International Cartographic Association
Association Cartographique Internationale

# GAGE-Q: Reinforced Genetic Algorithm using Spatial Neighborhood Graph Embedding for Green Intermodal Transportation

Hadi Aghazadeh [a,+], Reza Safarzadeh [a,+] , Xin Wang [a*]

[a] *University of Calgary, 622 Collegiate Place NW, Calgary, Alberta, Canada*
*hadi.aghazadeh@ucalgary.ca, reza.safarzadeh@ucalgary.ca, xcwang@ucalgary.ca*
[+] *Hadi Aghazadeh and Reza Safarzadeh contributed equally to this work and share the first authorship.*
* Corresponding Author

**Abstract:**
Intermodal transportation, using multiple modes in a single journey, promotes sustainable logistics. This paper addresses the Intermodal Vehicle Routing Problem and proposes *GAGE-Q*, which integrates graph embedding and Reinforcement Learning into a Genetic Algorithm. Our method models cities and their multi-modal connections as a graph, leveraging embedding for spatial dependencies and RL-based crossover for faster convergence and better solutions. Experiments on synthetic and real-world data show that GAGE-Q outperforms state-of-the-art methods, offering improved solution quality and efficiency in intermodal route planning.

**Keywords:** Intermodal Transportation, Vehicle Routing Problem , Reinforcement Learning , Graph Embedding , Genetic Algorithm

## 1. Introduction

Intermodal transportation employs multiple modes (trains, trucks, planes) in a single journey, often increasing sustainability by reducing energy use, emissions, and congestion Liu et al. (2022). However, efficiently planning and integrating these modes while minimizing travel costs and energy consumption remains a significant challenge.

Intermodal transportation route planning extends the Vehicle Routing Problem (VRP) by choosing the best mode of travel (e.g., highways, railways, air) for each trip segment. The network is modeled as a graph with nodes as cities and edges as mode-specific travel costs. The goal is to minimize total cost while meeting supply and demand constraints. For instance, in Figure 1, flying first from start point to node 5 and then taking a train to the end point costs 25, whereas using trucks from start point to node 3 then node 5, and then using a train to the end point might cost 24, proving more efficient.

Intermodal transportation differs from VRP by allowing multiple edges between nodes, adding complexity. Strategies to address intermodal VRP include exact algorithms, metaheuristics, and machine learning models Göçmen and Erol (2019). Exact algorithms work for small instances but struggle with larger, complex problems. Metaheuristics Yang et al. (2023)Mohammed et al. (2017) excel at scale, while machine learning, especially Reinforcement Learning (RL) Qin and Sun (2022)Aghazadeh and Wang (2024), offers near real-time solutions but requires extensive training and lacks metaheuristic precision.

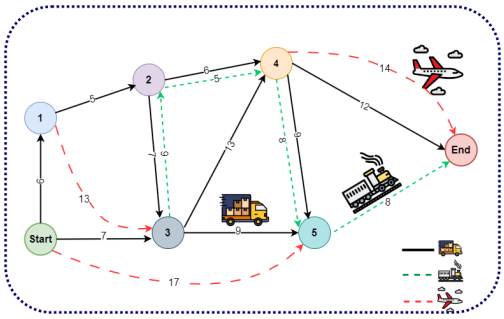


Figure 1. Overview of Intermodal Transportation.

Recent research on intermodal VRP often uses Genetic Algorithms (GAs) Fazayeli et al. (2018), favored for their simplicity and global solution exploration Okyere et al. (2022). However, GAs can struggle with slow run times and local optima Zhu et al. (2022), especially in the complex solution space of intermodal VRP.

To address GA's challenges in intermodal VRP, solutions include customizing GA for specific problems Sun et al. (2017) or adding extensions to accelerate convergence Fazayeli et al. (2018). While these improve results, they lack adaptability to intermodal VRP's complexities. Crossover, a vital GA component, navigates the search space but varies in effectiveness across problems Kang et al. (2018). Conventional operators like single-point or multi-point crossovers rely on fixed probabilities, often ignoring individual problem features, causing convergence issues Sun et al. (2017).

To enhance crossover in GAs, we propose integrating Reinforcement Learning (RL) to replace random gene selec-

tion with smarter, data-driven decision-making. RL learns from previous crossovers to identify weak chromosome segments and optimize exchanges, guiding the algorithm toward better solutions. To our knowledge, guided crossover using RL in GAs for intermodal VRP remains unexplored.

Another key limitation in traditional GAs is the neglect of local spatial dependencies within network graphs, which can lead to suboptimal solutions or delayed convergence, especially in complex intermodal VRPs with multi-layered connectivity. Utilizing graph embedding addresses this by capturing hidden mode-aware spatial relationships and providing a similarity metric, which we incorporate as an RL reward signal.

We propose **GAGE-Q** (Genetic Algorithm with Graph Embedding and Deep-Q-Networks), combining graph embedding and RL to improve GA performance. Graph embedding captures spatial neighborhood similarities as RL rewards, while an advanced RL model guides adaptive crossover, balancing exploration and exploitation for faster convergence and superior solutions.

