Graph attention reinforcement learning for 3D multi-modal automated on-demand delivery system

Farzan Moosavi[a,*], Bilal Farooq[a]

[a] Laboratory of Innovations in Transportation (LiTrans), Toronto Metropolitan University, Toronto, Canada
farzan.moosavi@torontomu.ca, bilal.farooq@torontomu.ca
* Corresponding author

Extended abstract submitted for presentation at the Conference in Emerging Technologies in Transportation Systems (TRC-30)
September 02-03, 2024, Crete, Greece

April 22, 2024

Keywords: On-demand multi-modal pickup and delivery, Airspace design, Dynamic Transformer, Deep reinforcement learning, Graph attention network

1 Introduction

This work introduces an on-demand multi-modal delivery framework with automated drones and sidewalk robots on a managed 3D cyber-physical road network for urban areas (Figure 1). In this context, a novel dynamic transform-based deep reinforcement learning approach is proposed for network planning and vehicle routing optimization, considering weather and traffic uncertainties and further infrastructure constraints like obstacle avoidance and vehicle operations such as battery usage and capacity. The goal is to create a flexible, scalable and managed airspace network adapted to city characteristics and balancing the "free flight" and "structured" concepts in airspace design (4).

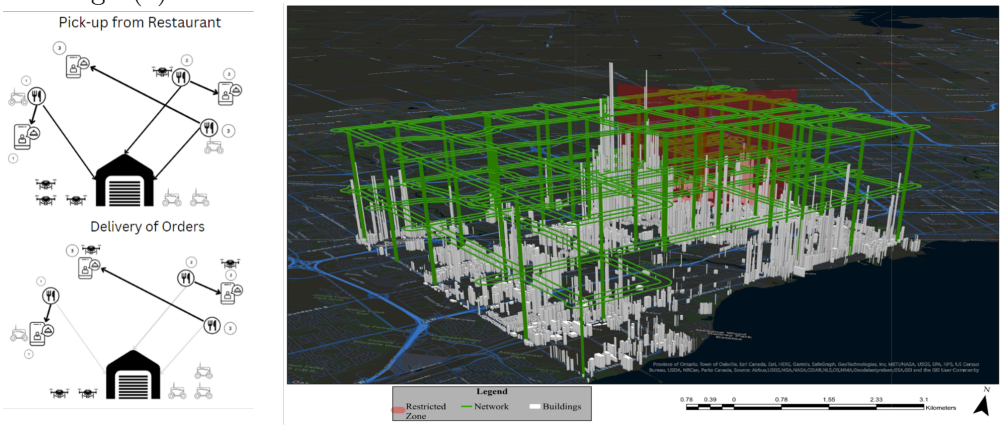

Figure 1 – *3D Road Network for Mississauga, Canada*

Regarding vehicle routing, this research considers the pick-up and delivery problem with time windows (PDPTW). Given the dynamic nature of on-demand delivery requests, assigning them to the best available vehicle in the fleet within a limited planning horizon presents computational challenges due to the time inefficient of traditional methods (7). Thus, this study integrates a Graph Attention Network (GAT) and Transformer architecture to address these challenges using an end-to-end learning approach. Deep reinforcement learning (DRL), augmented with a dynamic transformer model and a novel customized graph attention network, is utilized to solve

PDPTW and minimize the delivery time delay. The proposed approach coordinates a multi-modal fleet with a network on the ground and in the air, using a dual encoder architecture to capture the embedding of both modes, along with heterogeneous attention to account for precedence constraints (5) and, more importantly, a customer's spatial and temporal correlation to enrich node and graph context embedding for dynamic routing. A dynamic encoding mechanism is introduced to update edges in addition to node embedding, serving as a dynamic encoder that reflects the online routing problem and addresses on-demand delivery.

