Deep Reinforcement Q-Learning for Intelligent Traffic Control in Mass Transit

Shurok Khozam, Nadir Farhi

▶ **To cite this version:**

Shurok Khozam, Nadir Farhi. Deep Reinforcement Q-Learning for Intelligent Traffic Control in Mass Transit. Sustainability, 2023, 15 (14), pp.11051. 10.3390/su151411051 . hal-04420008

HAL Id: hal-04420008
https://hal.science/hal-04420008

Submitted on 4 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

sustainability

MDPI

Article

Deep Reinforcement Q-Learning for Intelligent Traffic Control in Mass Transit

Shurok Khozam and Nadir Farhi *

Cosys-Grettia, University Gustave Eiffel, F-77447 Marne-la-Vallée, France; shurok.khozam@univ-eiffel.fr
* Correspondence: nadir.farhi@univ-eiffel.fr

Abstract: Traffic control in mass transit consists of the regulation of both vehicle dynamics and passenger flows. While most of the existing approaches focus on the optimization of vehicle dwell time, vehicle time headway, and passenger stocks, we propose in this article an approach which also includes the optimization of the passenger inflows to the platforms. We developed in this work a deep reinforcement Q-learning model for the traffic control in a mass transit line. We first propose a new mathematical traffic model for the train and passengers dynamics. The model combines a discrete-event description of the vehicle dynamics, with a macroscopic model for the passenger flows. We use this new model as the environment of the traffic in mass transit for the reinforcement learning optimization. For this aim, we defined, under the new traffic model, the state variables as well as the control ones, including in particular the number of running vehicles, the vehicle dwell times at stations, and the passenger inflow to platforms. Second, we present our new deep Q-network (DQN) model for the reinforcement learning (RL) with the state representation, action space, and reward function definitions. We also provide the neural network architecture as well as the main hyper-parameters. Finally, we give an evaluation of the model under multiple scenarios. We show in particular the efficiency of the control of the passenger inflows into the platforms.

Keywords: mass transit; deep reinforcement learning; traffic control

check for updates

Citation: Khozam, S.; Farhi, N. Deep Reinforcement Q-Learning for Intelligent Traffic Control in Mass Transit. *Sustainability* **2023**, *15*, 11051. https://doi.org/10.3390/su151411051

Received: 15 May 2023
Revised: 4 July 2023
Accepted: 4 July 2023
Published: 14 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction and Related Work

Mass transit is a crucial transportation system that profoundly impacts various aspects of daily life. However, it faces numerous disruptions, including mechanical defects, energy supply issues, information system malfunctions, and external environmental factors. Furthermore, the constant rise in travel demand in large cities worldwide presents additional challenges for mass transit systems. Operating under high travel demand and passenger densities exacerbates the impact of delays, affecting not only a single line but also the entire network.

To enhance efficiency, reduce latency, save time, improve reliability, and minimize disruptions, it becomes imperative to analyze the operations of mass transit systems under high demand scenarios. Our study introduces a significant novelty by proposing an optimization model that employs deep reinforcement learning (DRL) to manage traffic in mass transit systems. Taking a comprehensive and holistic approach, our research aims to surpass previous achievements by leveraging a traffic model and DRL algorithm to optimize various aspects, including train operations, passenger experiences, and overall system efficiency.

With a specific focus on a circular mass transit line without junction, we address the complexities associated with vehicular and passenger traffic. Our optimization model targets criteria such as train frequency and passenger comfort, capturing the dynamics of trains, interactions with passengers, and control variables. Our novel traffic model aims to enhance performance in terms of operating costs, travel times, and passenger satisfaction.

1.1. Related Work

Reinforcement learning algorithms have emerged as a promising option for tackling traffic management in mass transit, although most optimization problems related to complex systems have yet to implement optimal solutions. Studies have attempted to overcome capacity constraints in urban public transportation during peak hours by employing Q-learning methods to regulate passenger entry onto platforms and curb congestion at specific stations [1]. Reinforcement learning techniques have also been utilized to improve service regularity and decrease headway deviation in bus-holding problems as well as to optimize scheduling and train deployment in metro networks to reduce operational costs and enhance service regularity [2,3].

To address the issue of unforeseen disruptions that impact train service scheduling, stochastic methods have been found to perform better than deterministic rolling-horizon methods in generating rescheduling solutions [4]. Other studies have focused on improving the reliability of railway systems, including punctuality to face train speed variations and external environmental factors [5]. Q-learning has also been employed to handle train delays in Slovenia, with promising results, and to optimize passenger waiting and travel times using a new deep neural network algorithm called Auto-Dwell [6]. Another attempt in [7] presents a novel application of reinforcement learning techniques in traffic signal control. This paper introduces a multi-agent deep Q-learning algorithm. The study demonstrates high potential in enhancing traffic light controller coordination and mitigating the negative effects of traffic congestion. The proposed approach aims to manage traffic networks more efficiently, and it yields substantial improvements in transportation systems, achieving an impressive 11% reduction in fuel consumption and a notable 13% decrease in average travel time.

Other approaches proposed multi-agent deep reinforcement learning (MDRL) to manage the bus bunching problem and to solve the train timetable rescheduling (ETTR) problem while minimizing energy consumption amidst random disturbances, respectively [8,9]. Similar to the latter references, Yan et al. addressed in [10] the problem of multiline dynamic bus timetable optimization. Their approach involved employing a multi-agent deep reinforcement learning framework to overcome computational limitations and improve efficiency. After applying their method to multiple bus lines in Beijing, China, their results showcased a substantial 20.30% reduction in operating and passenger costs when compared to actual timetables. Recent research has focused on improving passenger satisfaction by proposing algorithms that minimize delay and travel time in the face of uncertain disturbances. An algorithm was developed and tested on real-world cases to propose optimal solutions, and DQN reinforcement learning methods were proposed to handle larger-scale real networks and ensure passenger satisfaction [11–13]. Additionally, DQN and policy gradient approaches were suggested to optimize traffic management by rescheduling traffic lights or duration in order to minimize vehicle travel time [14].

1.2. The Objectives of This Work

The objective of this study is to propose an optimization model using deep reinforcement learning (DRL) for managing traffic in mass transit, encompassing both vehicular and passenger traffic. The scenario we consider is a circular mass transit line with no junctions, where trains operate without overtaking each others and they stop at all stations. Passengers wait at platforms for the train's arrivals, board, and disembark at their respective destinations. In this work, we use a traffic model that describes both train dynamics and interactions between trains and passengers. The objective is to optimize various criteria, including train frequency, operating costs (in particular the number of running trains), passengers' travel and waiting times, passengers' comfort at platforms and inside the trains, etc. The control variables in this problem are the dwell times of trains at the platforms and the inflow of passengers onto the platforms. Our study introduces a significant novelty by proposing an optimization model that utilizes deep reinforcement learning (DRL) for the management of traffic in mass transit systems. Through a comprehensive and holistic

approach, our research aims to surpass previous achievements by leveraging a traffic model and DRL algorithm to optimize various aspects, including train operations, passenger experiences, and overall system efficiency. We specifically address the complexities associated with vehicular and passenger traffic in a circular mass transit line without junction with a focus on optimizing criteria such as train frequency and passenger comfort. Our novel traffic model captures the dynamics of trains, interactions with passengers, and control variables, with the objective of improving performance in terms of operating costs, travel times, and passenger satisfaction. Moreover, our study extends the existing traffic model by incorporating the control of passenger inflows to platforms, resulting in a comprehensive optimization approach. In contrast to prior studies that relied on discrete event systems modeling, our work harnesses the capabilities of deep reinforcement learning to propose a fresh optimization technique for the management of mass transit traffic.

Our model is developed using a case study on Metro Line 1 of Paris. We obtained the line's parameters and characteristics from a dataset provided by RATP Group, which is the operator of Paris metro lines. Metro Line 1 spans a distance of 16.5 km and has 25 stations, connecting the western and eastern parts of Paris. The journey takes approximately 36 min. Historically, Metro Line 1 Paris has been the busiest one in the network. In 2010, the line transported 207 million passengers, averaging 184.4 million passengers annually, with a peak of 750,000 passengers per day in 2018 [15].

1.3. Overview of the Proposed Approach

We developed our DRL algorithm which is based on a traffic model described in subsequent sections. This traffic model serves as a simulator for the environment in which our RL model operates. Our approach has the objective of improving what has already been realized in previous studies by proposing a new optimization technique, as it will be illustrated later on. Previous related studies have used discrete event systems modeling and Max-plus algebra approaches to describe vehicular and passenger traffic in mass transit systems. Specifically, the train dynamics are represented by a discrete event system that captures various traffic constraints such as dwell and safe-separation time constraints [16–18]. Additional research has extended this approach to incorporate passenger demand [19–24], model traffic on mass transit lines with junctions [25–27], and simulate mass transit lines with skip-stop policies [28,29]. In our work, we extend the existing traffic model by introducing the control of passenger inflows to platforms. Our approach also optimizes the traffic of the entire system, taking into account various criteria such as train frequency, operating costs, wait and travel times for passengers, and passenger comfort for passengers inside the trains and at the platforms.

1.4. Main Assumptions

We consider the following assumptions:

- An automated circular metro line without junction, where trains move without passing each others and they stop at all stations.
- Passenger demand volumes as well as an origin–destination travel matrix are provided. The optimization problem is solved with an origin–destination (OD) matrix for the flow demand of passengers from one station to another.
- The operator is able to control the passenger inflows to platforms by closing and opening station's gates; removing, adding or inverting escalators; etc.