- We employed a spatial neighborhood-based graph neural network embedding model to learn similarity scores for different transportation modes within the proposed network graph of the problem as a new approach for reward signal in RL.
- By utilizing the similarity score obtained from the graph embedding model as a novel reward signal, we trained an RL model to identify optimal gene change points within the crossover function in the GA. Integrating RL enables the acceleration of convergence and facilitates the discovery of superior solutions due to the guided and adaptive nature of crossover based on the problem's graph structure.
- Extensive experiments have been conducted on synthetic and real-world datasets, demonstrating GAGE-Q's superior efficiency and effectiveness compared to state-of-the-art algorithms.

Additionally, to calculate travel costs, we go beyond standard distances by factoring in transportation modes and greenhouse gas emissions, aligning with green intermodal transportation goals.

The paper is organized as follows: Section 2 reviews prior studies and intermodal VRP challenges. Section 3 details the proposed GAGE-Q. Section 4 presents the experimental setup and comparisons with baselines. Section 5 concludes the study and suggests future research directions.

## 2. Related Works

Genetic Algorithms (GAs) are widely used in transportation and logistics, addressing challenges like time windows, capacity constraints, and environmental concerns, especially in green transportation Konstantakopoulos et al. (2022), Sherif et al. (2021), Ren et al. (2020), Moghdani et al. (2021). Multimodal transport, a focus in intermodal logistics, is more sustainable than unimodal road transport, offering cost savings and reduced environmental impact Utama et al. (2020), Okyere et al. (2022).

GAs play a key role in optimizing intermodal logistics, tackling objectives such as distance, time, and emissions. Adaptive GAs with dynamic crossover and mutation probabilities have shown promise in reducing greenhouse gas emissions Yang et al. (2023). Similarly, GA models addressing $CO_2$ emissions, time, and distance Okyere et al. (2022) have been proposed. Other studies use GAs for complex logistics networks, integrating green transportation with inventory challenges or incorporating fuzzy demands and time windows Sherif et al. (2021), Fazayeli et al. (2018).

Beyond green logistics, GAs have been applied to multimodal transportation problems, optimizing routing and mode selection. Adaptive GAs enhance global search performance Sun et al. (2017), while hybrid GAs address specific constraints like multimodal time windows Fazayeli et al. (2018). However, challenges such as slow convergence and local optima persist.

Emerging methods, including Reinforcement Learning, show promise for intermodal VRPs, offering strong generalization capabilities Adi et al. (2020), Zhang et al. (2024), Song et al. (2023). Yet, RL demands extensive data and fine-tuning, and its accuracy remains below metaheuristic methods like GAs or Mixed-Integer Linear Programming.

Hybrid models combining GAs with other methods address these limitations. For example, combining GAs with dynamic programming improves stability in drug design Fu et al. (2022), while Q-learning-based GAs have achieved superior results in solving NP-hard problems like the Traveling Salesman Problem (TSP) Zheng et al. (2023). Applications of graph-enhanced GAs, such as minimizing neural network computation costs Paliwal et al. (2019), highlight the potential of leveraging graph structures for optimization.

Despite progress, integrating adaptive mechanisms into GAs to address intermodal VRPs remains underexplored. Current methods fail to fully utilize spatial dependencies in intermodal transportation networks, limiting their scalability and precision.

To address this gap, proposed GAGE-Q method combines graph embedding and RL within the GA framework. By capturing spatial neighborhood similarities as RL rewards, GAGE-Q enhances the crossover process, enabling faster convergence and superior solutions. This novel integration dynamically adapts to complex intermodal VRPs, overcoming traditional GA limitations and achieving high computational efficiency and solution quality.

## 3. Methods

In this section, we will overview of the GAGE-Q, and a detailed explanation of the technical methods employed in this study.

### 3.1 Overview of GAGE-Q

As shown in Figure 2, the GAGE-Q algorithm for the intermodal VRP starts with the initialization of a population comprising routes with distinct sequences of locations and transportation modes. Subsequently, the fitness of each solution is evaluated based on criteria such as total travel distance, time, and greenhouse gas emissions. Employing elitism, the algorithm selects the best-performing individuals to propagate to the next generation unaltered, while parent selection is conducted to determine individuals for crossover.

During crossover, instead of traditional methods, RL is utilized to identify weaknesses in each chromosome's route sequence and transportation mode selection. The transportation network is embedded in a three-layer graph, taking into account transportation modes, to train the RL agent. Resulting similarity matrices from this embedding act as rewards for the RL agent, guiding it in identifying the weakest points in the routes. Guided by the RL agent's suggestions, new chromosomes are generated, replacing old ones to produce offspring solutions with enhanced characteristics. This iterative process continues for multiple generations until a stopping criterion (maximum number of iterations) is met.

### 3.2 Components of GAGE-Q

In this section, we will explain the key components of GAGE-Q and provide insights into how each element, including reinforcement learning and graph embedding, contributes to enhancing the performance of the algorithm.

#### *3.2.1 Chromosome Encoding*

Designing an effective chromosome representation is crucial for the proposed Genetic Algorithm (GA) model, as all subsequent tasks, including the RL model, depend on it.