2 Methodology

The delivery comprises two unique network graphs of operation for a fleet of drones and robots based on the road network of an urban area. The nodes are represented by a directed graph for drone network $G^d = (X, E^d)$, where $P = \{x_1, \ldots, x_N\}$ denotes the set of N pickup nodes with depot, x_0, and $D = \{x_{N+1}, \ldots, x_{2N}\}$ as corresponding delivery nodes. In addition, $E^d = \{(i, j) \mid x_i, x_j \in X\}$ denotes the set of edges connecting the locations. The same representation is applied for the robot graph network $G^r = (X, E^r)$ with the same node but different edges. The coordinate of the ith location, weight of the order and time window is denoted by $\mathbf{u}_i$, q_i, and $[e_i, l_i]$ respectively with $q_i > 0$ and $q_{i+N} = -q_i$. Each vehicle must serve the pickup and delivery of requests together, accounting for precedence constraints within the time window of each point; otherwise, they get delayed, and the penalty is considered. There are N^d and N^r drones and robots correspondingly. The kth vehicle, $k \in \{1, \ldots, N^{d,r}\}$, has a capacity $Q_k^{d,r}$ and a battery size $B^{d,r}$. If the kth vehicle is used, it departs from the depot with a sequence of locations and returns to the depot after its final delivery or when it needs recharging. The drone and robot networks vary by wind and congestion, respectively, meaning their impedance updates at different times of the day. According to Figure 2, the method comprises graph attention encoding part moving to the cooperative decoding for handling both fleets. Before the encoder, the time and distance-based hop neighbourhood for edges are incorporated to find a spatial-temporal correlation for aerial and terrestrial network embedding.

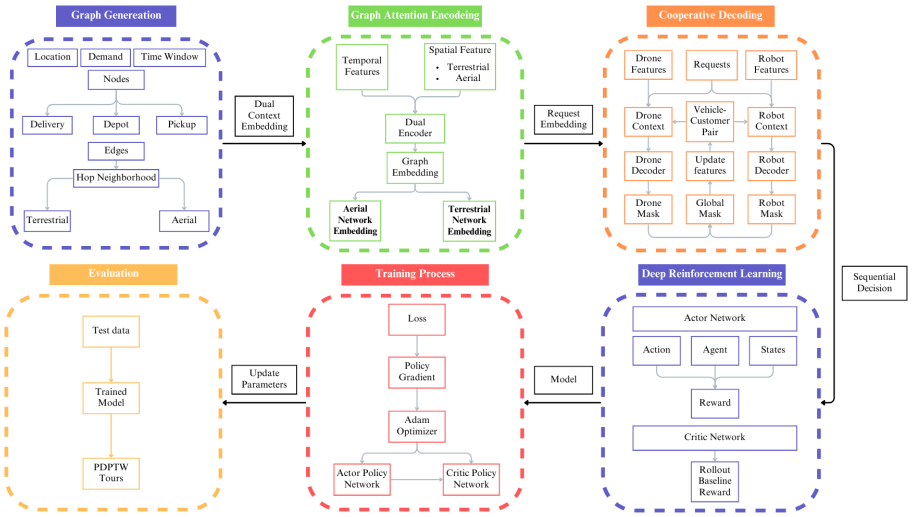

Figure 2 – *Overview of the methodology*

2.1 DRL formulation. The model's scope is defined as the Markov decision process (MDP) with **a) States.** which are defined as composed of the graph vertex state and the vehicle state, denoted as $s_t = \{x_t, v_t\}$ at step t, where $x_t = (u, q_t, e_t, l_t)$. The vehicle state v_t is composed of its load u_t, battery level E_t, and travelled time τ^t, expressed as $v_t = [E_t, \tau_t, u_t]$. **b) Actions.** a_t determines the node selection of the vehicle at step t. The sequence of actions generated from the initial to the final step should be combinations of nodes starting and ending with the depot. **c) Reward.** PDPTW aims to minimize fleet delay and travel time. Therefore,

the reward function is given in $R = \sum_{k \in N^r \cup N^d} \sum_{(i,j) \in N} t_{ijk} + \sum_{i \in P \cup D} \alpha_1 \max\{e_i - T_{ik}, 0\} + \sum_{i \in P \cup D} \alpha_2 \max\{T_{ik} - l_i, 0\}$, where t_{ijk} is time travelled by vehicle k between node i and j, and T_{ik} is the arrival time of the vehicle k at node i. Also, α_1 and α_2 are penalty factors. **d) Transition.** The system state will be updated from S_t to S_{t+1} based on the currently executed action a_t. The dynamic features of the problem, such as vehicle load, battery level, and travelled time, are being changed through consecutive nodes based on the vehicle's features.