1.5. Problem Formulation

The problem addressed in this study can be summarized as follows: Our approach involves the utilization of a double deep Q-learning (DDQN) agent to optimize public transportation traffic. The primary objective is to enhance passenger satisfaction by improving their comfort and minimizing waiting times within stations. Additionally, we aim to minimize costs for operators by reducing the number of operating vehicles on the same line. To achieve this, we develop a mathematical traffic model for the dynamics of both trains

and passengers. This model incorporates a macroscopic representation of passenger flows. Consequently, this newly proposed model serves as the environment for reinforcement learning optimization in the context of mass transit traffic management.

1.6. Contributions

The main contributions of this work are summarized as follows:

- We extended the modeling of the train and passengers dynamics, as well as the modeling of their interactions, based on the works cited previously [17,23,24].
- We introduced a new DDQN model that aims to optimize traffic flow in mass transit lines, specifically in automated circular lines. Our model considers eighty-one possible actions, each of which is a multi-action represented in the ternary counting system and affects multiple traffic parameters simultaneously. Furthermore, we provide a comprehensive explanation of the DDQN architecture and its hyper-parameters beside the reward function, which is defined in a way to cover multiple objectives to ensure a clear understanding of the implementation process.
- In addition, we are optimizing not only the cost of number of running vehicles respecting passengers' comfort but also the passengers inflow to the platforms.
- We evaluated the robustness of performance of our DDQN model through three scenarios of passenger travel demand: nominal, high and ultra-high levels.
- We interpreted the results obtained by the DDQN optimization and investigated the ability of the proposed methodology to be applied in real life.

2. Deep Reinforcement Q-Learning

2.1. Q-Learning Background

Q-learning is a model-free and off-policy reinforcement learning algorithm [1]. The Q-learning agent (controller) acts on a Markov Decision Process (MDP) defined by the tuple <S, A, P, R>, where S is the continuous state space with state $s \in S$; A is the discrete action space with action $a \in A$; P is the transition operator with $p = P(s'|s, a)$ representing the probability of transitioning to state s' after taking action a in state s; R is the reward function with $r' = R(s, a, s')$ representing the reward of taking action a in state s and transitioning to state s'; and the transition is noted as (s, a, s', r'). The objective of the learning process is to determine a policy $\pi(s)$ mapping from states to actions that maximizes the cumulative discounted reward $G_t = \sum_{k=t}^{t+T} \gamma^{(k-t)} \cdot r_{k+1} = r_{t+1} + \gamma \cdot G_{t+1}$, where $\gamma \in [0, 1]$ is a discount factor and T is a finite time horizon.

The action-value function $Q(s, a) := \mathbb{E}[G_t \mid S_t = s, A_t = a]$ estimates, for state–action pairs, expected cumulative discounted rewards for successive states in the MDP, with the Bellman recursion. For an optimal policy $\pi^*(s)$, and associated optimal action-value function $Q^*(s, a)$, the Bellman equation is written:

$$Q^*(s, a) = \mathbb{E}[r_{t+1} + \gamma \cdot \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a, \pi^*].$$

Thus, a Q-learning algorithm approximates the action-value function $Q(s, a)$ by iterating a parameterized function $Q(s, a, \Theta)$ which tends to the optimal action-value function $Q^*(s, a) = Q(s, a, \Theta^*)$, with Θ representing a vector of approximation parameters, and Θ^* representing the parameters' values at convergence.

2.2. Deep Reinforcement Q-Learning (DQN) and Double DQN Algorithm

The deep Q-learning (DQN) algorithm has the same principle as the Q-learning algorithm, but the action-value function Q is approximated with a neural network, with weights Θ. The neural network maps a state s (with a number of entries given by the dimension of the vector s) to the number $|\mathcal{A}|$ of all actions [30].

Double DQN Algorithm (DDQN)

In the DQN algorithm, there is generally one action value function (Q-function). In contrast, in the DDQN algorithm, instead of having one Q-function, we have two different Q-functions. One of them is used for decision making (selecting actions), while the other one is used to estimate the value (evaluating actions) [2,31]. The use of two action-value functions is important to stabilize training and to avoid overestimation. A DDQN pseudo-code is illustrated in the following Algorithm 1:

Algorithm 1 Double Deep-Q-learning (DDQN) algorithm

Initialize primary network Q_θ, target network Q'_θ, replay buffer D, $\tau << 1$;
for each iteration **do**
  for each environment step **do**
    Observe state s_t and select $a_t \sim \pi(a_t, s_t)$;
    Execute a_t and observe next state s_{t+1} and reward $r_t = R(s_t, a_t)$;
    Store (s_t, a_t, r_t, s_{t+1}) in replay buffer D;
  end for
  for each update step **do**
    sample $e_t = (s_t, a_t, r_t, s_{t+1}) \sim D$;
    Compute target Q value:
    $Q^*(s_t, a_t) \approx r_t + \mathrm{Y}Q_\theta(s_{t+1}, \mathrm{argmax}_{a'}\, Q_\theta{}'(s_{t+1}, a'))$
    Perform gradient descent step on $(Q^*(s_t, a_t) - Q_\theta(s_t, a_t))^2$
    Update target network parameters:
    $\theta' \leftarrow \tau * \theta + (1 - \tau) * \theta'$
  end for
end for

3. Mass Transit Traffic Model

We propose in this section a new traffic model on a linear mass transit line (a metro line for example), where we combine a discrete-event description for the vehicles dynamics with a macroscopic model for the passenger flow dynamics. This combination is in particular described by the definition of the vehicle dwell time as a function of the passengers demand; see Equation (7). We will use the proposed traffic model as the simulation environment while solving the optimization problem.

3.1. Vehicle Dynamics

In order to describe the dynamics of vehicles, we follow [17] where the line is partitioned into segments, the length of each segment being longer than one vehicle; see Figure 1. We then use as the main variable the vehicle departure times from each segment. The vehicle dynamics notations are used as following:

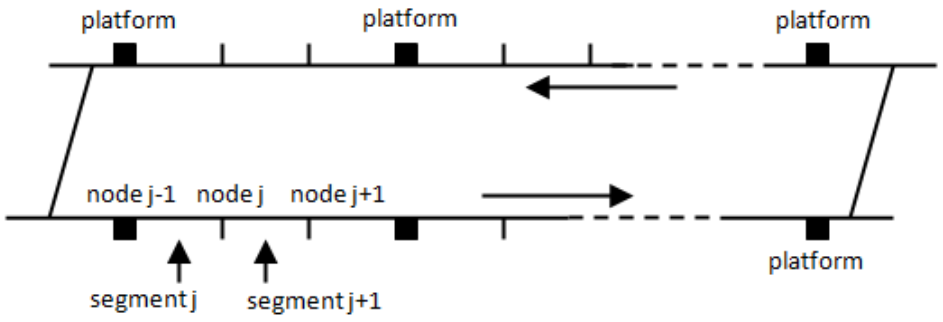

Figure 1. Representation of a mass transit line.

N Total number of platforms on the line
n Total number of segments
m Number of running trains
L Length of the whole line
b_j $\in \{0;1\}$ Boolean number of trains initially positioned on segment j (at time zero)
$\bar{b}_j$ $= 1 - b_j, \in \{0;1\}$
d_j^k Instant of the kth train departure from node j
a_j^k Instant of the kth train arrival at node j. If node j is not a platform, the train does not stop; thus, d_j^k and a_j^k are equal
r_j Train running time on segment j, i.e., from node $j-1$ to node j
w_j^k $= d_j^k - a_j^k$, train dwell time, i.e., delay time between the kth arrival at node j and kth departure from node j
t_j^k $= r_j + w_j^k$: train travel time from node $j-1$ to node j at the kth departure
g_j^k $= a_j^k - d_j^{k-1}$: Node (or station) safe separation time applied for the kth arrival at node j
h_j^k $= d_j^k - d_j^{k-1} = g_j^k + w_j^k$: Departure time headway at node j associated to the (k-1)th and kth departures from node j
s_j^k $= g_j^{k+b_j} - r_j^k$: Node safe separation time, running time excluded.

Lower and upper bounds, as well as average values of the variables on asymptotic (on j and k), were also defined for some parameters:

- $\bar{r}_j, \bar{t}_j, \bar{w}_j, \bar{g}, \bar{h}_j, and \bar{s}_j$, respectively, denote upper bounds for running, travel, dwell, safe separation times, headway and s variables.
- $\underline{r}_j, \underline{t}_j, \underline{w}_j, \underline{g}, \underline{h}_j, and \underline{s}_j$, respectively, denote lower bounds for the pre-cited variables.
- $r, t, w, g, h, and s$, respectively, denote the average values of the pre-cited variables.

The vehicle dynamics are then described [17] by the two following constraints:

- The departure time of the kth vehicle from segment j has to hold after the departure of the $(k - b_j)$th vehicle from segment $(j-1)$, plus (+) the minimum running time $\underline{r}_j$ from segment $(j-1)$ to segment j, plus (+) the dwell time w_j^k of the vehicle at segment j. We write:

$$d_j^k \geq d_{j-1}^{k-b_j} + \underline{r}_j + w_j^k, \tag{1}$$

  where the formula for w_j^k is given below in Section 3.2.
- The departure of the kth vehicle from segment j has to hold after the departure of the $(k - \bar{b}_{j+1})^{th}$ vehicle from segment $(j+1)$, plus (+) the minimum safe separation time $\underline{s}_{j+1}$ at segment $(j+1)$. We write:

$$d_j^k \geq d_{j+1}^{k-\bar{b}_{j+1}} + \underline{s}_{j+1}. \tag{2}$$

With the assumption that only the inequalities (1) and (2) constrain the vehicle dynamics, and that the vehicles depart as soon as both constraints (1) and (2) are satisfied, the vehicles dynamics are written:

$$d_j^k = \max\left\{ d_{j-1}^{k-b_j} + \underline{r}_j + w_j^k, d_{j+1}^{k-\bar{b}_{j+1}} + \underline{s}_{j+1} \right\}, \tag{3}$$

3.2. Passenger Flow Dynamics

We present in this section the dynamics of the passengers flows starting from their arrival to stations, including their access to platforms, waiting for vehicles, and boarding, and then ending when they leave the vehicles at their destinations. For that, we will use the following additional notations:

λ_{ij} Passenger arrival rate from platform i (the origin) onto train for destination platform j. If either i or j is not a platform (but just a disretization node), the rate is zero.

