Dynamic service operation for collaborative passenger-parcel transport: A deep reinforcement learning based approach

A. Yitong Yu[a] and B. David Z.W. Wang[a*]

[a] <School of Civil and Environmental Engineering>,< Nanyang Technological University>, <Singapore>

a.yitong001@e.ntu.edu.sg, *wangzhiwei@ntu.edu.sg

* 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 28, 2024

Keywords: Collaborative Passenger-Parcel transport, Last-mile Delivery, Deep Reinforcement Learning, Dynamic Attention Model

1 INTRODUCTION

In major cities, the rapidly growing urban population and fast-expanding e-commerce activities has presented significant challenges to both passenger transportation and city logistics, especially for last-mile passenger movement and goods delivery. It is particularly evident during morning and evening peak hours when an influx of commuters seeks transportation between their homes and the nearest public transportation station. The surging demand requires a large fleet of vehicles to do a demand-responsive transport service, while surplus transport capacity during off-peak hours is in a waste, leading to inefficiencies in utilizing the available transport resources. To address this challenge, this paper proposes a collaborative passenger and parcel transport service. This innovative approach allows for the transportation of both passengers and small parcels in the same vehicle. Priority is given to urgent passengers or parcels during peak hours, while others are served during off-peak hours by utilizing the spare capacity. Hence, it is crucial for operators to establish an optimal routing plan for the passenger-parcel transport.

However, the dynamic and fluctuating demand brings difficulties in making optimal routing plans. Since the demand is unknown before occurring, decisions are made based on incomplete information. The limited visibility makes it challenging to consider future impacts, impeding the optimization of routing plans. It necessitates a strategy to do routing plan in a dynamic environment. In this study, we deal with the dynamic vehicle routing problem with time windows for passenger-parcel transport of last-mile delivery (DVRPTW-PPL) to maximize profit.

Vehicle routing problems have been well studied in the literature, but few papers focus on dynamic vehicle routing problem due to the difficulties in designing an effective algorithm. Most of them used greedy methods which is short-sighted. To address such a complex optimization problem with long-term vision, deep reinforcement learning (DRL) is powerful, that makes decisions considering future impact. To the best of our knowledge, we are the first to design a DRL framework specifically for DVRPTW-PPL. Besides, we design a novel dynamic-attention model in this DRL framework. Attention model (Kool *et al.*, 2018) is influential in feature learning, greatly aiding in decision-making, and has never been applied to dynamic routing problems.

This algorithm designs a dynamic encoder-decoder architecture with attention layers to iteratively generate feasible routing solutions. The attention layer masks irrelevant features and learns by adaptively assigning weights to different parts of the input, enabling efficient and accurate learning of decision policies. Additionally, this algorithm addresses the challenge of making

decisions considering future demand, which is hard to achieve by optimization models and greedy algorithms. Notably, after offline training in ten thousand instances, the DRL framework demonstrates remarkable speed in generating solutions and shows good quality.

2 METHODOLOGY

2.1 Problem Description

This problem studies the last-mile delivery within a single community for both passengers and parcels. All demands originate from the community's depot, since passengers and parcels are centralized after long-haul transport (e.g., passenger by metro and parcel by trunk). Additionally, the demand is dynamically generated, meaning it is unknown until it arrives at the depot. Decisions are made whenever new demands arise from the depot. The operator's task is to assign vehicles to deliver demands to their destinations and maximize profit. The profit comes from the revenue of fulfilling demands and the cost of traveling. The decision entails determining loaded demands and delivery route for each vehicle. It is assumed that an adequate number of vehicles are available at the depot. All vehicles must load demand at the depot and return to the depot after delivery. Once decisions are made at the depot, they cannot be altered, and vehicles must deliver all demands on board before returning to the depot.

This problem is structured as the spatial-temporal graph s, where station is defined as $j \in J = \{0, 1, ..., \bar{j}\}$. Depot is represented by 0. Each station feature consists of the 2-dimensional coordinate in Euclidean space $y_j = (\alpha_j, \beta_j)$. Demand is generated dynamically over time $t \in \{1, ..., \bar{t}\}$. Each demand $i \in I = \{0, ..., \bar{i}\}$ incorporates feature x_i from eight dimensions, $x_i = [g_i = t, v_i = j, \alpha_j, \beta_j, d_i, l_i, \gamma_i, p_i], i \in I \backslash \{0\}$. The first element $g_i = t$ is the demand generation time. The second element $v_i = j$ is the delivery station required by demand i. The fifth d_i is the demand quantity. The sixth element l_i is the time window of delivery. The seventh element γ_i is the demand type, either passenger or parcel. The eighth element p_i is the income of the operator for serving demand i.

