly valuable in network optimization, where bottleneck links or critical paths can decisively influence resource assignment and overall system utility.

B. Deep Sets Encoder

A principal challenge in learning network admission and routing policies is that the set of outstanding tasks is inherently variable in size and unordered across both training and inference. Traditional neural architectures struggle to process such

variable-length input sets, often requiring inefficient padding or suboptimal aggregation methods. The Deep Sets Encoder addresses this through a permutation-invariant architecture that guarantees representations independent of task ordering.

The Deep Sets framework processes a set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ through two neural networks:

$$f(\mathcal{X}) = \rho\left(\sum_{i=1}^{n} \phi(x_i)\right) \tag{9}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^m$ transforms individual elements and $\rho : \mathbb{R}^m \to \mathbb{R}^k$ processes the aggregated representation. This formulation ensures permutation invariance while enabling reasoning about both individual task attributes and holistic set-level properties.

In implementation, tasks are first processed through the ϕ network, optionally masked for invalid entries, then summed to form a single feature vector per batch. The ρ network produces the final set-level embedding, which is fused with other network state representations to enable context-aware decision-making.

C. Adaptive Path GNN

Learning representations for candidate paths that capture both sequential link interactions and evolving resource constraints poses a significant challenge. Standard GNNs model overall network structure but struggle to encode variable-length paths as distinct objects with both local and global information. The Adaptive Path GNN module addresses this limitation by enabling context-rich, path-level embeddings adaptable to both basic and enhanced feature inputs.

For a path $p = (v_1, v_2, \ldots, v_\ell)$ with link features $\mathbf{e}_{(v_i, v_{i+1})}$, the module applies multi-head self-attention:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{10}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are query, key, and value matrices derived from the sequence of link features. Masking ensures robust handling of variable-length paths during batching.

The intuition is that path feasibility and optimality depend not just on individual link properties, but on the sequence of characteristics along the entire path—bottlenecks, utilization gradients, and network position. The attention mechanism dynamically highlights critical hops and aggregates information into compact, expressive path representations.

D. Statistical/Structural Path Features Encoder

To complement learned representations with interpretable indicators of path quality, the Statistical/Structural Path Features Encoder computes hand-crafted descriptors of each candidate path. These include normalized path length, average utilization after task admission, minimum remaining capacity, and maximum link congestion.

For a path p and task with rate r, key features include: normalized path length ($\ell_p/\ell_{\max}$), average utilization ($\frac{1}{|\mathcal{E}_p|}\sum_{e\in\mathcal{E}_p}\frac{u_e+r}{c_e}$) and minimum remaining capacity ($\min_{e\in\mathcal{E}_p}\frac{c_e-u_e-r}{c_{\max}}$), where $\mathcal{E}_p$ represents links in path p, u_e is current utilization, c_e is capacity, and $c_{\max}$ is maximum link capacity. These features are processed through a multi-layer perceptron and fused with learned path embeddings, providing the agent with direct access to well-understood routing heuristics while enabling nuanced, experience-driven decision-making.

E. State Representation, Action Space and Reward Function

The state representation $s_t \in \mathcal{S}$ is a comprehensive multi-component tuple: $s_t = (\mathbf{N}_t, \mathbf{E}_t, \mathbf{G}_t, \mathbf{T}_t, \mathbf{P}_t)$, where $\mathbf{N}_t \in \mathbb{R}^{n\times d_n}$ contains node features (source/destination indicators, degrees, task interactions), $\mathbf{E}_t \in \mathbb{R}^{n\times n\times d_e}$ represents edge features (utilization, remaining capacity, path membership), $\mathbf{G}_t \in \mathbb{R}^{d_g}$ captures global network state (progress, admission rates, congestion metrics), $\mathbf{T}_t \in \mathbb{R}^{m\times d_t}$ encodes the task set, and $\mathbf{P}_t \in \mathbb{R}^{k\times d_p}$ contains path features for candidate routes.

The action space $\mathcal{A} = \{0, 1, 2, \ldots, k\}$ includes task rejection (action 0) and selection of one of k candidate paths (actions 1 to k). Action masking enforces feasibility constraints by restricting available actions to those satisfying current capacity constraints: $\mathcal{A}_t = \{a \in \mathcal{A} : \text{feasible}(a, s_t)\}$.