λ_i Average rate of passenger flow arriving to platform i. If i is not a platform, the rate is zero.

$$\lambda_i = \begin{cases} \sum_j \lambda_{ij} & \text{if node i is a platform} \\ 0 & \text{Otherwise} \end{cases}$$

$A^k_{l,j}$ The number of passengers of destination j that are willing to enter the platform l between arrivals of vehicles $k-1$ and k.

$I^k_{l,j}$ Flow of passengers (in passengers per time unit) of destination j entering platform l between arrivals of vehicles $k-1$ and k.

$Q^k_{l,j}$ Stock of passengers of destination j present at platform l at the time of the kth vehicle arrival at platform l.

$\mu^k_{l,j}$ Flow of passengers (in passengers per time unit) of destination j boarding at the time of the kth vehicle arrival at platform l.

$P^k_{i,j,l}$ Stock of passengers of origin i and destination j present at the time of the kth vehicle arrival at platform l inside the vehicle.

E^k_l Flow of passengers (in passengers per time unit) alighting at the time of the kth vehicle arrival at platform l.

K^t Passenger capacity of each vehicle (in max. number of passengers).

K^p Passenger capacity of each platform(in max. number of passengers).

γ_l Maximum entering rate of passengers to platform l (maximum value of the variable $I^k_{l,j}$ in passengers per time unit).

α Maximum boarding rate (maximum value of the variable $\mu^k_{l,j}$ in passengers per time unit).

β Maximum alighting rate (maximum value of the variable E^k_l in passengers per time unit).

The passenger flow dynamics are then described by the formulas of the three stock variables A, Q and P, three flow variables I, μ and E, and the vehicle dwell time at platforms w (Figure 2).

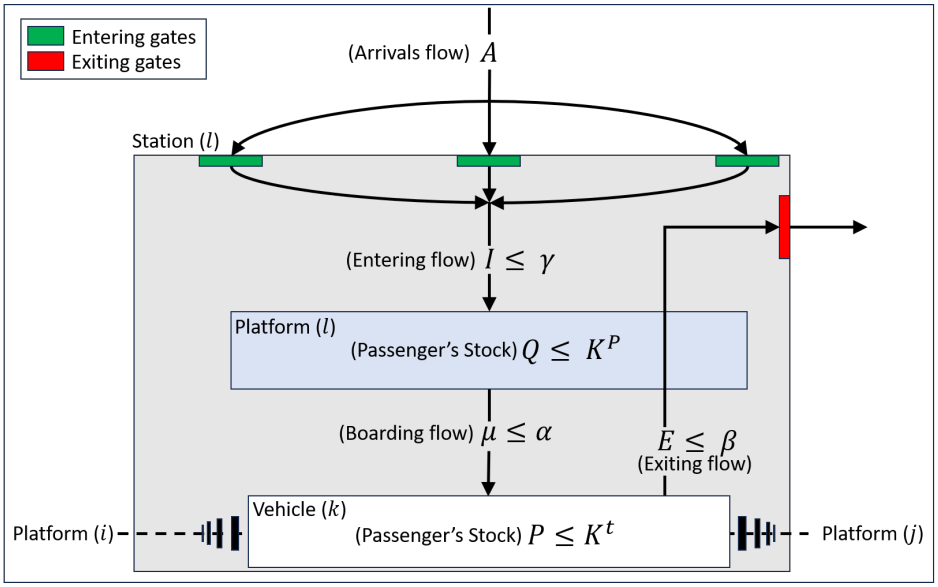

Figure 2. Illustration of the passenger flow dynamics.

3.2.1. Passenger Arrivals A

We assume that passengers arrive with a fixed arrival rate λ (passengers by time unit), corresponding to a fixed period of time, e.g., peak hour. Then, since we control the entry to the platforms, we consider the stock of passengers A who already arrived to the station but do not have access to the platform. The stock is updated by adding the newly arrived passengers during the time period $(a_l^k - a_l^{k-1})$ and by substracting the passengers who have already had access to the platform:

$$\begin{aligned} A_{lj}^1 &= \lambda_{ij} a_l^1, \\ A_{lj}^k &= A_{lj}^{k-1} + \lambda_{ij}(a_l^k - a_l^{k-1}) - I_{ij}^{k-1} \end{aligned} \tag{4}$$

3.2.2. Passenger Inflows I to the Platforms

The flow of passengers entering platform l is limited by three terms: (1) the number of passengers willing to enter the platform, (2) the flow rate multiplied by the time between the $(k-1)$th and kth vehicle arrivals at platform l, and (3) the space available at platform l. The available capacity is then distributed to all the flows I_{lj}^k according to their destinations j by respecting their distributions at the arrival A_{lj}^k, which justifies the appearance of terms $A_{l,j}^k / \sum_s A_{l,s}^k$ in Formula (5).

$$I_{lj}^k = \min \begin{cases} A_{l,j}^k \\ \gamma_l \frac{A_{l,j}^k}{\sum_s A_{l,s}^k} \left(a_l^k - a_l^{k-1} \right) \\ \max \left[0, \frac{A_{l,j}^k}{\sum_s A_{l,s}^k} \left(K_{opt}^p - \sum_{j>l} \left(Q_{lj}^{k-1} - \mu_{lj}^{k-1} \right) \right) \right]. \end{cases} \tag{5}$$

3.2.3. Stock of Passengers Q at Platforms

The number of passengers at platform l is updated by simply adding the newly entered passengers and removing the boarded ones onto the vehicles.

$$Q_{l,j}^k = \begin{cases} Q_{l,j}^{k-1} + I_{l,j}^k - \mu_{l,j}^{k-1} & \text{if } l \text{ is a platform} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3.2.4. Vehicle Dwell Time w

Vehicles dwell at platforms to permit passengers alighting (time required: $\sum_{s=1}^{l-1} P_{slj}^k / \beta_l^k$) and passengers boarding (required time: $\sum_{s=l+1}^{N_b} Q_{sl}^k / \alpha_l^k$). The dwell time must also respect lower and upper bound constraints. We need here to impose a minimum vehicle dwell time given by $\sum_{s=1}^{l-1} P_{slj}^k / \beta_l^k$ in such a way that we guarantee that all passengers with destination l have the possibility to alight at l. The upper bound $\bar{w}$ must be bigger than $\sum_{s=1}^{l-1} P_{slj}^k / \beta_l^k$ in order that Formula (7) makes sense.

$$w_l^k = \max \left(\frac{\sum_{s=1}^{l-1} P_{slj}^k}{\beta_l^k}, \min \left(\bar{w}, \frac{\sum_{s=1}^{l-1} P_{slj}^k}{\beta_l^k} + \frac{\sum_{s=l+1}^{N_b} Q_{sl}^k}{\alpha_l^k} \right) \right) \tag{7}$$

3.2.5. Passenger Boarding Flow μ

The passenger boarding flow is limited by three terms: (1) the number of passengers at platform Q, (2) the flow rate multiplied by the remaining time from the train dwell time (train dwell time minus $(-)$ the time required for passengers alighting), and (3) the space available inside the train. The available capacity is then distributed to all the flows μ_{lj}^k according to their destinations j by respecting their distributions at the platform Q_{lj}^k, which justifies the appearance of terms $Q_{l,j}^k / \sum_s Q_{l,s}^k$ in Formula (8).

$$\mu_{l,j}^{k} = \min\left[Q_{l,j}^{k}, \alpha \frac{Q_{l,j}^{k}}{\sum_s Q_{l,s}^{k}}\left(w_l^k - \frac{1}{\beta}\sum_i P_{i,l,l)}^{k}\right), \frac{Q_{l,j}^{k}}{\sum_s Q_{l,s}^{k}}\left(K^t - \sum_{j>l} P_{i,j,l}^{k}\right)\right] \quad (8)$$

3.2.6. Stock of Passenger P inside the Vehicles

The number of passengers with origin i and destination j inside a vehicle at the time of the kth departure of the vehicle from platform l is simply given by the passenger boarding flow from platform i of passengers with destination j.

$$P_{ij,l}^{k} = \begin{cases} \mu_{i,j}^{k-\sum_{p=i+1}^{l} b_p} & \text{if } i < l \leq j \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

3.2.7. Passenger Exits from Vehicles E

The number of passengers alighting from a vehicle at platform l is simply given by the number of passengers at the vehicle with any origin and with destination l. Let us note that we lower-bound the vehicle dwell time at every platform l in order to let all the passengers with destination l alight at l; see Formula (7).

$$E_l^k = \sum_i P_{ill}^k \quad (10)$$

4. DDQN Model for Traffic Control in Mass Transit

As mentioned above, we propose in this work a DDQN (double deep Q-learning) algorithm for traffic optimization in a mass transit line. We give in this section all the details about the state representation (Section 4.1), agent actions (Section 4.2), reward function (Section 4.3) as well as neural network architecture and DDQN algorithm implementation (Section 4.10).