2.2 Encoder. The incorporation of attention layers builds upon the recent work by (6, 3). Two adjacency matrices of spatial and temporal distribution are defined as $A_{i,j} = 1$ if the distance of the two customers is close and if the late time window l_i of customer x_i is nearest to late time window l_j of customer x_j; respectively; otherwise, $\boldsymbol{A}_{ij} = 0$. The spatial and temporal hop neighbourhood for a node x_i is defined by NB_i^S and NB_i^T, respectively, which limits the neighbourhood in adjacency matrices with a time window and distance threshold. A broad spectrum of customers' time and space correlation can be found by tuning these thresholds. First, the initial embedding for each node and both fleet edges is computed in Equation 1 and 2 through a linear layer.

$$\mathbf{h}_i^0 = \begin{cases} \mathrm{BN}\left(\boldsymbol{W}_1(\mathbf{x}_i; \mathbf{x}_{i+N}) + \boldsymbol{b}_1\right), & \text{if } i \in \{1, \ldots, N\}, \\ \mathrm{BN}\left(\boldsymbol{W}_2(\mathbf{x}_i) + \boldsymbol{b}_2\right), & \text{if } i \in \{0, N+1, \ldots, 2N\} \end{cases} \tag{1}$$

$$\hat{\mathbf{e}^{\mathbf{d}}}_{ij} = \mathrm{BN}\left(\boldsymbol{W}_3 E_{ij}^d + \boldsymbol{b}_3\right), \hat{\mathbf{e}^{\mathbf{r}}}_{ij} = \mathrm{BN}\left(\boldsymbol{W}_4 E_{ij}^r + \boldsymbol{b}_4\right), \text{ if } i, j \in NB_i^S \cup NB_i^T \tag{2}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{W}_3$, $\boldsymbol{W}_4$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\boldsymbol{b}_3$, and $\boldsymbol{b}_4$ represents the learnable parameters and $BN(.)$ represent batch normalization. The graph attention network can assign different importance to the customers within the neighbourhood through the attention mechanism (2) by computing the pairwise attention weight a^{ij} at lth layer as in equation 3:

$$\alpha_{ij}^{\ell} = \frac{\exp\left(\sigma\left(\boldsymbol{g}^{\ell^T}\left[\boldsymbol{W}^{\ell}\left(\boldsymbol{h}_i^{(\ell-1)} \left\| \boldsymbol{h}_j^{(\ell-1)} \right\| \hat{\boldsymbol{e}}_{ij}\right)\right]\right)\right)}{\sum_{z \in NB_i^S \cup NB_i^T} \exp\left(\sigma\left(\boldsymbol{g}^{\ell T}\left[\boldsymbol{W}^{\ell}\left(\boldsymbol{h}_i^{(\ell-1)} \left\| \boldsymbol{h}_z^{(\ell-1)} \right\| \hat{e}_{iz}\right)\right]\right)\right)} \tag{3}$$

where $(\cdot)^T$ represents transposition, $\cdot\|\cdot$ is the concatenation operation, $\boldsymbol{g}^{\ell}$ and $\boldsymbol{W}^{\ell}$ are learnable weight vectors and matrices respectively, and $\sigma(\cdot)$ is the softmax activation function. Afterwards, we use feed-forward with a residual connection and (BN) layer followed by calculating K multi-head attention of the weight value vector for lth layer $\boldsymbol{h}_i^l = \sum_{j \in NB_i^S \cup NB_i^T} a_{ij} \boldsymbol{W}_l^V \boldsymbol{h}_j^{(\mathbf{l-1})}$ as the output of the attention mechanism to get the final and average embedding.

2.3 Decoder. In the decoder, the context embedding of each vehicle, will be aggregated by fleet states concatenated with node embedding to get agent embedding as $\boldsymbol{x}_k^{(a)} = \boldsymbol{v}_{k,t} + \boldsymbol{W}_5 \cdot \left[\overline{\boldsymbol{h}}_{(N)}; \boldsymbol{v}_{1,t}; \boldsymbol{v}_{2,t}; \ldots; \boldsymbol{v}_{K,t}\right], \quad \forall k \in N^d, N^r.$