The reward function balances multiple objectives through weighted components: $r_t = \alpha_u R_u + \alpha_e R_e + \alpha_c R_c + \alpha_f R_f + \alpha_r R_r$, where R_u represents utility gain, R_e captures efficiency benefits, R_c penalizes congestion, R_f accounts for future opportunity costs, and R_r penalizes rejections. Time-dependent weights encourage exploration early in episodes and exploitation toward episode completion, facilitating robust policy learning that maximizes long-term cumulative utility rather than myopic gains.

IV. System Evaluation

To evaluate the performance of the DQN agent, we constructed a controlled experimental framework for network routing and admission control. The network topology is generated randomly for a fixed number of nodes, with link connectivity determined by a specific link density, while guaranteeing that the generated network remains fully connected. Each directed link is assigned a random data capacity drawn uniformly from $[C_{\min}, C_{\max}]$. For each episode, a fixed number of tasks is produced, with every task assigned a random source and destination node, a data rate from $[R_{\min}, R_{\max}]$, and a utility value from $[U_{\min}, U_{\max}]$, both sampled from uniform distributions.

Given the combinatorial nature of routing in graphs, enumerating all acyclic source-destination paths for each task quickly becomes computationally prohibitive as the network size and number of tasks increase. To address this, we precalculate all acyclic routes per task and select only K candidate paths (either the K shortest or K random ones) for both training and testing. This path subset restriction is essential to reducing the search space, ensuring tractable computations for the DQN agent and reference algorithms, while preserving a representative diversity of routing options sufficient for meaningful learning and performance comparison.