4.1. State Representation

The choice of state representation variables is determinant in the optimization process, since the agent takes its actions based on the observation of environment state, which is given by the state representation. In order to summarize correctly and accurately the state of the traffic on a transit line, the state is represented as a matrix ($n = 7 \times m = 2$), where ($n = 7$) is the number of state variables and ($m = 2$) represents the mean and standard deviation of each of the following variables:

- Flow of passengers entering platforms (matrix I).
- Number of passengers inside the vehicles (family of matrices P).
- Number of passengers on the platforms (matrix Q).
- Flow of passengers boarding on vehicles (matrix μ).
- Vehicle time headway at each station (vector h).
- Number of passengers willing to enter to stations (matrix A).
- Vehicle dwell time at each station (vector w).

4.2. Agent Actions

In order to optimize the traffic on the transit line, the agent needs to have a set of actions (decision variables) which is sufficiently large to master the whole system and to ensure its controllability. We propose in this work a set of possible actions to be taken during the training process, which are composed of three changes either (increase, stay the same, or decrease) for each of the four control variables (number of vehicles, vehicle's speed upper-bound, vehicle dwell time, and passengers entering rate to platforms.) Therefore, the agent is able to act on the variables by increasing, decreasing or keeping the same values of the following variables:

- Number m of running vehicles.
- Maximum vehicle speed or equivalently minimum vehicle running time $\underline{r}$.
- Maximum vehicle dwell time at stations $\bar{w}$.
- Maximum passengers entering rate to platforms γ.

Hence, the total number of possible actions of DDQN output would be 3^4, which is 81 action (three being the number of possible actions on a variable, four being the number of variables). Each of these actions is itself a multi-action. In other words, the output of DDQN is interpreted as a number from 0 to 80, which is represented in a 3-base system. As an example of an action (action = 16) which is represented in a 3-base system as $(0121)_3$, this action is interpreted from left to right as: (0) decrease number of metros, (1) keep the maximum vehicle speed the same, (2) increase maximum vehicle dwell time, (1) keep the maximum passengers rate entering the platform the same. Table 1 below sets the minimum and maximum values of the action variables used in the case study of Metro Line 1 Paris.

Table 1. Minimum and maximum values of action variables.

Variable Name	Min. Value	Max. Value
number of veh.	20	148
max. speed (meter/second)	5	22
max. dwell time (second)	16	45
inflow to plat. (passengers/second)	1	80

4.3. Reward Function

The reward function plays a crucial role in evaluating the actions taken by the agent and guiding the learning process toward optimal actions and policies. In this work, we propose a reward function that considers both metro operator and passenger criteria. On the operator side, we assign a cost to running vehicles and track the overall number of served passengers. Hence, our objective is to minimize the number of running vehicles (m) and maximize the number of passenger exits (E). On the passenger side, we consider the criteria of time, which includes the vehicle time headway or frequency as well as the comfort of passengers on the trains and platforms. We explicitly outline the functions used for all the components of the reward below.

4.4. Passenger Comfort inside the Trains

The passengers' comfort inside vehicles is generally measured by the number of passengers per square meter inside the vehicle. For example, in Paris, the RATP Group (the operator of Paris metro lines) considers that up to four passengers per square meter is comfortable. Another way to define the passengers' comfort in a vehicle is to consider the ratio $P/P_{\max}$ between the number of passengers in a given area and the maximum number of passengers in the same area.

The objective here is not to minimize or maximize $P/P_{\max}$ but rather to approach its ideal value $P^*/P_{\max}$, where P^* is the number of passengers corresponding to the ideal value of the passengers' comfort inside vehicles. Therefore, we propose the following reward term for the passengers' comfort inside vehicles:

$$R_1 = -\left(\frac{P}{P_{\max}} - \frac{P^*}{P_{\max}}\right)^2. \tag{11}$$

4.5. Passenger Comfort at the Platforms

The comfort index of passengers at platforms is defined as $Q/Q_{\max}$ (in a similar way as for the passengers comfort inside vehicles), where Q and $Q_{\max}$ are the number and maximum number, respectively, of the passengers at a platform. The objective here is to

maintain the passengers' density at platforms under a given threshold in order to ease the passengers boarding and alighting. Therefore, we propose the following reward term for the passengers' comfort at platforms:

$$R_2 = \left(\frac{Q_{\text{accept}}}{Q_{\max}} - \frac{Q}{Q_{\max}} \right), \tag{12}$$

where Q_{accept} is the given threshold beneath which the passengers' density at the platform is acceptable.

4.6. Vehicle Time Headway

Vehicle time headway is very important for the efficiency of a transit line. It is the inverse of the vehicle frequency which gives the number of vehicles per time unit and which is clearly related to the number of passengers served by the time unit. This index is important for passengers, as it is directly related to the average waiting time at platforms. It is also important for operators, since they are generally engaged with the transport authorities to realize predefined vehicle frequencies and satisfy given levels of passenger demand. The objective here is to minimize the vehicle time headway for both passengers' and operators' point of views. Therefore, we propose the following reward term for the vehicle time headway:

$$R_3 = \left(\frac{h_{\text{acc}}}{h_{\max}} - \frac{h}{h_{\max}} \right), \tag{13}$$

where h_{acc} is an acceptable vehicle time headway value, and $h_{\max}$ is a maximum value for the vehicle time headway.

4.7. Number of Exiting (Served) Passengers

The objective here is to optimize the number of passengers who reach their destination (station). We propose the following reward term:

$$R_4 = \frac{E}{\beta}. \tag{14}$$

4.8. Number of Operating Vehicles

Our objective is to minimize the number of operating vehicles to reduce operation costs for the operator. However, we observe that this reward term conflicts with the reward terms mentioned earlier. Maximizing passengers' comfort inside the vehicles and at platforms, maximizing the number of served passengers, and minimizing vehicle time headway require an increase in the number of operating vehicles. As a result, there is a trade-off between minimizing operation costs and maximizing passengers' satisfaction. We propose here the following reward term for the number of operating vehicles:

$$R_5 = -\frac{m}{m_{\max}}, \tag{15}$$

where $m_{\max}$ is the maximum number of vehicles that can operate on the line, which is given generally by the total number of segments.

4.9. The Overall Reward Function

The overall reward combines linearly the five reward terms defined above:

$$R = -a_1 \left(\frac{P}{P_{\max}} - \frac{P^*}{P_{\max}} \right)^2 + a_2 \left(\frac{Q_{\text{accept}}}{Q_{\max}} - \frac{Q}{Q_{\max}} \right) + a_3 \left(\frac{h_{\text{acc}}}{h_{\max}} - \frac{h}{h_{\max}} \right) + a_4 \frac{E}{\beta} - a_5 \frac{m}{m_{\max}}, \tag{16}$$

where a_1, a_2, a_3, a_4 and a_5 are positive weighting parameters.

4.10. NN Architecture & DDQN Algorithm Implementation

To train the controller (agent), we use an RL model implemented through a neural network with four intermediate layers comprising a total of 1264 neurons. The rectified linear unit (RELU) activation function is used in all layers except the last layer, which employs the softmax activation function as it represents the 81 output actions. We use the mean-squared-error loss function along with the Adam optimizer, which is a stochastic gradient descent method. The DDQN training algorithm's key tuned variables are as follows: discount rate: 0.85, exploration rate starting at 1 and decreasing by a factor of 0.99 at each step, learning rate: 0.001, number of episodes: 50, and a maximum of 100 steps per episode. Figure 3 illustrates the training algorithm.

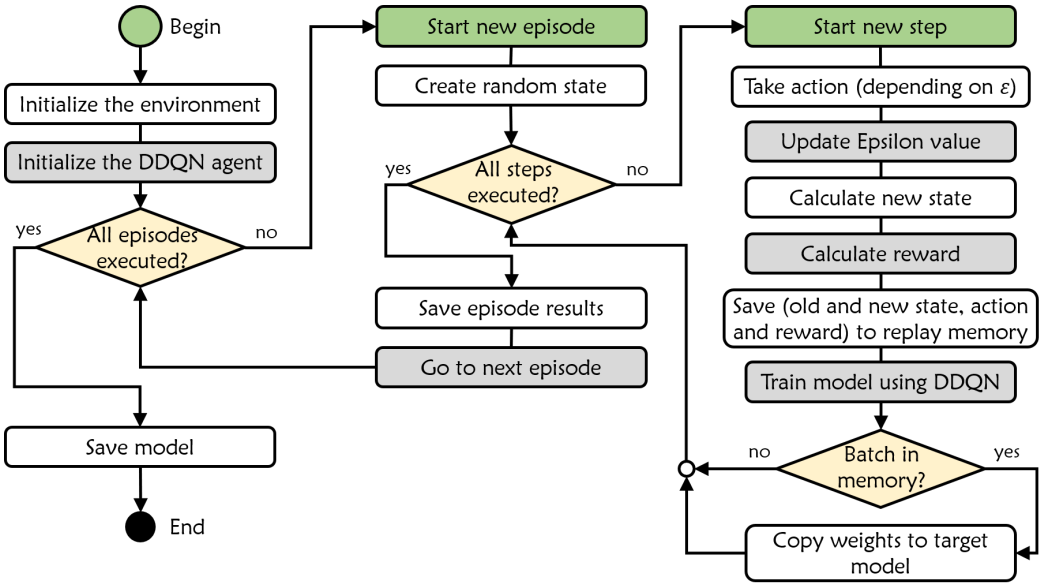

Figure 3. Training algorithm.

