Adaptive routing and scheduling of network-wide rail transit services with flexible train composition

Shou-yi Wang[a], Andy H.F. Chow[a*] and Cheng-shuo Ying[b]

[a] Department of Systems Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong SAR

[b] College of Metropolitan Transportation, Beijing University of Technology, Beijing, China

shouywang3-c@my.cityu.edu.hk, andychow@cityu.edu.hk, icsying@outlook.com

* Corresponding author

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)

June 22-27, 2025, Okinawa, Japan

February 27, 2025

Keywords: transit routing, service scheduling, flexible train composition, reinforcement learning

1 Introduction

This paper presents a novel adaptive control framework with flexible train composition for routing and scheduling in network-wide rail transit services. This framework aims to minimize passenger waiting times and operating costs driven by stochastic travel demand. The control problem is formulated as a Markov decision process (MDP) to reflect the practicality in real-world applications. To address the computational challenges associated with the control problem, deep reinforcement learning techniques are applied to seek potential optimal solutions to the optimization problem. The proposed control framework is tested using real-world scenarios and the data collected from the Hong Kong Light Rail Transit (LRT) network. The experiment results demonstrate that the proposed routing and scheduling control framework using flexible train composition can effectively reduce passenger waiting time and operating costs. This study contributes to the real-time routing and scheduling of network-wide rail transit services by integrating advanced optimization technology.

2 Methodology

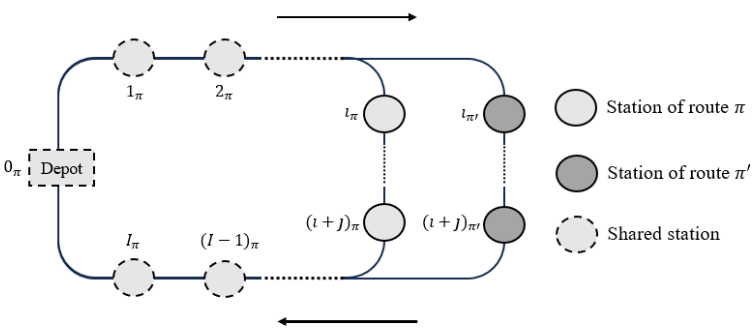


Figure 1 – *An example of service runs of the transit network.*

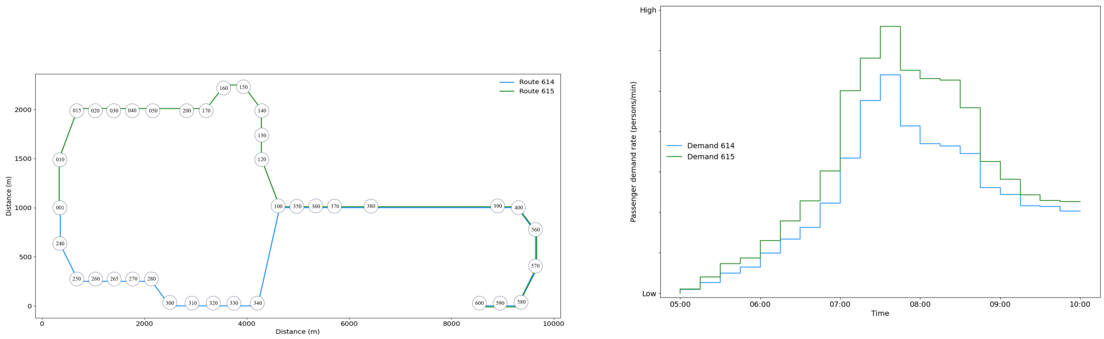
We consider a rail transit network consisting of a set of routes denoted as Π. Each specific route $\pi \in \Pi$ consists of an ordered set of stations $\mathcal{I}_\pi = \{1_\pi, 2_\pi, ..., \imath_\pi, ..., (\imath + \jmath)_\pi, ..., (I-1)_\pi, I_\pi\}$. Here $\{\imath_\pi, ..., (\imath + \jmath)_\pi\}$ denotes the stations exclusive to a single route (i.e. non-shared stations), while $\{1_\pi, 2_\pi, ..., (I-1)_\pi, I_\pi\}$ denotes the stations utilized by multiple routes (i.e. shared stations). Figure 1 provides a basic example of a transit network. In this network, a service run on route π will depart from depot 0_π, follow the ordered set of stations $\mathcal{I}_\pi$ on route π towards the terminal station I_π, and then return to depot 0_π. To provide high-quality services to passengers in crowded transit networks, this study focuses on determining the dispatching headway and dispatching route for each service run. Instead of fixed fleet sizes, each service run departing from the depot consists of one driving module and a variable number of trailers. The advantage of this flexible fleet size strategy is that it enhances passenger satisfaction and helps mitigate congestion. Consequently, the optimization problem involves determining the dispatching headway, dispatching route, and assigned fleet size for each service run, ensuring a real-time response to dynamic variations in passenger demand. Furthermore, the objectives of this study are to minimize both the total passenger waiting time and the operating cost of the rail transit network.