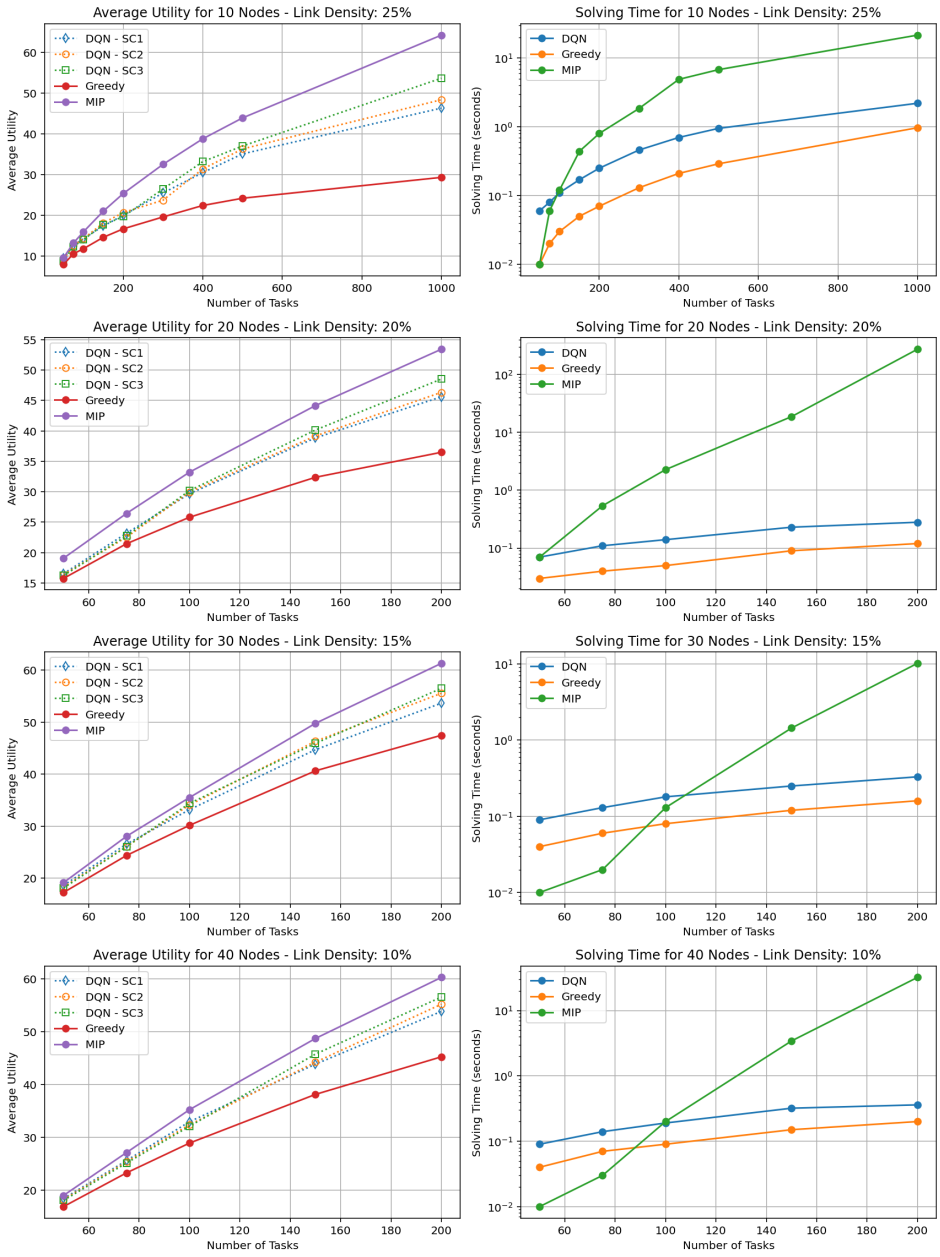

Fig. 1. Comparison of average utility (left) and solving time (right) achieved by the DQN agent, Greedy allocation, and MIP optimizer across varying network sizes and link densities. The results are shown for different numbers of tasks and three DQN evaluation scenarios.

Three scenarios are designed to systematically probe the DQN agent's ability to generalize under varying degrees of network variability:

- **Scenario 1**: The network topology and link capacities are fixed throughout training and evaluation. A different set of random tasks is used in each episode, measuring the agent's robustness to changing traffic demand on a static infrastructure.
- **Scenario 2**: The network topology and connectivity are fixed, but the link capacities and sets of tasks are re-generated for each episode. This emphasizes the DQN's adaptation to variable resource constraints.
- **Scenario 3**: Only the number of network nodes remains constant; both the network connectivity (link set), link capacities, and task assignments vary in every episode. This is the most general scenario, testing the agent's adaptability to entirely new topologies and network states.

For benchmarking, the DQN agent's results are compared to two established reference solvers (for strict fairness, each algorithm is evaluated on the exact same network topologies, link capacities, and sets of tasks):

- **Greedy agent**: Admission and routing decisions are made

by sequentially admitting the first feasible task-path that maximizes immediate utility, without consideration of long-term resource consequences. It represents an efficient yet myopic allocation strategy that is commonly used in online settings due to its speed and simplicity.

- **MIP optimizer**: The Mixed Integer Programming (MIP) optimizer formulates the problem described in Section II as an exact combinatorial optimization problem over the same set of candidate paths and under identical constraints. While it provides an upper bound for utility, the MIP is computationally expensive and typically infeasible for large-scale or real-time scenarios.

The evaluation results (Fig 1), comprehensively compare average utility and computational solving time for increasing numbers of tasks, across network sizes and scenarios. The evaluation takes place over 50 sets of randomly generated tasks, and the average values are presented. The results demonstrate that the DQN agent consistently surpasses the Greedy baseline in admitted utility across all tested network topologies. This improvement becomes more pronounced as the number of tasks increases, with the DQN outperforming Greedy by approximately 18-20% at lower task loads (50 tasks) and reaching up to 58% improvement for large task sets of 1000 tasks in 10-node networks. These gains are achieved despite training the DQN agent on significantly smaller task sets (50 tasks), highlighting its strong generalization capability in dynamic and complex environments. For example, in the 20-node network with 200 tasks, DQN achieves utility values ranging from 38.1-48.5 across scenarios compared to Greedy's 36.48, while the MIP optimizer reaches 53.42. This pattern holds steady across all topologies, from 10-node networks with 25% link density to 40-node networks with 10% link density, reflecting the robustness of the learned policies.