The proposed representation comprises two segments: city sequences and transportation modes. The first segment prioritizes city traversal order, while the second specifies the transportation mode between consecutive cities. Chromosome lengths remain fixed by padding unused cities or routes with zeros. Figure 3 illustrates this structure.

In the example, six cities are available. The first six genes represent visited cities, and the next five indicate transportation modes. Unused cities (e.g., cities 1 and 3) and modes are set to zero. Figure 3 shows a route starting at the initial node, moving to city 4 by train (mode 2), then to city 2 by airplane (mode 3), and finally to the destination. This fixed-length representation ensures consistency, simplifies the GA process, and includes a penalty for infeasible routes.

#### *3.2.2 Fitness Function*

The fitness function is crucial in optimization, assigning scores to chromosomes based on performance. For this intermodal VRP, it balances minimizing total GHG emissions and travel costs, accounting for mode-specific costs and emission rates.

Let $F$ denote the fitness function as the sum of GHG emissions and travel costs across all routes. For each route $\tau$, the GHG emissions $E_\tau$ and travel costs $C_\tau$ can be calculated as follows:

$$\text{GHG Emissions:} \quad E_\tau = \sum_i \sum_j \sum_{\text{mode}} \text{dist}_{ij} \times \text{GHG rate}_{ij}^{\text{mode}} \tag{1}$$

$$\text{Travel Costs:} \quad C_\tau = \sum_i \sum_j \sum_{\text{mode}} \text{time}_{ij} \times \text{travel cost}_{ij}^{\text{mode}} \tag{2}$$

where $\text{dist}_{ij}$ is the distance traveled on route $\tau$ between nodes $i$ and $j$, $\text{GHG rate}_{ij}^{\text{mode}}$ is the GHG emission rate per kilometer for the transportation mode between nodes $i$ and $j$, $\text{time}_{ij}$ is the time taken to travel between nodes $i$ and $j$, and $\text{travel cost}_{ij}^{\text{mode}}$ is the travel cost per hour for the transportation mode between nodes $i$ and $j$.

To prevent infeasible solutions in GA, we added a penalty term $P_{\text{inf}}$ to the fitness function. This large positive value penalizes routes $\tau$ that violate constraints, reducing their selection probability. Thus, the fitness function $F$ is defined as a minimization problem:

$$\text{Minimize} \quad F = \sum_k (w_1 E_k + w_2 C_k) + P_{inf} \tag{3}$$

where $w_1$ and $w_2$ are the weights assigned to GHG emissions and travel costs, respectively, for normalization purposes to ensure that their impact on the overall fitness function is balanced, preventing bias toward larger values. These weights were determined through trial and error.

**Algorithm 1** Adaptive Crossover with Deep Q-Networks

**Input:** Selected Chromosomes $P$, Graph Embedding $\mathscr{G}$, Parameters for DQN
**Output:** New Chromosomes $P'$
**for** each pair of parent chromosomes $p_i, p_j \in P$ **do**
  Initialize state $s$ with chromosome pairs $(\vec{p}_i, \vec{p}_j)$
  Initialize $P_{new} \leftarrow \{\}$
  **while** not converged **do**
    Select action $a$ from set of all actions $A$
    Apply action $a$ to determine gene change point
    Generate new chromosome pair $(p'_i, p'_j)$ based on action $a$
    Compute reward $r$ using graph embedding similarity matrices
    Update DQN with $(s, a, r, s')$
    Set $s \leftarrow s'$
    Add $(p'_i, p'_j)$ to $P_{new}$
  **end while**
**end for**
$P' \leftarrow P_{new}$

#### *3.2.3 Crossover Operation*

The crossover operation in GAGE-Q leverages RL to optimize gene selection from parent chromosomes, producing offspring with desirable traits. The RL model learns from previous crossovers to make informed decisions, reducing reliance on randomness. Algorithm 1 outlines the steps, with further details in Section 3.3.