Decisions are made sequentially over time. Due to the limited visibility, the operator only knows about demands generated up to the current moment. Whenever new demands arise, the operator needs to re-plan the route for available vehicles. We define the decision made for the demand generated up to time t as partial solution $\pi_t = \{\pi_{t,1}, ..., \pi_{t,w}, ..., \pi_{t,\bar{w}}\}$. It contains the route $\pi_{t,w}$ for each vehicle w. Specifically, route $\pi_{t,w}$ comprises the visiting sequence, $\pi_{t,w} = \{0, \pi_{t,w,1}, ..., \pi_{t,w,n}, ..., \pi_{t,w,\bar{n}}\}$, where 0 means the vehicle starts from the depot, and $\pi_{t,w,n}$ is the n-th visiting demand by vehicle w, among demands generated up to time t. It should be noted that each demand is visited only once while the depot can be visited multiple times in π_t.

2.2 Markov Decision Process

The DVRPTW-PPL is a dynamic decision problem, so we model it as a Markov decision process (MDP) by five components: exogenous information, state variable, decision variable, transition function, and objective function.

Exogenous Information. The demand is dynamic and is unknown in advance, so the exogenous information comprises the cumulative demand observed up to time t and is updated whenever new demands occur. Here we defined the exogenous information at t as U_t, $U_t = \{x_i | g_i \leq t, i \in I\}$, where x_i is the feature of demand i.

State Variables. To fulfill demands observed up to time t, the state occurs when a vehicle w is assigned to visit a node $\pi_{t,w,n}$ at sequence n. We denote this state as $State_{t,w,n}$, where $t \in T$ represents the time point up to which demands have been observed, $w \in W$ represents the vehicle index, and $n \in W$ represents the sequence that a vehicle visits the node. The state variable contains four components: 1. The node selected for the vehicle, denoted as $\pi_{t,w,n}$. The vehicle is currently visiting this node. 2. The time that the vehicle visits the node, denoted as $\tau_{t,w,n}$. 3. The remaining capacity of the vehicle, denoted as $D_{t,w,n}$. 4. Nodes that have been assigned to

vehicles before visiting node $\pi_{t,w,n}$. The previously assigned node is denoted as $prev_{t,w,n}$. The overall state variable takes the form of: $State_{t,w,n} = (\pi_{t,w,n}, \tau_{t,w,n},\ D_{t,w,n}, prev_{t,w,n})$.

Decision Variables. Based on the $State_{t,w,n}$, when the vehicle is at the node $\pi_{t,w,n}$, decisions are made to determine the next visiting node. The action space, denoted as $A_{t,w,n}$ is based on the demand provided by exogenous information U_t , $A_{t,w,n} = A_{t,w,n}(U_t)$. We define $\phi_{t,w,n} \in A_{t,w,n}$ as the decision variable determining the next visiting node for the vehicle currently at node $\pi_{t,w,n}$. To ensure feasibility, action space must obey some conditions: 1. the planned visiting time for the next node cannot exceed its time window. 2. the vehicle's remaining capacity should be greater than zero after loading the next node's demand. 3. nodes that have been assigned to vehicles before cannot be chosen again.

Transition Function. Knowing the exogenous information U_t, the $State_{t,w,n}$ and decision $\phi_{t,w,n}$, the transition function $F(U_t, State_{t,w,n}, \phi_{t,w,n})$ update for the next state:

Most commonly, when the route plan of vehicle w for cumulative demand up to t hasn't been finished (i.e., the vehicle hasn't returned to the depot), the operator selects the next node $\phi_{t,w,n}$ for vehicle. All state variables related to n are updated to $n+1$. The current visiting node by vehicle is replaced by the decision, $\pi_{t,w,n:n+1} = \phi_{t,w,n}$. The vehicle visiting time is updated by adding the travel time between two nodes, $\tau_{t,w,n:n+1} = \tau_{t,w,n} + T^{\pi_{t,w,n}}_{\phi_{t,w,n}}$. The remaining capacity is reduced by the capacity of the selected node, $D_{t,w,n:n+1} = D_{t,w,n} - d_{t,w,n}$. The assigned nodes increases by contacting the selected node, $prev_{t,w,n:n+1} = [prev_{t,w,n}; \phi_{t,w,n}]$.