Interestingly, as the number of tasks grows during evaluation, models trained under Scenario 3 (dynamic link changes) demonstrate superior performance compared to those trained in more static settings (Scenario 1). Scenario 2 (capacity changes) follows closely behind, with both outperforming Scenario 1. For instance, in the 10-node network with 1000 tasks, Scenario 3 achieves a utility of 53.65 compared to Scenario 1's 46.36, representing an 83% improvement over Greedy versus 58% for Scenario 1. This occurs because agents trained in more variable environments learn generalized admission and routing policies that rely less on memorizing static topology characteristics. Instead, they develop adaptive strategies focused on underlying control principles, enabling better flexibility and transfer to unseen conditions. Regarding computational efficiency, all three scenarios share similar evaluation durations since only the training data differs, with DQN inference remaining consistently under 0.4 seconds across all cases. The simpler network topologies, such as the 10-node case with 25% link density, allow for evaluation up to 1000 tasks efficiently. Conversely, for larger topologies, the MIP solver's computation time grows exponentially, with solving times reaching 272.90 seconds for 20 nodes with 200 tasks and 32.07 seconds for 40 nodes with 200 tasks. For task sets exceeding 200 tasks in larger topologies, the MIP solver's computation time exceeds 15 minutes per evaluation, rendering detailed timing analysis impractical while the DQN agent maintains sub-second inference time, delivering near-optimal utility with vastly reduced computational overhead.

V. Conclusions

We presented a comprehensive GNN-based Deep Reinforcement Learning framework for task admission and routing optimization in dynamic IoBT environments, successfully combining GATs, Deep Sets encoders, Adaptive Path GNNs, and statistical feature processing to address the critical need for fast, adaptive decision-making in adversarial battlespace conditions. The experimental results demonstrate that the DQN agent consistently outperforms greedy baselines by 5-58% across all tested scenarios while maintaining sub-second inference times, exhibits strong generalization capabilities by scaling from 50-task training sets to 1000-task evaluation scenarios, and shows superior performance when trained under dynamic conditions rather than static environments, suggesting that exposure to network variability produces more robust policies. The approach effectively balances the speed-versus-optimality trade-off by achieving 70-90% of optimal utility within tight time constraints demanded by battlespace applications, while MIP solvers require impractical computation times. Future research directions include multi-objective optimization, adversarial training, distributed implementations, and uncertainty quantification techniques, with this GNN-DRL approach establishing a solid foundation for next-generation adaptive networking solutions in adversarial IoBT systems where rapid adaptation over perfect optimization is essential for mission success.

References

[1] A. Gkelias, P.J. Baker, K.K. Leung, O. Worthington, and C. Melville, "Digital Twins for Internet of Battlespace Things (IoBT) Coalitions," in International Conference on Military Communication and Information Systems (ICMCIS), 2025.

[2] Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Veličković, "Combinatorial Optimization and Reasoning with Graph Neural Networks," in Journal of Machine Learning Research, vol. 24, no. 130, pp. 1–61, 2023.

[3] T. Perera, S. Atapattu, Y. Fang, P. Dharmawansa and J. Evans, "FlexNet: A Graph Neural Network Approach to Resource Management in Flexible Duplex Networks," in 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 2023.

[4] A. Asheralieva, D. Niyato and Y. Miyanaga, "Efficient Dynamic Distributed Resource Slicing in 6G Multi-Access Edge Computing Networks With Online ADMM and Message Passing Graph Neural Networks," in IEEE Transactions on Mobile Computing, vol. 23, no. 4, pp. 2614-2638, April 2024.

[5] R. Wang et al., "Learning to Branch in Combinatorial Optimization with Graph Pointer Networks," in IEEE/CAA Journal of Automatica Sinica, vol. 11, no. 1, pp. 157-169, January 2024.

[6] C. -L. Liu and T. -H. Huang, "Dynamic Job-Shop Scheduling Problems Using Graph Neural Network and Deep Reinforcement Learning," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 11, pp. 6836-6848, Nov. 2023.

[7] Y. Zhang, J. Hare, A. Prügel-Bennett "Deep Set Prediction Networks," in Advances in Neural Information Processing Systems 32 (NeurIPS 2019), pp. 3212-3222, 2019.