Given the specific configuration of the transit network, the optimal control system for routing and scheduling service runs can be formulated as a Markov decision process (MDP) (Bertsekas, 2019). To accurately capture passenger boarding and alighting details, the state needs to encompass not only the dispatching information of the current service run but also those of previous service runs and historical passenger demand information (with dimensions increasing over time t). Consequently, the state space becomes huge and varying, posing a significant challenge when applying reinforcement learning in real-world scenarios. In such cases, a partially observable Markov decision process (POMDP) (Powell, 2007) is considered in our study. In a POMDP, the controller no longer knows directly the system state. Instead, the controller receives an observation generated from the underlying system state and makes the decision based on the observation. In this study, the observation set $\boldsymbol{o}_n$ be defined as the number of remaining passengers l^{n-1}_{K,i_π} at each station after the last train service has passed, as well as the last departure time $\hat{\sigma}^{n-1}_{K,i_\pi}$ at each station at stage $n-1$. Associate with the observation, the decision vector $\boldsymbol{x}_n = (\pi_n, u_n, h_n)$ contains the chosen dispatching route π_n of service run n, the assigned train composition u_n of service run n, and the headway between service run $n-1$ and n. At decision stage n, the Markovian system is in a specific state $\boldsymbol{s}_n$. The service run n associated with the decision $\boldsymbol{x}_n$ is dispatched based on the given observation $\boldsymbol{o}_n$. Subsequently, the system received next observation $\boldsymbol{o}_{n+1}$ according to the transition function $\boldsymbol{s}_{n+1} \sim \boldsymbol{P}(\boldsymbol{s}_n, \boldsymbol{x}_n)$ and receives a reward r_n from the reward function $r_n \sim \boldsymbol{R}(\boldsymbol{s}_n, \boldsymbol{x}_n)$. The terminal condition of the dynamic system is specified by a binary variable η_n, where $\eta_n = 1$ if all passenger demand in the transit network has been served at stage n, and $\eta_n = 0$ if otherwise.

To overcome the computational challenge of the optimal control problem, we adopt a multi-layer artificial neural network (ANN) surrogate, denoted as Q_ϕ, defined by a parameter set $\boldsymbol{\phi}$. The ANN surrogate comprises an input layer, an output layer, a long short-term memory (LSTM) layer (Hochreiter & Schmidhuber, 1997), and two fully connected layers with element-wise non-linear activation function (e.g. hyperbolic tangent (Tanh) and rectified linear unit (ReLU) function). Incorporating an LSTM layer in the framework allows for integrating a long history of observations, enabling more accurate estimation of the Q-value (Hausknecht & Stone, 2015). Compared to linear kernel regression models, ANN surrogates can capture more sophisticated system dynamics. Before the proposed control framework can achieve satisfactory performance, it is necessary to train the parameters $\boldsymbol{\theta}$ of the ANN surrogate. This study uses a deep recurrent Q network (DRQN) training algorithm (Hausknecht & Stone, 2015) due to its computational efficiency and stability in partial observation environments.

3 Results and discussion

The proposed optimization framework is tested on three routes of the Hong Kong Light Rail Transit (LRT) network, as shown in Figure 2a, and Figure 2b shows the average passenger demand rate over time. A total of $\hat{U} = 50$ train units are available for dispatching, and each train unit can carry up to $Cap = 200$ passengers.



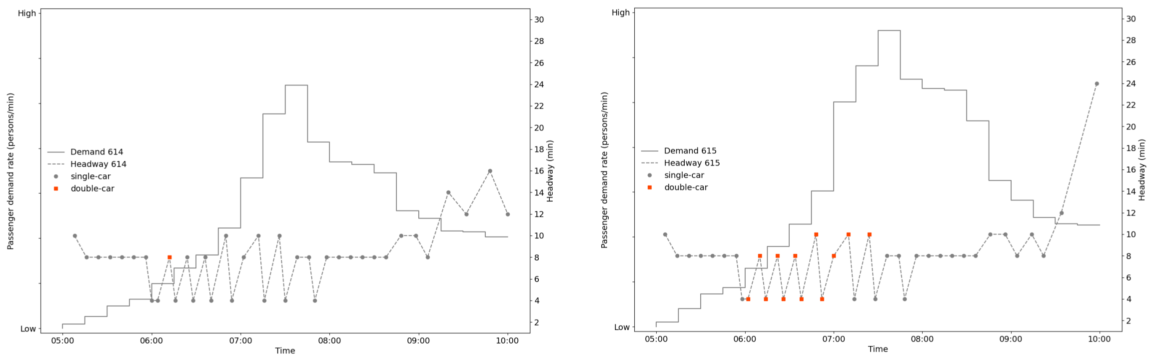
(a) *routes in the test network* (b) *Passenger demand in the test network*