It occurs that vehicle w returned to the depot, while other vehicles are pending assignment at the depot. The operator will assign for a new vehicle: $w := w+1$, $n := 0$, $\pi_{t,w:w+1,n:0} = 0$, $\tau_{t,w:w+1,n:0} = t$, $D_{t,w:w+1,n:0} = D$, $prev_{t,w:w+1,n:0} = [prev_{t,w,n}; \phi_{t,w,n}]$. Furthermore, when all vehicles selected to fulfill cumulative demand t return to the depot, the operator plans to serve new demand $t := t+1$ by assigning a new vehicle $w+1$: $U_{t:t+1} = \{x_i | g_i \leq t+1, i \in I\}$, $\pi_{t:t+1,w:w+1,n:0} = 0$, $\tau_{t:t+1,w:w+1,n:0} = t$, $D_{t:t+1,w:w+1,n:0} = D$, $prev_{t:t+1,w:w+1,n:0} = [prev_{t,w,n}; \phi_{t,w,n}]$.

Objective Function. A solution for DVRPTW-PPL is a decision policy $\theta \in \Theta$,and a decision rule Φ^θ mapping exogenous information U_t and $State$ to decision $\phi_{t,w,n} = \Phi^\theta(U_t, State_{t,w,n})$. Each decision contributes a profit $R(U_t, State_{t,w,n}, \phi_{t,w,n})$, including the income of fulfilling demand and the cost of traveling. The objective is to find a policy θ^* to maximize the profit:

$$\max_{\theta \in \Theta} \underset{s \in S}{E} \left\{ \sum_{t,w,n} R\left[(U_t, State_{t,w,n}), \Phi^\theta(U_t, State_{t,w,n}) | State_{0,0,0} \right] \right\} \tag{1}$$

where $State_{new} = F\left(U, State, \Phi^\theta(State)\right)$.

3 SOLUTION APPROACH

Dynamic Attention Model. This problem can be viewed as the *graph attention network*, for which the attention model is effective (Kool *et al.*, 2018). Unlike the traditional one, we propose the dynamic attention model using dynamic encoder-decoder. It dynamically updates the feature embeddings and filters invalid information, improving the accuracy of feature learning. The computation process and formula of dynamic encoder-decoder are in Fig.1,2. The mathematical form is not shown in extended abstract. The algorithm framework is as follows (Fig.3):

Step1: Input graph instance s, exogenous information $U(s)$ and initialize state variables.

Step2: Obtain partial solution π_t for demand $U_t(s)$, by dynamic encoder-decoder (Fig.1).

Step2.1: *Encoder* embeds the input (Fig.2) through Multi-Head attention sublayer and Feed Forward sublayer for L time, then output embeddings as the input for *Decoder*

Step2.2: *Decoder* construct the route for vehicle w incrementally, selecting one node at each action step. The construction stops when the vehicle returns to the depot, then outputs the route $\pi_{t,w}$, and updates vehicle to $w := w+1$. Specifically, the decoder predicts probability distribution $P(\pi_{t,w,n+1} | s, prev_{t,w,n})$ over nodes, and masks infeasible nodes. Then one node is selected, added to the route, and states are updated.

Step2.3: Input the lastest state to step 2.1, and conduct loop2.1-2.3 until the operator stops

assigning any other vehicles to fulfill cumulative demand t. Then we output the partial solution π_t, which consists of several routes serving cumulative demand t.

Step3: Loop step2 to construct the partial solution for each cumulative demand. The time point is updated to $t := t + 1$ at the end of each loop. The loop stops until all demands are visited. We finally summarize all partial solutions to obtain the solution π.