5. Case Study: Metro Line 1 Paris

In order to illustrate our approach, we take as a case study Metro Line 1 Paris. It is one of the oldest and busiest metro lines in the city, connecting the western suburbs to the heart of Paris. It was opened in 1900, making it the first metro line in the city. Line 1 starts at the La Defense business district and passes through popular tourist destinations such as the Champs-Elysees, the Louvre Museum, and the Bastille area before ending at Chateau de Vincennes. The line is automated, has 25 stations, and runs underground, making it a convenient mode of transportation for both locals and tourists [15]. With its frequent service and convenient stops, the Paris Metro Line 1 is an essential part of the city's transportation system. The line proposed several connections with other metro lines and modes of transportation (RER A suburban train line, several metro lines (6, 9, etc.), Transilien suburban train lines, the T2 tramway, etc). These connections provide passengers with easy access to other parts of Paris and the surrounding suburbs. Furthermore, Metro Line 1 in Paris stands out as one of the busiest metro lines, grappling with high daily passenger volumes and capacity limitations, particularly during peak hours. To address these challenges, effective optimization strategies are essential to manage passenger inflows, alleviate congestion, and elevate the overall service quality. The line's linear alignment offers advantages in terms of analyzing and optimizing passenger flows along its route. Furthermore, Metro Line 1 is an automatic line that relies on mathematical models rather than AI techniques. Studying this metro line allows for the application of our approach in real-life scenarios, considering factors such as controlling the opening and closing of metro doors and accounting for various traffic variables such as dwell time and others. By examining and optimizing an existing system with defined parameters, we can establish a

practical foundation for future implementations, ensuring the feasibility and applicability of our approach in real-world transit networks.

In Figure 4, we give the scheme of the Metro Line 1 Paris. In Table 2, we give the values taken for different parameters of the case study: Metro Line 1 Paris. In Figure 5, we illustrate the passengers' flow origin–destination (OD) matrix considered for Metro Line 1 stations in Paris. The purpose of this matrix is to illustrate the passengers' entering flow from an origin (represented in columns) to various destinations (represented in rows). Each cell in the matrix represents the flow of passengers from an origin station to a destination one. For instance, the cell colored in aquamarine, located in the third row and fifth column, indicates that the passenger flow from station "Nation" to station "Saint-Mandé" is approximately 0.45 passengers per second. Additionally, the matrix highlights a specific pattern where the diagonal row represents the flow from a station to itself: for example, the flow from station "Nation" to the same station "Nation". These diagonal cells are colored in dark blue and have a value of zero, indicating that there is no passenger from the station to itself.

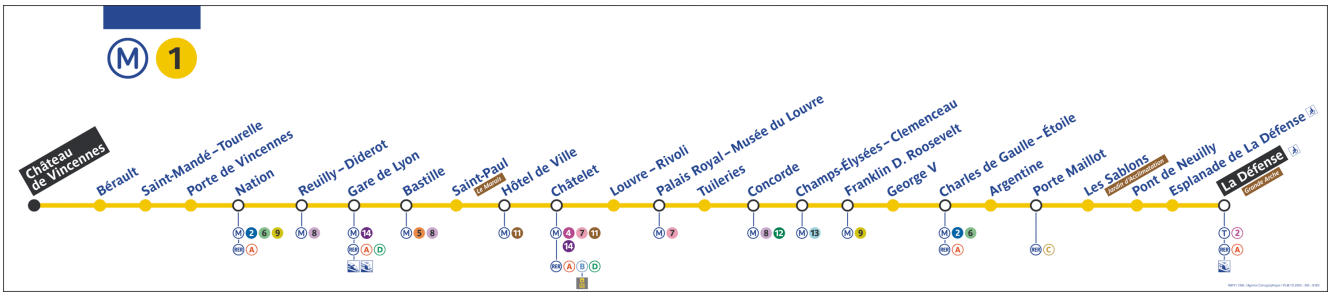

Figure 4. Metro Line 1 Paris [15].

Table 2. Parameters' values used in the case study of Metro Line 1 Paris.

Passengers capacity of vehicles (P_{max})	700 passengers
Ideal number of pass. in vehicles (P^*)	525 passengers
Max. passengers density in vehicles	4 pass./m^2
Maximum number of trains (m_{max})	148
Maximum acceptable veh. time headway (h_{max})	1000 s
Acceptable average veh. time headway (h_{accept})	600 s
Area of platforms	270 m^2
Max. passengers density at platforms	4 pass./m^2
Passengers capacity of platforms (Q_{max})	1080 passengers
Acceptable number of passengers at platforms (Q_{accept})	540
a_1, a_2, a_3, a_4, a_5	32, 10, 0.25, 4, 1.

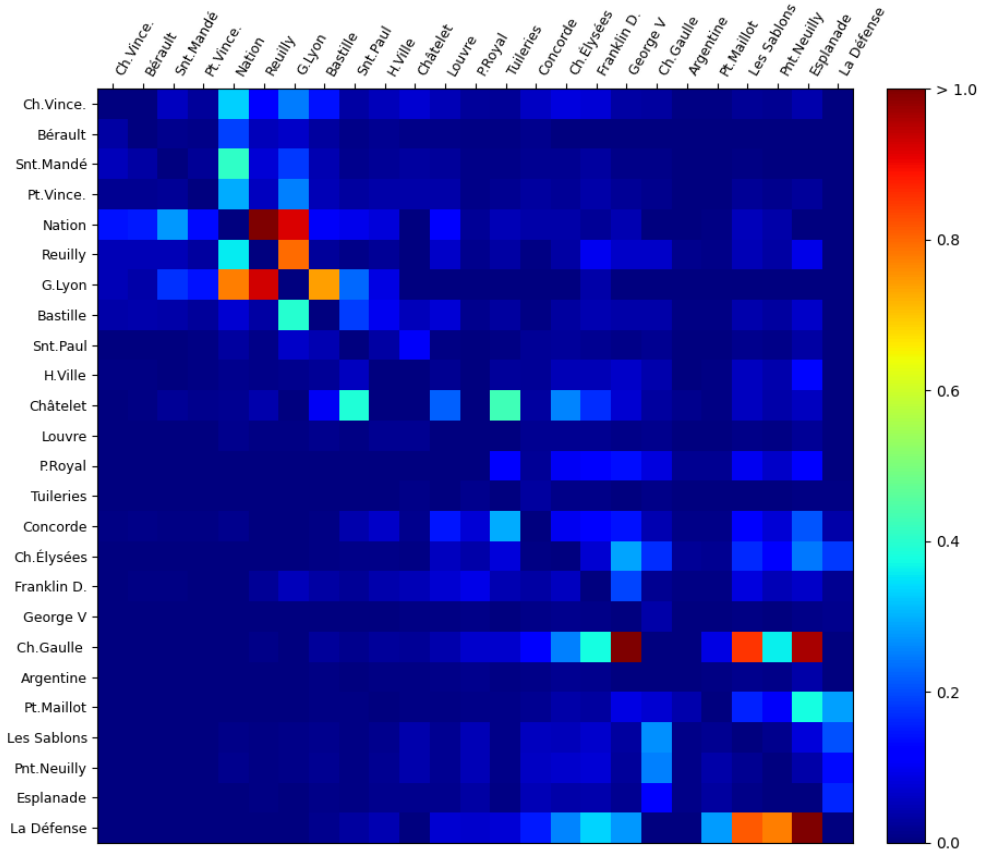

Figure 5. Origin–destination (OD) matrix for passenger flows (passenger/second) in Metro Line 1 stations.

5.1. The Results of Optimization

The training process lasted generally for more than 48 h, as it is illustrated in Table 3, where the training was conducted on two different normal-performance personal computers. The minimum training time was about 36 h, with a mean time of 2 min per step. Each step is simulated for a loop of vehicle circulations through the whole dedicated rail network as defined by the mathematical model. During each training process, the agent was trained to make predictive optimal decisions. We used the passengers' demand profile, which includes the origin–destination matrix depicted in Figure 5 to optimize the system. The optimization process considered three levels of passengers' demand: the actual level, a high level that is twice the actual level, and an ultra-high level that is ten times the actual level.

Table 3. Resources utilization for the optimization approach.

		PCs	
		PC 1	**PC 2**
Processor		Intel Core i7 (8 cores)	Intel Core i5 (4 cores)
PC's RAM		16 GB	8 GB
Utilized RAM		4.3 GB (Python 2.5 GB and Octave 1.8 GB)	
Approximative Runtime	$\lambda = 2$	36 h	60 h
	$\lambda = 4$	45 h	76 h
	$\lambda = 10$	62 h	90 h

Table 4 presents the optimized values obtained by the DDQN algorithm after training for the three passenger demand levels considered. For the actual level of passengers'

demand, the optimal number of vehicles is 35, with an average time headway of 165 s (approximately 22 vehicles per hour) and an average dwell time of 20 s. The inflow (I) of passengers entering platforms is limited to 68 passengers per second (244,800 passengers per hour), resulting in a stock of 100 delayed passengers outside the platforms. The average number of passengers at platforms and inside the trains are 380 (out of 1080) and 85 (out of 700), respectively, indicating a high level of comfort for passengers.