Figure 2 – *Existing configuration of the test network.*

According to the MTR operational guidelines, the safety separation between successive train services Δ_S and the shunting time $\Delta_{\mathcal{T}}$ are both set to 60s and the nominal dwell time w_i at each station is set to 30s. There are five admissible levels of dispatching headways, which are 2, 4, 6, 8, and 10 minutes, with corresponding $h_{min} = 2$ and $h_{max} = 10$. Furthermore, the minimum headway H_{min} and maximum headway H_{max} for each route are set to 2 minutes and 30 minutes, respectively. Considering vehicle characteristics, each service run is restricted to a maximum of two train units (i.e. $u_{max} = 2$).

Figure 3 first presents the dispatching headways and train unit deployment for each route at the lowest achieved total cost. In the figure, the 'point' and 'square' markers represent dispatches of 'single-car' and 'double-car' trains, respectively. The figure also includes the average passenger demand profile corresponding to each route as a reference. Specifically, in Figure 3, it can be observed that the controller dispatches more train units with shorter headways between 6:00 and 8:00 to cater to the potential peak passenger demand between 7:00 and 9:00. Figure 3 also shows the controller maintains relatively stable dispatching headways during the periods of 5:00 to 6:00 and 8:00 to 9:00 to meet the passenger demand during the off-peak period (i.e., before 7:00 and after 9:00). After 9:00, taking advantage of the remaining service resources in the transit network, the controller dispatches service runs with longer headways. This strategy helps minimize operating costs without significantly decreasing passenger satisfaction, showing the adaptiveness of the control framework to the prevailing passenger demand.

For benchmarking purposes, we compare the proposed DRQN algorithm with the well-established GA (genetic algorithm) and PSO (particle swarm optimization). Table 1 shows the detailed performance. It is observed that the reinforcement learning algorithms outperform GA and PSO in total cost. That can be attributed to the approximation of the state space based on the ANN surrogate, which is specifically tailored to the original service scheduling problem. As a result, specialized optimization techniques such as the ADAM algorithm can be employed for the training process instead of relying on less efficient meta-heuristic algorithms like GA. Table 1 also shows the performance of two benchmark control settings to test the effectiveness of the proposed flexible routing and train composition strategy. The first benchmark control setting is 'fixed train', where a flexible routing strategy is allowed, but only fixed train compositions are considered. In this setting, double-car trains are used for all service runs on the lines. The second benchmark control setting is 'fixed routing', where flexible train composition is allowed, but



(a) *Route 614* (b) *Route 615*

Figure 3 – *Dispatching headways and train units deployment with respect to passenger demand profile.*

service runs are dispatched in a fixed sequence (e.g., ...-615-614-615-614-...) until all passengers have been served.

Table 1 – *Performances of different models.*

Model	Total cost ($)	Passenger cost ($)	Operating cost ($)	Number of services	Average train units
Base-DRQN	**1175901**	**389412**	**786489**	76	1.15
Base-GA	1326712	428421	898291	83	1.18
Base-PSO	1304314	411197	893117	82	1.20
Fixed train-DRQN	1409256	510618	898637	53	2.00
Fixed routing-DRQN	1239171	431931	807240	77	1.21

From the result of Table 1, we can also observe the proposed model (denoted as the 'base model' in the table) delivers the lowest costs. Specifically, compared to the 'fixed train' model and the 'fixed routing' model, the base model achieves a reduction in total cost, with a 19.8% and 5.4% reduction, respectively. This reduction can be attributed to the flexibility of the base model in train composition and routing based on passenger demand in the transit network. Furthermore, the base model shows a 10.9% reduction in passenger costs compared to the 'fixed routing' model, despite the latter dispatching more service runs with higher average train units. This reduction in passenger costs can be attributed to the base model's flexible routing, which allows for increased service runs on highly demanded service lines during peak hours. Additionally, it can be observed that both the base model and the 'fixed routing' model dispatch fewer trains to reduce operating costs. Consequently, although the 'fixed train' model dispatches fewer service runs, its operating costs remain significantly higher than the other two models. These findings suggest the importance of having flexible train composition and routing for optimizing the operations of a transit network throughout different times of the day.

References

Bertsekas, Dimitri. 2019. *Reinforcement learning and optimal control.* Vol. 1. Athena Scientific.

Hausknecht, Matthew, & Stone, Peter. 2015. Deep recurrent q-learning for partially observable mdps. *In: 2015 aaai fall symposium series.*

Hochreiter, Sepp, & Schmidhuber, Jürgen. 1997. Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Powell, Warren B. 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality.* Vol. 703. John Wiley & Sons.