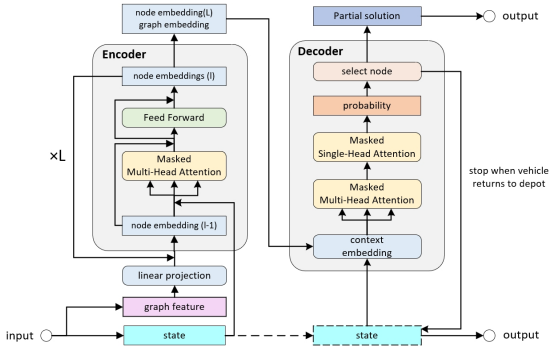

Figure 1 – *Dynamic Encoder-Decoder*

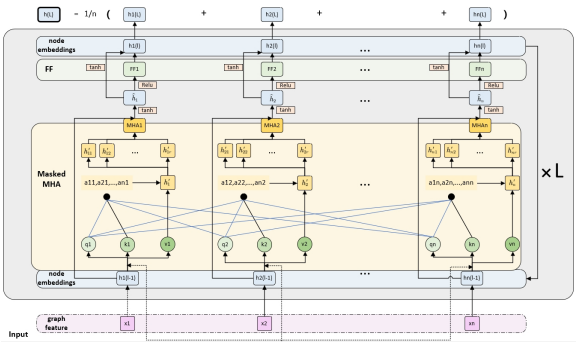

Figure 2 – *Encoder*

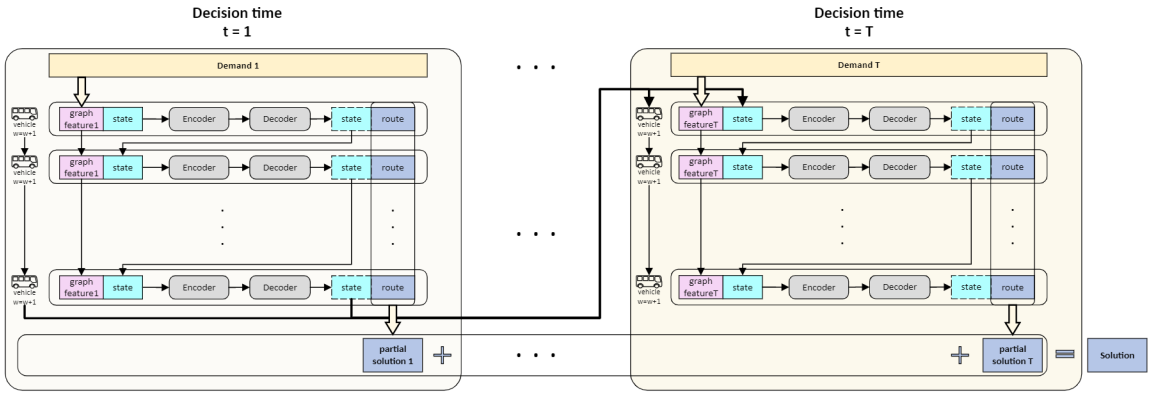

Figure 3 – *Algorithm framework*

Model Training. Dynamic attention model above generates the solution for DVRPTW-PPL. To optimize the solution, we adopt the policy gradient using REINFORCE algorithm. The policy θ represents the probability distribution parameter in attention model. We use sample rollout and greedy rollout to measure the value of the trained model and baseline. Then compute the gradients by $\nabla_\theta J(\theta|s) \approx 1/B \sum_{b=1}^{B} \left(L(\pi_b^S) - L(\pi_b^G)\right) \nabla_\theta \; log \; P_\theta(\pi_b^S|s_b)$, $P_\theta(\pi|s) = \prod_{t,w,n} P_\theta(\pi_{t,w,n}|s, prev_{t,w,n-1})$, where B is the batch size, $L(\pi_b^G)$ is the baseline, π_b^S, π_b^G is computed by $SampleRollout\,(s_b, P_\theta)$ and $GreedyRollout\,(s_b, P_{\theta_{BL}})$. Using the above procedures, we train the attention model for about 15000 epochs. The current demand the impact of future demand are jointly considered during learning. We periodically update policy and finally get the optimal one θ^*.

4 RESULT AND DISCUSSION

We set 30 stations, 10 time intervals (each for 10mins), and randomly generated demands of passengers and parcels. The model is trained by 512 batch sizes for 20000 epochs. So far, the training and test results have converged. The trained model can generate a solution for a given instance within a few seconds, and outperforms the greedy approach, which is mainly attributed to the consideration of both current and future impact. Subsequently, we will test larger datasets, do the sensitivity analysis, and compare the results with other heuristic algorithms.

References

Kool, Wouter, Van Hoof, Herke, & Welling, Max. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475.*