We define the query vector as the agent embedding $\boldsymbol{x}_k^{(a)}$, key vectors and value vectors as the customer embedding $\boldsymbol{h}_i$, and utilize the attention mechanism to compute the importance $\lambda_{k,i}$ of each customer i to agent k in $u_{k,i} = (\boldsymbol{W}_6 \cdot \boldsymbol{x}_k^{(a)})^T \cdot (\boldsymbol{W}_7 \cdot \boldsymbol{h}_i), \quad \forall i \in N, \quad \forall k \in N^d, N^r$. Next, we calculate the agent-customer joint information embedding as the weighted sum of value vectors in $h_{v,k} = \sum_{j \in N} \frac{e^{u_{k,i}}}{\sum_{j \in N} e^{u_{k,j}}} \cdot \mathcal{V}_i, \quad \forall k \in N^d, N^r$. Furthermore, the decoding process encodes the joint information embedding to a query. It compares it with the key of each customer to acquire the attention coefficient, which represents the compatibility between vehicle k and customer i at time t, in $\tilde{h}_{k,i} = (\boldsymbol{W}_8 \cdot h_{v,k})^T \cdot (\boldsymbol{W}_9 \cdot h_i), \quad \forall i \in N, \quad \forall k \in N^d, N^r$. To guarantee that each vehicle would not select the same node, a global mask is used to handle such situations and other operational and delivery constraints; the masking procedure is used for both fleets in the probability of the selecting node i, which is noted in $P(i) = softmax(C \cdot \tanh\left(\tilde{h}_{k,i}\right))$, with Clip parameter of C. As a result, a tour can be generated by vehicle-node pair selection at every step. The output will be given to the encoder for re-encode the states to account for removing the

visited node and re-embedding the edges states to update to features of the network; thereby, the decoder can be influenced by a variation on network embedding at the decoding step to select the superior pair.

3 Preliminary Results

The experiment is carried out for 10 requests, 20 nodes for pickup and delivery, where node locations are drawn by random number from unit square $[0, 1]km$ and time windows of nodes are driven by Poisson distribution for evening peak hour, e.g. 5 pm. in addition to random number from $[10, 50]min$ based on the node. The speed of the vehicles is set as $60km/h$ and $10km/h$ for drones and robots, respectively. The training curve for the cost for 100 epochs, and in each epoch, 2500 batches with 512 instances is depicted in Figure 3.

The test experiment is conducted by 10,000 instances with the same distribution, and the comparison baselines are Google OR-Tools, Lei et al. (3), and Fellek et al. (1). The test results for the graph size of 20 with the training parameters are shown in the Table 1. Furthermore, the gap denotes the average optimal gap between a result and the baseline solution, $\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \frac{L(\hat{\pi}|s) - L(\pi|s)}{L(\pi|s)}$.

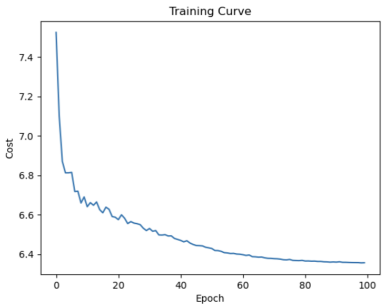

Figure 3 – *Convergence curve*

Table 1 – *Test Result*

Model	20 nodes - 2 vehicles	
	Cost (Gap)	CPU Time
OR-Tools	6.43 (0.00%)	3600 s
Lei et al.	6.26 (2.71%)	2 s
Fellek et al.	6.19 (3.88%)	5 s
Proposed	6.20 (3.71%)	2 s

4 Discussion

This study proposes a novel framework for multi-modal automated on-demand delivery in an urban environment through fusing nodes and edges in the dynamic transformer architecture. A parsimonious yet effective graph representation and dual re-encoding mechanism are updated at every decoder step to deal with uncertainty within a network layer. The initial result of the simulation of the network shows that this methodology can compete with recent work and global solvers. Future steps involve generalization, considering more extensive networks and fleets on the real case study of Mississauga, Canada, and network planning to reduce the computational time and quality of the solution.

References

[1] G. Fellek et al. Graph transformer with reinforcement learning for vehicle routing problem. *IEEJ Transactions on Electrical and Electronic Engineering*, 18(5):701–713, 2023.
[2] W. Kool et al. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.
[3] K. Lei et al. Solve routing problems with a residual edge-gann. *Neurocomputing*, 508:79–98, 2022.
[4] N. Patrinopoulou et al. Metropolis ii: Investigating the future shape of air traffic control in highly dense urban airspace. In *2022 30th MED*, pages 649–655. IEEE, 2022.
[5] K. Zhang et al. Transformer-based reinforcement learning for pickup and delivery problems with late penalties. *IEEE Trans. on ITS*, 23(12):24649–24661, 2022.
[6] K. Zhang et al. Graph attention reinforcement learning with flexible matching policies for multi-depot vehicle routing problems. *Physica A*, 611:128451, 2023.
[7] Z. Zong et al. Mapdp: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In *Proceedings of the AAAI Conference*, volume 36, pages 9980–9988, 2022.