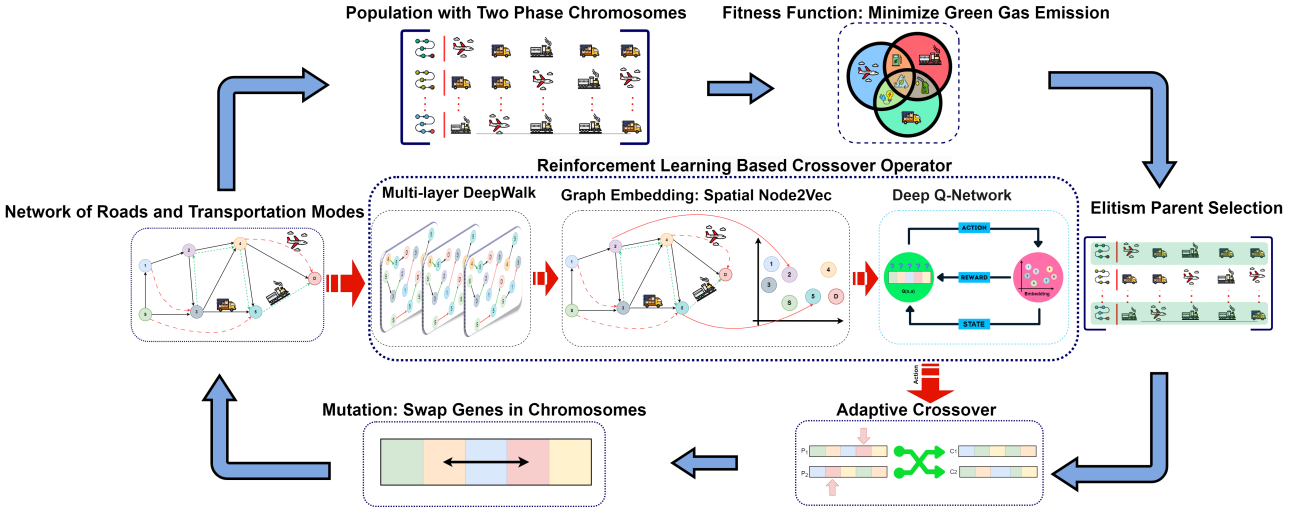


Figure 2. Overview of Proposed GAGE-Q algorithm.

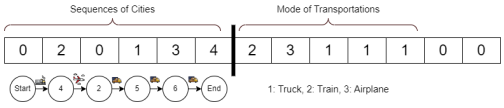


Figure 3. Chromosome representation example.

### 3.2.4 *Mutation Operation*

Mutation introduces diversity by randomly altering genes in the chromosome, affecting route sequences and transportation modes. Based on a predefined mutation rate, selected genes are modified to generate new paths, replacing old ones and exploring novel solutions in the population.

## 3.3 Adaptive Crossover based on Reinforcement Learning and Graph Embedding

As shown in Figure 2, the adaptive crossover operator utilizes an RL model with a reward signal derived from graph embeddings. A random walk on the graph captures pathways of varying lengths, processed by Node2Vec to compute a similarity metric. This metric, sensitive to node relationships, guides the RL model to optimize the chromosome sequence by placing related nodes closer together. The next section details the RL and graph embedding components.

### 3.3.1 *Deep Q-Networks for Crossover Operation*

The adaptive crossover leverages an RL model to determine the optimal gene change point, formulated as a Markov Decision Process (MDP) and solved using Deep Q-networks (DQN).

DQN, an RL algorithm combining deep learning and Q-learning, approximates the optimal action-value function $Q^*(s,a)$. Here, $s$ represents the state (selected parent chromosomes), and $a$ is the action (gene change point). The DQN, parameterized by weights $\theta$, takes the state $s$ as input and outputs Q-values for all possible gene change points. Actions are selected using an $\varepsilon$-greedy strategy to balance exploration and exploitation. The network minimizes the Temporal Difference (TD) error between target and estimated Q-values using the following loss function:

$$L(\theta) = \mathbb{E}_{s,a,r,s' \sim D}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta)\right)^2\right] \tag{4}$$

where $r$ is the immediate reward, $s'$ is the next state, $\gamma$ is the discount factor, $\theta^-$ represents the target network parameters, and $D$ is the replay buffer storing experiences $(s_t, a_t, r_t, s_{t+1})$. Through gradient descent, the weights $\theta$ are iteratively updated, improving the Q-value approximation and enabling effective gene change point selection.

DQN is particularly well-suited for this discrete action space and integrates seamlessly with the MDP-based approach. The components of the MDP are defined as follows:

## 3.4 State ($S$)

The state is defined as the pair of selected parent chromosomes $\vec{p}_i$ and $\vec{p}_j$ from the population $P$:

$$S = \{\vec{p}_i, \vec{p}_j\}, \quad i,j \in P \tag{5}$$

This state space provides sufficient information for the agent to identify the optimal gene change point.

## 3.5 Action ($A$)

The action space $A$ corresponds to all gene positions in the chromosomes:

$$A = \{g_k^i : g_k^i \in \vec{p}_i, i \in P, k \in \mathcal{G}\} \tag{6}$$

The agent's task is to select the gene change point that maximizes the long-term reward.

The next section introduces a novel reward signal design to guide the agent's decision-making process effectively.

### 3.5.1 *Graph embedding based Reward*

A key element in the DQN formulation is the reward, designed to capture spatial relationships in the graph, considering transportation modes. Using node2vec Grover and Leskovec (2016), embeddings are learned via random walks,

preserving graph topology and neighborhood structures. For multi-modal transportation graphs, subgraphs are created for each mode (e.g., road, rail, air), weighted by travel costs. Node2vec is then applied to each subgraph to generate embeddings.

a *Random Walks*: We perform random walks on each mode-specific subgraph to generate sequences of nodes. These random walks capture the spatial structural properties and relational context within each transportation mode.

b *Transition Probabilities*: During random walks, we determine transition probabilities based on a combination of breadth-first search (BFS) and depth-first search (DFS) strategies. This ensures a balance between exploring local neighborhoods and global topology within each transportation mode.

c *Embedding Learning*: We use the continuous bag-of-words (CBOW) model to learn embeddings from the observed node sequences generated during random walks to predict the spatial neighborhood of a given node based on its embedding representation, capturing the relational information encoded in the mode-specific subgraph.