Table 4. Optimal values obtained by the RL algorithm for three levels of passengers' demand ($\lambda = 2$ actual level, $\lambda = 4$ high level, and $\lambda = 10$ ultra-high level).

	m	w (s.)	h (s.)	A (p.)	I (p./s.)	Q (p.)	μ (p./s.)	P (p.)
$\lambda = 2$	35	20	165	100	68	85	68	380
$\lambda = 4$	48	24	155	250	95	145	95	420
$\lambda = 10$	55	28	150	2500	115	240	115	460

To accommodate twice the level of passengers' demand, the train frequency needs to be increased by decreasing the train time headway. The DDQN algorithm suggests a time headway of 155 s (approximately 23 vehicles per hour) instead of 165 s. However, in order to achieve this frequency, more trains are required with the DDQN algorithm recommending 48 trains instead of 35 (an additional 13 trains). Additionally, train dwell times need to be increased to accommodate the increased flow of alighting and boarding passengers. As a result, the inflow (I) of entering passengers to platforms is now limited to 95 passengers per second (342,000 passengers per hour) instead of 68 passengers per second. This is completed to minimize the stock of delayed passengers outside the platforms, which increases to 250 passengers instead of 100 passengers. Despite these changes, the average number of passengers at platforms and inside the trains remain comfortable, with 420 passengers out of 1080 at platforms and 145 passengers out of 700 inside the trains. The DDQN algorithm demonstrates that it is feasible to serve ten times the actual level of passengers while maintaining acceptable passenger comfort by limiting the inflow of passengers to the platform to 115 passengers per second (414,000 pass./h). However, this leads to a greater number of passengers being delayed outside the platform, resulting in an average stock of 2500 passengers outside the platforms. In order to achieve a 5 s improvement in the train time headway (150 s. (24 veh./h) instead of 155 s.), an additional seven trains are required compared to the case of $\lambda = 2$. As a result, the average train dwell time increases to 28 s. Despite these changes, the average number of passengers at platforms and inside the trains remains comfortable at 460 (/1080) and 240 (/700), respectively.

Generally, the RL agent was capable of moving from randomness to prediction in a way that the relationship between states was logically correlated. As well, the agent was flexible when it came to dealing with new situations such as increasing the passenger's demand level. In other words, the agent tried during all previous scenarios to obtain the maximum reward.

In Figure 6, we show the result of optimization with the actual level of passenger demand. This figure is obtained by running the DDQN at the final episode. The figure shows the action of increasing, decreasing or keeping unchanged the number of running trains on the metro line (green, red or black colors, respectively), as a function of the average number P of passengers at platforms and the average inflow I of passengers to platforms.

We can first see that when the average number P of passengers at platforms increases, the DDQN algorithm responds by limiting the inflow I of passengers to platforms. Second, as the average number P of passengers at platforms is beneath the ideal number P^* of passengers in vehicles (for passengers' comfort, about 525 passengers in the figure), the optimization recommends to decrease the number m of running vehicles. When the average number P of passengers at platforms is beyond the ideal number P^* of passengers in vehicles, the optimization recommends to increase the number m of running vehicles

in order to improve the passengers' comfort inside vehicles. Only one case of keeping unchanged the number m of running vehicles is obtained (the black point in the figure). This point is obtained for about 525 passengers (P^*) and about $I = 115$ passengers per second (414,000 pass./h) of passengers' inflow to platforms.

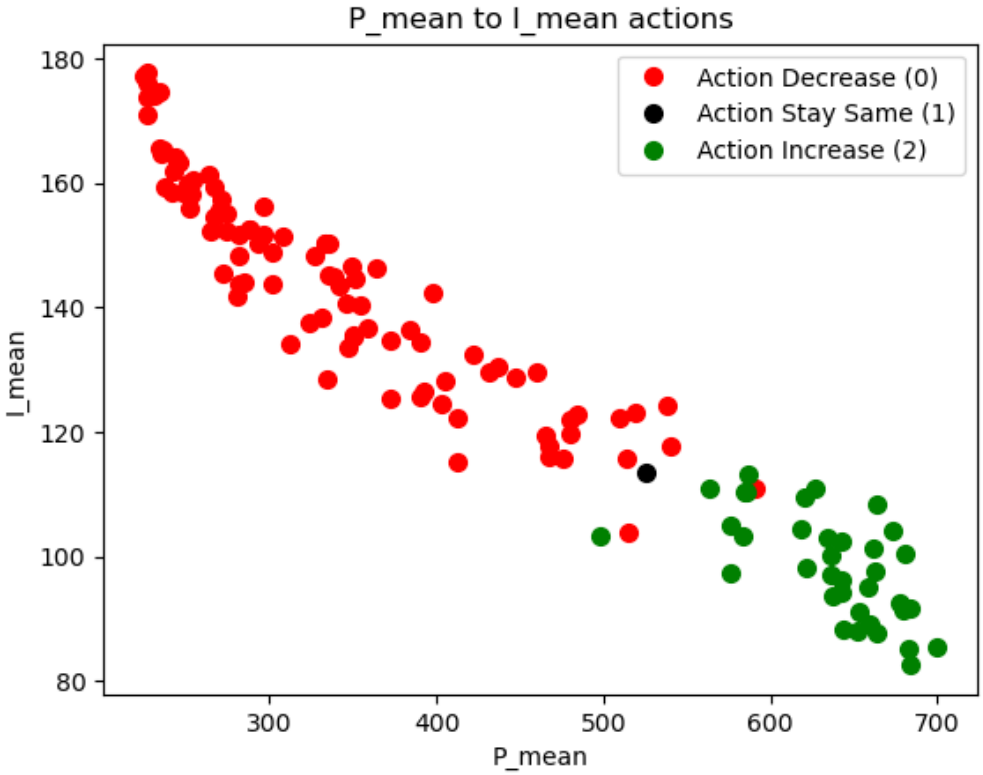

Figure 6. Controlling number of vehicles given P_mean (average number of passengers inside the metro) and I_mean (average inflow of entering passengers to platforms).

Figure 7 illustrates the results of optimization by analyzing the relationship between the variables m, A and $f = 1/h$ (vehicle frequency of the transit line). We note that m and h are both state and action variables, while A is only a state variable. We recognize in this figure the relationship between m and f; see [17]. Indeed, f starts by increasing with respect to m ($0 \leq m \leq 20$ approximately); then, it reaches the value of vehicle capacity of the line ($20 \leq m \leq 120$), and finally, it decreases again for big values of m ($m > 120$). This relationship is known under the name of the fundamental traffic diagram. We can also see that for big values of A (corresponding to big values of λ, i.e., high levels of passenger demand), the DDQN algorithm proposes only big values of m, and the vehicle frequency is decreasing with A. This can be explained by the need to increase the number of running vehicles in order to serve the high levels of passenger demand. However, as known from the fundamental traffic diagram, running big numbers of vehicles on the line implies more interactions between vehicles and induces the congestion of vehicles, which deteriorates the vehicles' frequency; see [17].

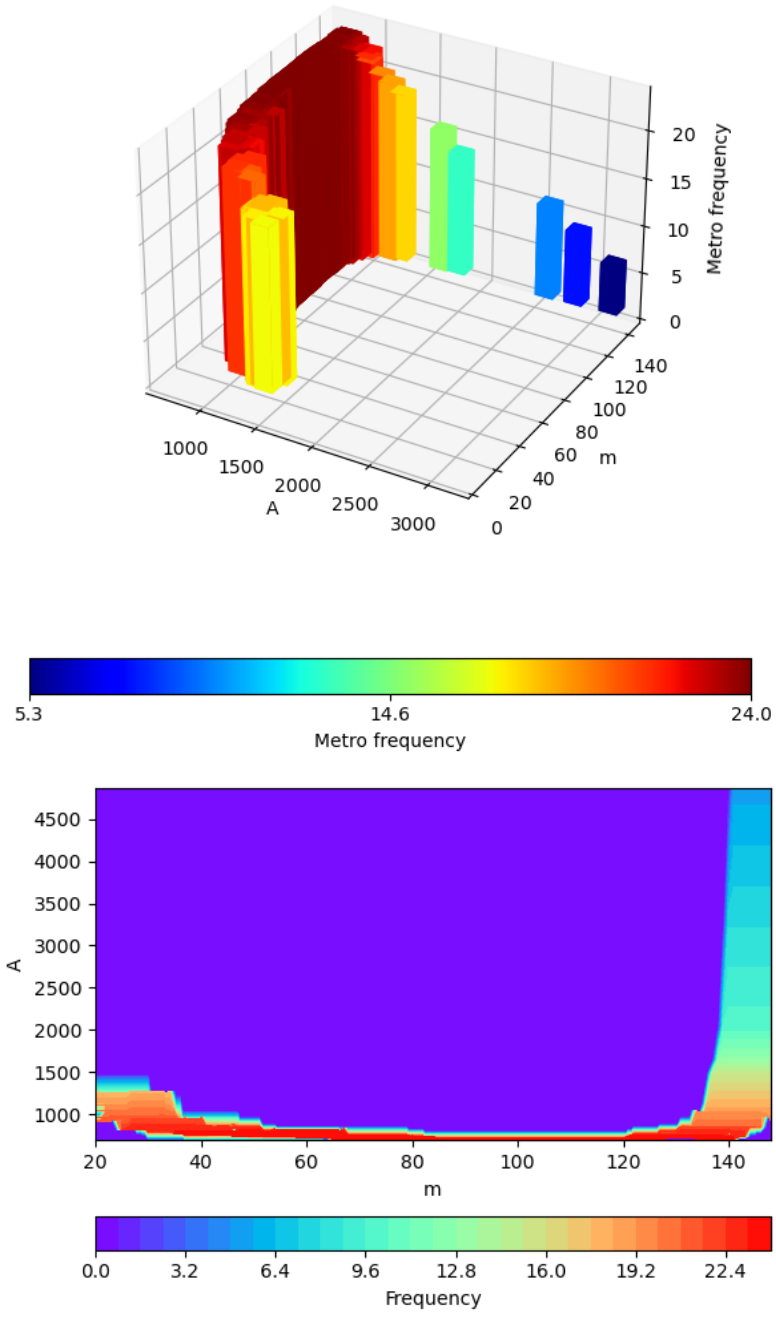

Figure 7. Top: Vehicle frequency (veh./h.) function of the number m of vehicles, and the stock A of passengers outside of the stations. Bottom: Counter of the vehicle frequency as a function of the number m of vehicles and the stock A of passengers outside of the stations.