Let $V$ denote the set of nodes in the transportation graph, and $d$ represent the dimensionality of the node embeddings. For each mode-specific subgraph $G_{mode}$, where $mode \in \{road, rail, air\}$ represents the transportation modes (highway, railway, airway), we learn node embeddings using node2vec:

$$\text{Emb}_{mode} = \text{node2vec}(G_{mode}, d) \tag{7}$$

Here, $\text{Emb}_{mode}$ represents the node embedding for transportation mode $mode \in \{road, rail, air\}$, obtained using the node2vec algorithm with $d$ dimensions. In this way, we obtain embedding for nodes corresponding to each transportation mode. Therefore we have have three-layered similarity matrix between each pair of nodes for different modes. After obtaining mode-specific node embedding, we can compute the similarity between nodes based on their embedding using cosine similarity. For two nodes $v_1$ and $v_2$ with embedding $e_1^{mode}$ and $e_2^{mode}$ , the cosine similarity is computed as:

$$\text{cosine\_similarity}^{mode}_{(v_1,v_2)} = \frac{e_1^{mode} \cdot e_2^{mode}}{\|e_1^{mode}\| \times \|e_2^{mode}\|} \tag{8}$$

This similarity matrix captures the pairwise spatial similarities between nodes in the transportation network within each transportation mode and serves as the reward for the RL agent in which the reward signal guides the agent toward its goal, allowing it to learn and maximize long-term gains. For the adaptive crossover, the goal is to place similar nodes in close proximity within the chromosome structure. For instance, if nodes 1 and 2 are close together in the graph, positioning them similarly in the chromosome encourages the RL model to maintain this proximity in the correct sequence.

## 4. Experiments

We evaluate GAGE-Q against traditional methods (TS Brandão (2011), SA Afifi et al. (2013), GA Tasan and Gen (2012)), Q-learning Aghazadeh and Wang (2024), and two GAGE-Q variants: Naive GAGE (no RL) and Naive GA-Q (no graph embedding, using distance as a reward). Experiments assess GAGE-Q's efficiency and effectiveness across varying scenarios, including different node sizes and complexities. An ablation study examines Fitness Function Curves, Model Robustness, and the impact of Spatial Graph Embedding and RL.

We will utilize two types of data: synthetic data of varying sizes to test the proposed model under different scenarios, and real-world case studies spanning major cities across Canada. The synthetic data encompasses three scenarios with city counts of 15, 30, and 60. As for the real-world data, all travel costs, greenhouse gas emissions, and total distances are derived from actual observations, which will be elaborated on in Section 4.3.

Model parameters include DQN parameters such as the learning rate of 0.1, the discount factor of 0.9, and the exploration rate of 0.2, with a three-layer architecture of 64, 32, and 16 nodes. For all GA models, there is a population size of 500, a mutation probability of 0.15, and a crossover probability of 1. Baseline models employ a Tabu Tenure of 10 in Tabu Search, an initial temperature of 10 decreasing by 0.9 in Simulated Annealing, and 500 episodes for all models. Additionally, a penalty of 100 is assigned for route infeasibility.

F

### 4.1 Computational Time Performance

This section presents the results of comparing the performance of the proposed GAGE-Q algorithm with four other algorithms across three different scenarios involving 15, 30, and 60 nodes. The results are summarized in Table 2, which shows the time each algorithm takes to reach its best solution and the corresponding fitness scores across scenarios involving 15, 30, and 60 nodes.

For the 15-node scenario, GAGE-Q achieves the best fitness score (10) in 45 seconds, outperforming TS, SA, GA, and Q-Learning in both quality and speed. In the 30-node scenario, GAGE-Q delivers a fitness score of 24 in 58 seconds, while other methods take longer and yield less optimal solutions. For 60 nodes, GAGE-Q maintains its lead with a fitness score of 89 in 95 seconds, outperforming all baselines. These results highlight GAGE-Q's superior efficiency and solution quality across varying problem sizes.

Overall, GAGE-Q outperforms TS, SA, GA, and Q-Learning with superior fitness scores and faster convergence, driven by its RL-enhanced crossover process.

### 4.2 Ablation Study

We conduct an ablation study to assess the impact of spatial graph embedding and RL in GAGE-Q by comparing it to Naive GAGE (without RL) and Naive GA-Q (without graph embedding).

Table 1. Comparison of Running Time and Best Solution Fitness Scores for Different Algorithms across Various Scenarios (15, 30, and 60 Nodes)

| | Parameter | 15 Nodes | 30 Nodes | 60 Nodes |
|---|---|---|---|---|
| **TS** | Fitness Score | 69 | 239 | 1019 |
| | Time | 47 Sec | 69 Sec | 95 Sec |
| **SA** | Fitness Score | 78 | 195 | 1065 |
| | Time | 53 Sec | 114 Sec | 182 Sec |
| **GA** | Fitness Score | 29S | 110 | 850 |
| | Time | 66 Sec | 160 Sec | 276 Sec |
| **Q-Learning** | Fitness Score | 21 | 67 | 417 |
| | Time | 175 Sec | 265 Sec | 461 Sec |
| **Naive GAGE** | Fitness Score | 17 | 42 | 136 |
| | Time | 90 Sec | 189 Sec | 355 Sec |
| **Naive GA-Q** | Fitness Score | 20 | 48 | 175 |
| | Time | 69 Sec | 83 Sec | 127 Sec |
| **GAGE-Q** | Fitness Score | 10 | 24 | 89 |
| | Time | 45 Sec | 58 Sec | 95 Sec |

#### *4.2.1 Fitness Function Convergence Curves*

We compare the convergence quality of GAGE-Q and baseline models across 15, 30, and 60-node scenarios using fitness scores from Section 3.2.2, as shown in Figure 4.

In the 15-node scenario, GAGE-Q achieved the best score (3.4), outperforming Naive-GAGE (12.4) and GA (26.8). For 30 nodes, GAGE-Q scored 8.5, surpassing Naive-GAGE (31) and GA (67). In the 60-node scenario, GAGE-Q again excelled with a score of 17, compared to Naive-GAGE (81) and GA (417), demonstrating the impact of RL-guided crossover on performance.

#### *4.2.2 Robustness Analysis of Models*

Achieving superior results requires both high accuracy and consistent performance across runs, with low variance indicating robustness and reliability. To evaluate this, each model was run 10 iterations per scenario, and deviations were calculated by normalizing each value against the minimum for that model. Figure 5 illustrates the findings.

As evident from the analysis, models incorporating adaptive crossover demonstrate greater robustness across multiple runs, exhibiting reduced susceptibility to variations in different iterations. Among these adaptive models, the proposed GAGE-Q model stands out for its superior robustness, consistently achieving comparable results across different runs.

#### *4.2.3 Influence of using spatial graph embedding*

We analyze the impact of spatial graph embedding on GAGE-Q performance by comparing its full version (green line, Fig. 6) with a version using only distance as the reward signal (red line, Fig. 6).

The gap analysis depicted in Figure 6 illustrates the significant impact of utilizing spatial graph embedding as a reward signal compared to using distance alone as a reward signal. This influence becomes particularly evident when comparing results across different edge numbers within each scenario. In Scenario 1, the increase in the number of edges correlates with changes in the fitness function value. However, it is notable that, on average, there is a difference of 4.6 between models with and without graph embedding. This discrepancy increases to 7.94 for Scenario 2 and 10.0 for Scenario 3. Consequently, employing spatial graph embedding leads to a dramatic improvement in fitness scores.

#### *4.2.4 Influence of using Reinforcement Learning*

Table 2. Comparison of Running Time and Best Solution

| | Parameter | 15 Nodes | 30 Nodes | 60 Nodes |
|---|---|---|---|---|
| **GA** | Fitness Score | 29 | 110 | 850 |
| | Iteration | 255 | 265 | 280 |
| **Naive GA-Q** | Fitness Score | 20 | 48 | 175 |
| | Iteration | 310 | 317 | 350 |

This section investigates the influence of RL mechanisms in the optimization of intermodal transportation routes by comparing the results obtained from simple GA, Naive-GAGE, Naive GA-Q, and GAGE-Q. The comparison is conducted across scenarios involving 15, 30, and 60 nodes to assess the impact of RL on convergence behavior and solution quality.

The results demonstrate a clear advantage of incorporating RL mechanisms in Naive GA-Q over the GA in terms of convergence iterations and solution quality. In all scenarios, Naive GA-Q exhibits significantly faster convergence, with convergence iterations of 255, 265, and 280 for 15, 30, and 60 nodes, respectively, compared to simple GA's iterations of 317, 310, and 350 in the corresponding scenarios. Moreover, Naive GA-Q achieves superior solution quality, as evidenced by its lower best solution fitness scores of 20, 48, and 175 for 15, 30, and 60 nodes, respectively, compared to simple GA's scores of 29, 110, and 850. These results underscore the effectiveness of RL in guiding the optimization process towards more optimal solutions, leading to improved solution quality and efficiency. The ability of RL to learn from previous iterations and adaptively adjust crossover operations contributes to faster convergence and enhanced exploration of the solution space.

### 4.3 Real Case Study

In this section, we conduct an empirical case study focused on the Canadian context to validate the effectiveness of our proposed approach.

#### *4.3.1 Case Description*

We conducted a case study on optimizing intermodal transportation routes across Canada, focusing on 30 major cities as nodes in the network (Figure 7). The transportation network was constructed using data from Canada's National Highway System, National Railway Network (Canada Open Government Portal [1]), and OpenFlights [2], ensuring comprehensive connectivity via road, rail, and air.