In Figures 8 and 9, we fix the demand passenger level to $\lambda = 10$ (ten times the current level) and show what the optimization proposes as solutions by analyzing the relationships between h, m, P and Q (Figure 8) and between w, m, P and Q (Figure 9). Figure 8 shows the relationship between h, m, P and Q with the optimal policy obtained by the DDQN algorithm. The number m of running trains is varied, and for each value of m, the optimal policy obtained by the DDQN algorithm is applied, and the resulted h, P and Q are shown in this figure. Two perspectives of the same figure are shown in Figure 8. We can see that for small values of the number m of running trains ($20 \leq m \leq 40$ approximately), the number of passengers inside the trains increases with m, while the number of passengers at the platforms decreases with m. Indeed, with more trains, we take more passengers from the platforms into the trains. For the values of m corresponding to the vehicle frequency of the

line ($40 \leq m \leq 120$ approximately), we first see that the vehicle frequency is at its maximum value (the vehicle capacity of the line). The number of passengers transported inside the trains still increases, while the number of passengers at platforms seems to be stable. For big values of m, as mentioned above, a vehicle congestion occurs with a degradation of the frequency of the line. In this case, the number of passengers inside the trains continues increasing, but slightly, while the number of passengers at platforms explodes, as the train frequency is very low.

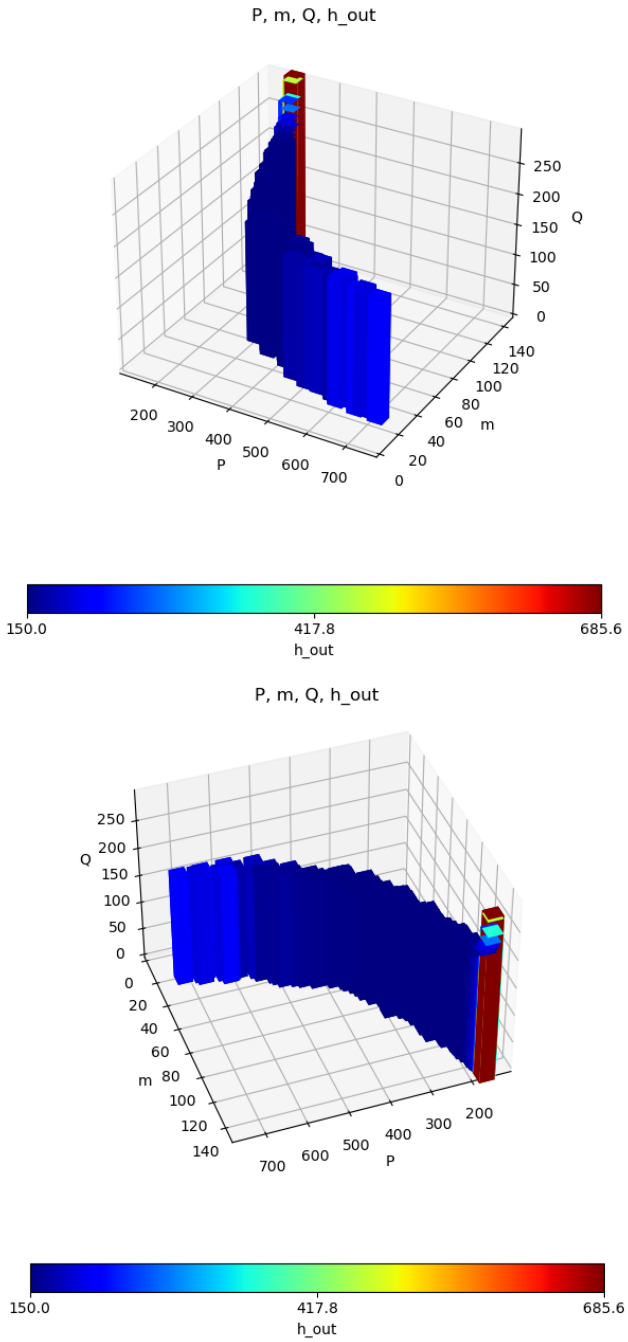

Figure 8. Two perspectives of the relationship of the vehicle time headway (h_{out}) in color as a function of the three variables: number of vehicles (m), number of passengers inside the vehicles (P) and number of passengers waiting on the platforms (Q), with an ultra-high-level demand.

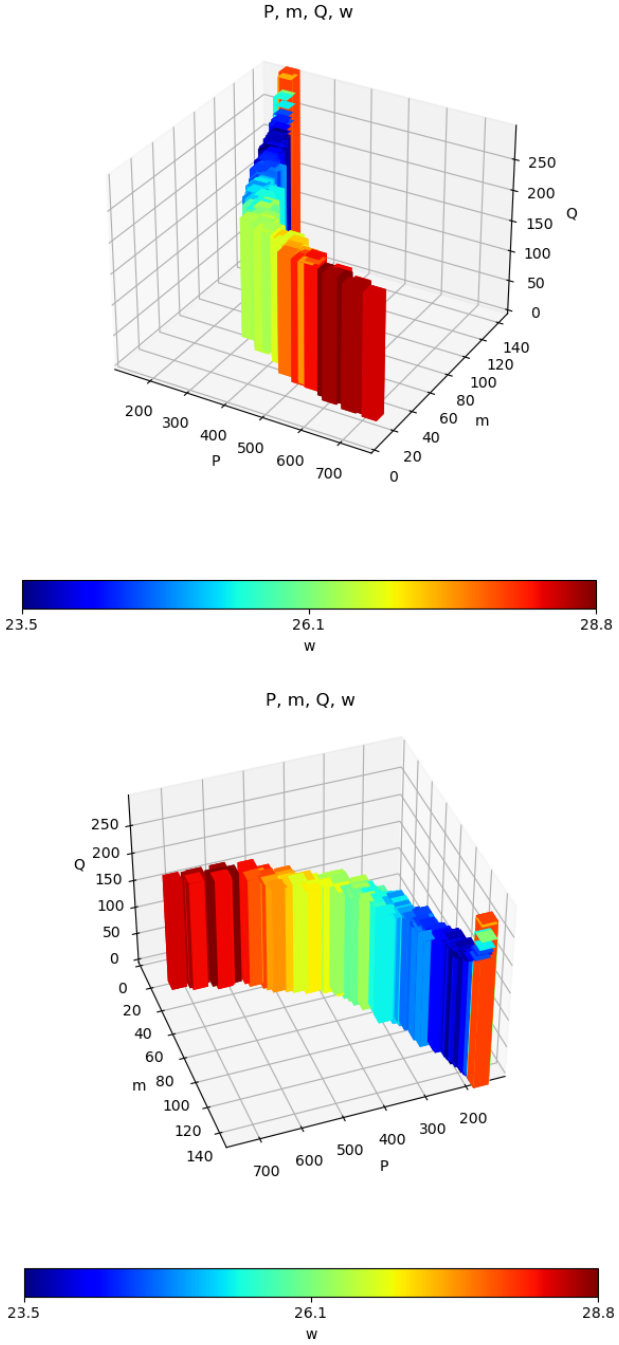

Figure 9. Two perspectives of the relationship of the dwell time (w) in color as a function of the three variables: number of vehicles (m), number of passengers inside the vehicles (P) and number of passengers waiting on the platforms (Q) with an ultra-high-level of demand.

Figure 9 shows the relationship between w, m, P and Q with the optimal policy obtained by the DDQN algorithm. As in Figure 8, the number m of running trains is varied, and for each value of m, the optimal policy obtained by the DDQN algorithm is applied, and the resulting w, P and Q are shown in this figure. We provide two perspectives of the same figure in Figure 9 in order to better show the relationship between w, m, P and Q. It is obvious that the relationships between m and P and Q are the same as in Figure 8. The first remark on the train dwell time w is that it is limited by the DDQN algorithm. Indeed, large dwell times imply mechanically a large time headway h which corresponds to low train frequencies. As we consider here a very high level of passenger demand, we need to have

high train frequencies in order to be able to serve the passengers demand. Second, we can see that w increases with m for low values of m (before the vehicle frequency $20 \leq m \leq 40$). Then, once the vehicle frequency is attained, contrary to the time headway h which remains stable during this phase ($40 \leq m \leq 120$), the vehicle dwell time w continues increasing with m. This permits taking more passengers with the same train frequency. Finally, for high values of $m \geq 120$, as mentioned above, vehicle congestion occurs.

As a conclusion from Figures 8 and 9, we can see that the DDQN algorithm is able to limit both the train time headway h and the train dwell time w under the objective of being able to serve a high level of passenger demand.

5.2. Practical Implementation

One of the major contributions of this work is the development of an algorithm that optimizes various factors, including the number of vehicles, vehicle frequency, passenger comfort, and inflow of passengers to platforms. The practical applicability of the algorithm can significantly increase passenger satisfaction while minimizing costs. To achieve effective control of passenger inflow and comfort in real-life situations, it is recommended to consider the following practical suggestions:

- Using cameras to monitor passenger behavior inside the vehicles and at platforms for accurate estimation of the environment state.
- Controlling passenger access gates by activating/deactivating/inverting escalators, opening/closing doors, and other such measures as control actions.

By implementing these recommendations in practice, the proposed algorithm can be efficiently utilized to optimize the public transportation system, leading to better passenger experience and reduced costs.