[1] https://search.open.canada.ca/opendata
[2] https://openflights.org

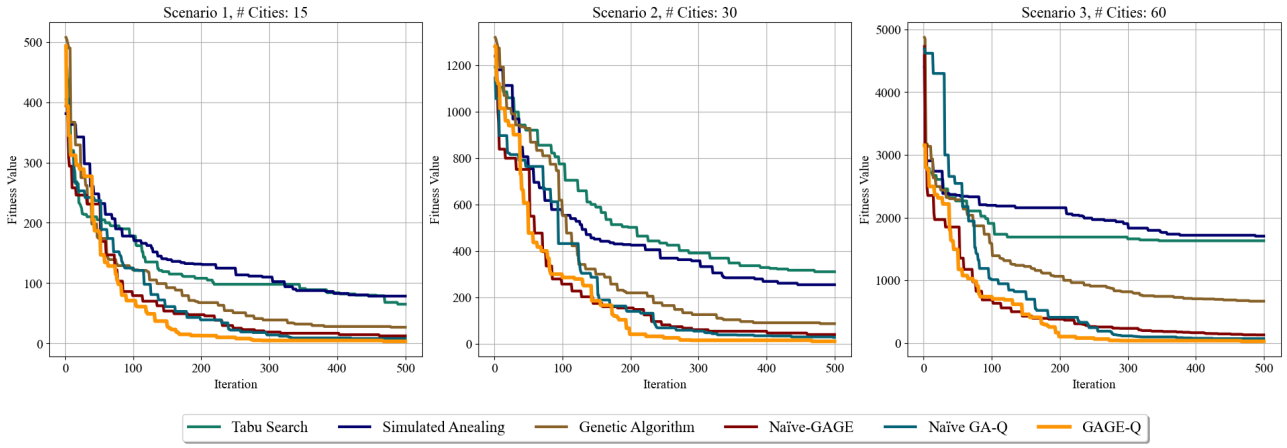


Figure 4. Comparison of GAGE-Q and Baseline Models.

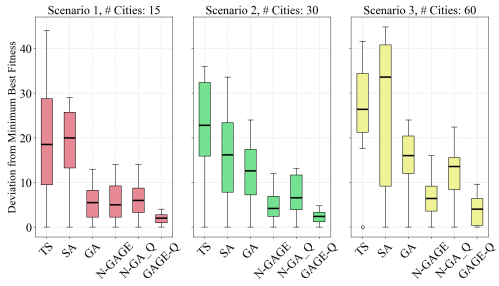


Figure 5. Robust analysis of models for three scenarios.

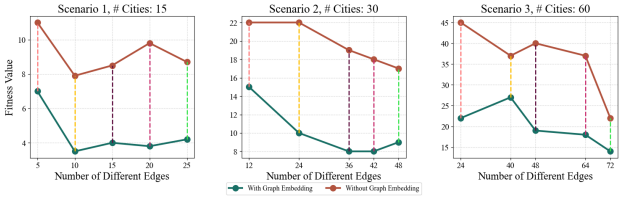


Figure 6. Influence of using Graph Embedding with different edge numbers.

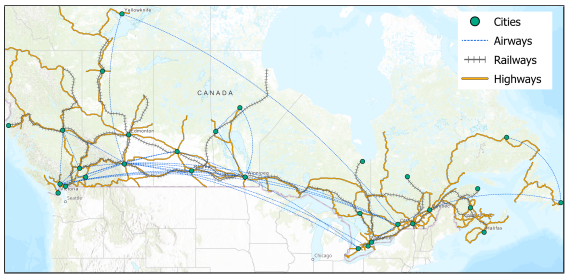


Figure 7. Road-rail-air intermodal transportation network in the real case study

### *4.3.2 Optimization results*

GAGE-Q optimized a green intermodal route from Mount Pearl to Victoria, minimizing costs and emissions while ensuring timely delivery (Fig. 8). It balanced air, rail, and road transport more effectively than Q-learning, demonstrating its eco-friendly and cost-efficient mode selection capabilities.

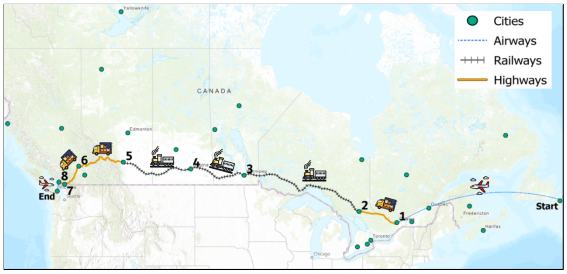


Figure 8. Optimized intermodal transportation route from Mount Pearl to Victoria with GAGE-Q

## 5. Conclusion

This study introduces GAGE-Q, a novel Reinforced Genetic Algorithm for optimizing green intermodal transportation routes. By integrating graph embedding and reinforcement learning into the genetic algorithm framework, GAGE-Q improves solution quality and computational efficiency while balancing economic and environmental goals. Our approach, which incorporates multiple transportation modes and greenhouse gas emissions, outperforms optimization methods like Tabu Search, Simulated Annealing, and standard Genetic Algorithms across various scenarios.

Future work could explore advanced techniques like graph neural networks and attention mechanisms to better capture spatial dependencies in transportation networks. Relaxing fixed cost assumptions and incorporating dynamic data sources, such as traffic patterns and weather, would enhance the model's realism and adaptability, paving the way for more robust and sustainable intermodal transportation solutions.

## References

Adi, T. N., Iskandar, Y. A. and Bae, H., 2020. Interterminal truck routing optimization using deep reinforcement learning. *Sensors* 20(20), pp. 5794.

Afifi, S., Dang, D.-C. and Moukrim, A., 2013. A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints.

In: *Learning and Intelligent Optimization: 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers 7*, Springer, pp. 259–265.

Aghazadeh, H. and Wang, X., 2024. Reinforcement learning for intermodal transportation planning with time windows and limited cargo capacity. In: *Proceedings of the 16th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, IWCTS '23, Association for Computing Machinery, New York, NY, USA, p. 28–31.

Brandão, J., 2011. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research* 38(1), pp. 140–151.

Fazayeli, S., Eydi, A. and Kamalabadi, I. N., 2018. Location-routing problem in multimodal transportation network with time windows and fuzzy demands: Presenting a two-part genetic algorithm. *Computers & Industrial Engineering* 119, pp. 233–246.

Fu, T., Gao, W., Coley, C. and Sun, J., 2022. Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems* 35, pp. 12325–12338.

Göçmen, E. and Erol, R., 2019. Transportation problems for intermodal networks: Mathematical models, exact and heuristic algorithms, and machine learning. *Expert Systems with Applications* 135, pp. 374–387.

Grover, A. and Leskovec, J., 2016. node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864.

Kang, J.-W., Park, H.-J., Ro, J.-S. and Jung, H.-K., 2018. A strategy-selecting hybrid optimization algorithm to overcome the problems of the no free lunch theorem. *IEEE Transactions on Magnetics* 54(3), pp. 1–4.

Konstantakopoulos, G. D., Gayialis, S. P. and Kechagias, E. P., 2022. Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Operational research* 22(3), pp. 2033–2062.

Liu, H., Zhan, P. and Zhou, M., 2022. Optimization of a logistics transportation network based on a genetic algorithm. *Mobile Information Systems*.

Moghdani, R., Salimifard, K., Demir, E. and Benyettou, A., 2021. The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production* 279, pp. 123691.

Mohammed, M. A., Abd Ghani, M. K., Hamed, R. I., Mostafa, S. A., Ahmad, M. S. and Ibrahim, D. A., 2017. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of computational science* 21, pp. 255–262.

Okyere, S., Yang, J. and Adams, C. A., 2022. Optimizing the sustainable multimodal freight transport and logistics system based on the genetic algorithm. *Sustainability* 14(18), pp. 11577.

Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P. and Vinyals, O., 2019. Reinforced genetic algorithm learning for optimizing computation graphs. *arXiv preprint arXiv:1905.02494*.

Qin, G. and Sun, J., 2022. Ride-hail to ride rail: Learning to balance supply and demand in ride-hailing services with intermodal mobility options. *Transportation Research Part C: Emerging Technologies* 144, pp. 103887.

Ren, Y., Wang, C., Li, B., Yu, C. and Zhang, S., 2020. A genetic algorithm for fuzzy random and low-carbon integrated forward/reverse logistics network design. *Neural Computing and Applications* 32, pp. 2005–2025.

Sherif, S. U., Asokan, P., Sasikumar, P., Mathiyazhagan, K. and Jerald, J., 2021. Integrated optimization of transportation, inventory and vehicle routing with simultaneous pickup and delivery in two-echelon green supply chain network. *Journal of Cleaner Production* 287, pp. 125434.

Song, Y., Wu, Y., Guo, Y., Yan, R., Suganthan, P. N., Zhang, Y., Pedrycz, W., Chen, Y., Das, S., Mallipeddi, R. et al., 2023. Reinforcement learning-assisted evolutionary algorithm: A survey and research opportunities. *arXiv preprint arXiv:2308.13420*.

Sun, Z., Sun, Z., Zhao, X., Jin, L. and Zhang, W., 2017. Application of adaptive genetic algorithm for multimodal transportation logistics distribution routing problem. In: *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing*, IEEE, pp. 75–80.

Tasan, A. S. and Gen, M., 2012. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering* 62(3), pp. 755–761.

Utama, D. M., Widodo, D. S., Ibrahim, M. F. and Dewi, S. K., 2020. A new hybrid butterfly optimization algorithm for green vehicle routing problem. *Journal of Advanced Transportation* 2020, pp. 1–14.

Yang, L., Zhang, C. and Wu, X., 2023. Multi-objective path optimization of highway-railway multimodal transport considering carbon emissions. *Applied Sciences* 13(8), pp. 4731.

Zhang, T., Cheng, J. and Zou, Y., 2024. Multimodal transportation routing optimization based on multi-objective q-learning under time uncertainty. *Complex & Intelligent Systems* pp. 1–20.

Zheng, J., Zhong, J., Chen, M. and He, K., 2023. A reinforced hybrid genetic algorithm for the traveling salesman problem. *Computers & Operations Research* 157, pp. 106249.

Zhu, J. et al., 2022. Solving capacitated vehicle routing problem by an improved genetic algorithm with fuzzy c-means clustering. *Scientific Programming*.