6. Conclusions and Perspectives

In this article, we presented a reinforcement learning model for optimizing vehicle and passenger traffic in a mass transit line. The model is based on a realistic mathematical description of the dynamics of vehicles and passenger flows in mass transit systems. We used a double deep-Q learning (DDQN) algorithm with multiple actions to optimize the control of traffic variables such as vehicle speed, dwell time, number of running vehicles, and passenger inflow to platforms.

Our approach is unique in that it includes the optimization of passenger inflows to platforms using control actions proposed by the DDQN algorithm, which includes optimizing the number of running vehicles, vehicle frequency, and passenger comfort. We conducted a case study on the Paris Metro Line 1, where the DDQN algorithm learned to minimize the number of passengers at platforms, increase their comfort, maintain a low train headway time, and serve the maximum number of passengers to reach their destination. Our optimized control of passenger inflows is achieved by adjusting the gate's capacity, which improves passenger satisfaction and guarantees their safety. Indeed, adjusting the gate's capacity permits controlling the number of passengers at platforms, which avoids congestion in passengers boarding and alighting, then limits the vehicle dwell times, and finally maximizes the vehicles' frequency. So, at the end, this is beneficial for the overall flow of passengers, even though some of them may observe some delay because of the passenger inflow limit. On the other side, adjusting the gate's capacity permits improving the passengers' comfort. The approach is scalable and can handle high levels of passenger demand. The agent was able to improve its service, increasing the flow of passengers boarding onto the vehicles and thus decreasing waiting passengers at platforms and outside the platforms, even under high passenger demand scenarios.

The proposed approach offers an efficient and effective method for optimizing mass transit systems with a focus on minimizing costs and enhancing passenger satisfaction. However, it is important to acknowledge certain limitations associated with the study's scope, as it solely considers a simple circular metro line without junction. Looking ahead, the future prospects of this work involve exploring more complex mass transit networks

that encompass multiple lines, transfer stations, and interconnections. This can be achieved by extending the reinforcement learning and mathematical models to larger-scale systems and diverse network typologies. Additionally, the execution time presented in Table 3 could be further improved to hope for a successful extension of the approach to mass transit networks. Furthermore, integrating real-time data from passenger flow sensors, fare collection systems, and mobile applications presents an opportunity to enhance mass transit operations. Researchers can investigate methods to effectively incorporate these data sources into the reinforcement learning model, leading to improved decision making and more accurate predictions.

The optimization problem at hand remains an open challenge that can be approached from various perspectives and optimized using different methodologies. Researchers can consider different objectives or incorporate additional input factors to optimize passenger inflows. For instance, predicting passenger demand patterns or considering real-time congestion levels can provide valuable insights for achieving efficient mass transit operations. Exploring these avenues can pave the way for further advancements in optimizing passenger flows and enhancing the overall performance of mass transit systems.

Author Contributions: Conceptualization, N.F.; methodology, N.F; software, S.K.; validation, S.K. and N.F.; formal analysis, S.K; investigation, S.K.; resources, S.K.; data curation, S.K.; writing—original draft preparation, S.K.; writing—review and editing, N.F.; visualization, S.K.; supervision, N.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiang, Z.; Fan, W.; Liu, W.; Zhu, B.; Gu, J. Reinforcement learning approach for coordinated passenger inflow control of urban rail transit in peak hours. *Transp. Res. Part Emerg. Technol.* **2018**, *88*, 1–16. [CrossRef]
2. Alesiani, F.; Gkiotsalitis, K. Reinforcement learning-based bus holding for high-frequency services. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3162–3168.
3. Ying, C.S.; Chow, A.H.; Wang, Y.H.; Chin, K.S. Adaptive metro service schedule and train composition with a proximal policy optimization approach based on deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6895–6906. [CrossRef]
4. Zhu, Y.; Goverde, R.M. Dynamic and robust timetable rescheduling for uncertain railway disruptions. *J. Rail Transp. Plan. Manag.* **2020**, *15*, 100196. [CrossRef]
5. Šemrov, D.; Marsetič, R.; Žura, M.; Todorovski, L.; Srdic, A. Reinforcement learning approach for train rescheduling on a single-track railway. *Transp. Res. Part Methodol.* **2016**, *86*, 250–267. [CrossRef]
6. Wang, Z.; Pan, Z.; Chen, S.; Ji, S.; Yi, X.; Zhang, J.; Wang, J.; Gong, Z.; Li, T.; Zheng, Y. Shortening passengers' travel time: A dynamic metro train scheduling approach using deep reinforcement learning. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5282–5295. [CrossRef]
7. Kolat, M.; Kővári, B.; Bécsi, T.; Aradi, S. Multi-agent reinforcement learning for traffic signal control: A cooperative approach. *Sustainability* **2023**, *15*, 3479. [CrossRef]
8. Wang, J.; Sun, L. Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework. *Transp. Res. Part Emerg. Technol.* **2020**, *116*, 102661. [CrossRef]
9. Liao, J.; Yang, G.; Zhang, S.; Zhang, F.; Gong, C. A deep reinforcement learning approach for the energy-aimed train timetable rescheduling problem under disturbances. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 3096–3109. [CrossRef]
10. Yan, H.; Cui, Z.; Chen, X.; Ma, X. Distributed Multiagent Deep Reinforcement Learning for Multiline Dynamic Bus Timetable Optimization. *IEEE Trans. Ind. Inform.* **2022**, *19*, 469–479. [CrossRef]
11. Liu, Y.; Tang, T.; Yue, L.; Xun, J.; Guo, H. An intelligent train regulation algorithm for metro using deep reinforcement learning. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November, 2018; pp. 1208–1213.
12. Krasemann, J.T. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transp. Res. Part Emerg. Technol.* **2012**, *20*, 62–78. [CrossRef]

13. Obara, M.; Kashiyama, T.; Sekimoto, Y. Deep reinforcement learning approach for train rescheduling utilizing graph theory. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 4525–4533.
14. Coşkun, M.; Baggag, A.; Chawla, S. Deep reinforcement learning for traffic light optimization. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 564–571.
15. Plan MÃl'tro Ligne 1. Available online: https://www.ratp.fr/plans-lignes/metro/1 (accessed on 20 June 2022).
16. Farhi, N.; Phu, C.N.V.; Haj-Salem, H.; Lebacque, J.P. Traffic modeling and real-time control for metro lines. *arXiv* **2016**, arXiv 1604.04593.
17. Farhi, N.; Phu, C.N.V.; Haj-Salem, H.; Lebacque, J.P. Traffic modeling and real-time control for metro lines. Part I-A max-plus algebra model explaining the traffic phases of the train dynamics. In Proceedings of the American Control Conference (IEEE), Seattle, WA, USA, 24–26 May 2017.
18. Farhi, N.; Phu, C.N.V.; Haj-Salem, H.; Lebacque, J.P. Traffic modeling and real-time control for metro lines. Part II-The effect of passenger demand on the traffic phases. In Proceedings of the American Control Conference (IEEE), Seattle, WA, USA, 24–26 May 2017.
19. Schanzenbächer, F.; Farhi, N.; Christoforou, Z.; Leurent, F.; Gabriel, G. Demand-dependent supply control on a linear metro line of the RATP network. *Transp. Res. Procedia* **2019**, *41*, 491–493. [CrossRef]
20. Schanzenbächer, F.; Farhi, N.; Leurent, F.; Gabriel, G. Comprehensive passenger demand-dependent traffic control on a metro line with a junction and a derivation of the traffic phases. In Proceedings of the Transportation Research Board (TRB) Annual Meeting, Washington, DC, USA, 13–17 January 2019.
21. Schanzenbächer, F.; Farhi, N.; Leurent, F.; Gabriel, G. Real-time control of the metro train dynamics with minimization of the train time-headway variance. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Maui, HI, USA, 4–7 November 2018.
22. Schanzenbächer, F.; Farhi, N.; Leurent, F.; Gabriel, G. A discrete event traffic model explaining the traffic phases of the train dynamics on a linear metro line with demand-dependent control. In Proceedings of the American Control Conference (IEEE), Milwaukee, WI, USA, 27–29 June 2018.
23. Farhi, N. Physical Models and Control of the Train Dynamics in a Metro Line Without Junction. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 1829–1837. [CrossRef]
24. Farhi, N. A discrete-event model of the train traffic on a linear metro line. *Appl. Math. Model.* **2021**, *96*, 523–544. [CrossRef]
25. Schanzenbächer, F.; Farhi, N.; Leurent, F.; Gabriel, G. Feedback Control for Metro Lines With a Junction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2741–2750. [CrossRef]
26. Schanzenbächer, F.; Farhi, N.; Christoforou, Z.; Leurent, F.; Gabriel, G. A discrete event traffic model explaining the traffic phases of the train dynamics in a metro line with a junction. In Proceedings of the IEEE Conference on Decision and Control (CDC), Melbourne, Australia, 12–15 December 2017.
27. Schanzenbächer, F.; Farhi, N.; Leurent, F.; Gabriel, G. A discrete event traffic model for passenger demand-depenbdent train control in a metro line with a junction. In Proceedings of the ITS World Congress, Singapore, 21–25 October 2019.
28. Farrando, R.; Farhi, N.; Christoforou, Z.; Schanzenbacher, F. Traffic modeling and simulation on a mass transit line with skip-stop policy. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Rhodes, Greece, 20–23 September 2020.
29. Farrando, R.; Farhi, N.; Christoforou, Z.; Urban, A. Impact of a fifo rule on the merge of a metro line with a junction. In Proceedings of the Transportation research Board (TRB) Annual Meeting, Washington, DC, USA, 9–13 January 2022.
30. Ning, L.; Li, Y.; Zhou, M.; Song, H.; Dong, H. A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3469–3474.
31. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.