UNIVERSITY OF LIVERPOOL

Multi-agent Near Real-Time Simulation of Light Train Network Energy Sustainability Analysis

Thesis submitted in accordance with the requirements of

the University of Liverpool for the degree of

Doctor in Philosophy

By

Yida Guo

June 2021

PGR Declaration of Academic Honesty

NAME (Print)	**Yida Guo**
STUDENT NUMBER	**201113807**
SCHOOL/INSTITUTE	**The University of Liverpool**
TITLE OF WORK	**PHD Thesis**

This form should be completed by the student and appended to any piece of work that is submitted for examination. Submission by the student of the form by electronic means constitutes their confirmation of the terms of the declaration.

Students should familiarise themselves with Appendix 4 of the PGR Code of Practice: PGR Policy on Plagiarism and Dishonest Use of Data, which provides the definitions of academic malpractice and the policies and procedures that apply to the investigation of alleged incidents.

Students found to have committed academic malpractice will receive penalties in accordance with the Policy, which in the most severe cases might include termination of studies.

STUDENT DECLARATION

I confirm that:

- I have read and understood the University's PGR Policy on Plagiarism and Dishonest Use of Data.
- I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.
- I have not copied material from another source nor committed plagiarism nor fabricated, falsified or embellished data when completing the attached material.
- I have not copied material from another source, nor colluded with any other student in the preparation and production of this material.
- If an allegation of suspected academic malpractice is made, I give permission to the University to use source-matching software to ensure that the submitted material is all my own work.

SIGNATURE……Yida Guo……

DATE……28th June, 2021……

Acknowledgements

This thesis reflects my dedication to pursuing a PhD degree at the University of Liverpool. I want to thank several people who have given me great support during this period.

I would like to first express my sincere gratitude to my supervisor, Associate Professor Cheng Zhang, who has spent a great deal of time supervising me throughout my PhD and guiding me to innovate my research methods. She carefully revised every academic paper I wrote and guided me in methodology, algorithm, and research logic.

Furthermore, I am very grateful to Associate Professor Shaofeng Lu, and Mr. Chaoxian Wu. Their explorations have provided a solid foundation for my research.

I also want to emphasize my appreciation to thank Associate Professor Kyeong Soo (Joseph) Kim and Professor Hui Liu. They have provided valuable suggestions for my research.

Finally, I would like to thank my family. They encouraged me when I was struggling in my research, and they reminded me that my doctorate is significant, but enjoying the process of life is more important.

Publications

[1]. Guo Yida, Cheng Zhang. 2021 “Near Real-time Timetabling for Metro System Energy Optimization Considering Passenger Flow and Random Delays.” Journal of Rail Transport Planning & Management (Under the Second Round Review)

[2]. Guo, Yida, Cheng Zhang, Chaoxian Wu, and Shaofeng Lu. 2021. “Multiagent System – Based Near-Real-Time Trajectory and Microscopic Timetable Optimization for Rail Transit Network.” Journal of Transportation Engineering, Part A: Systems 147(2):04020153. doi: 10.1061/JTEPBS.0000473.

[3]. Guo Yida, Cheng Zhang. 2021 “Real-time Railway Transit Management Based on Multi-Agent System (MAS)”, Journal of Physics: Conference Series, 1828(1): 012045. doi: 10.1088/1742-6596/1828/1/012045.

[4]. Guo, Yida, Qinlin Wang, Cheng Zhang, and Shaofeng Lu. 2019. “Multi-Agent Systems for Energy Efficient Train and Train Station Interaction Modelling.” in International Conference on Sustainable Buildings and Structures. Routledge.

[5]. Guo, Yida, Cheng Zhang, and Shaofeng Lu. 2019. “Enhancing Sustainability of Rail Transit System by Applying Multi-Agent System.” in The 2019 ASCE International Conference on Computing in Civil Engineering.

[6]. Guo, Yida, Cheng Zhang, and Shaofeng Lu. 2019. “The Application of a Multi-Agent System in Control Systems of Rail Transit.” in The 2019 Transportation Research Board.

[7]. Guo, Yida, Cheng Zhang, and Shaofeng Lu. 2018. “A Decision-Making Approach for Semi-Decentralized Rail Transit Control System.” in IET Doctoral Forum on Biomedical Engineering.

Abstract

As an attractive transportation mode, rail transit consumes a lot of energy while transporting a large number of passengers annually. Most energy-aimed research in rail transit focuses on optimizing the train timetable and speed trajectory offline. However, some disturbances during travel will cause the train to fail to follow the offline optimized control strategy, thus invalids the offline optimization. In the typical rail transit control framework, the moving authority of trains is calculated by the zone controller based on the moving/fixed block system in the zone. The zone controller is used to ensure safety when the travel plan of trains changes due to disturbance. Safety is guaranteed during the process, but the change of travel plan leads to extra energy costs. The energy-aimed optimization problem in rail transit requires ensuring safety, pursuing punctuality with considering track slope, travel comfort, energy transferring efficiency, and speed limit, etc. The complex constraints lead to high computational pressure. Therefore, it is difficult for the regional controller to re-optimize the travel plan for all affected trains in near real-time. Multi-agent systems are widely used in many other fields, which show decent performance in solving complex problems by coordinating multiple agents.

This study proposes a multi-agent system with multiple optimization algorithms to realize energy-aimed re-optimization in rail transit under different disturbances. The system includes three types of agents, train agents, station agents and central agents. Each agent exchanges information by following the time trigger mechanism (periodically) and the event trigger mechanism (occasionally). Trigger mechanism ensures that affected agents receive necessary information when interference occurs, and their embedded algorithms can achieve necessary optimization. Four types of cases

are tested, and each case has plenty of scenarios. The tested results show that the proposed system provides encouraging performance on energy savings and computational speed.

Contents

List of Abbreviation

ATC: Automatic Train Control	19
ATO: Automatic Train Operation	19
ATP: Automatic Train Protection	19
ATS: Automatic Train Supervision	19
CBI: Computer-Based Interlocking	19
CBTC: Communication-based Train Control	19
DCS: Data Communication System	19
DSU: Data Storage Unit	19
EETC: Energy-efficient Train Control	16
FBS: Fix Block System	72
GA: Genetic Algorithm	22
MAS: Multi-agent System	17
MBS: Moving Block System	18
MILP: Mixed-Integer Linear Programming	43
SCG: Scheduling and Control Group	21
TBTC: Traditional Track-based Train Control	19
ZC: Zone Controller	20

List of Figures

List of Tables

Chapter 1: Introduction

Due to the capability of transporting more passengers or goods simultaneously, the train consumes less energy per unit mass than other modes of transportation (tracks, cars, etc.) (Feitelson and Eran 1994). In addition, the punctuality and safety of the light rail transit system prescribed itself as one of the major public transportation. Light rail transit has a positive impact on increasing traffic volume, reducing traffic congestion, and other economic, social, and environmental benefits, including reducing greenhouse gas emissions and reducing dependence on automobiles, especially in urban expansion areas (Litman 2020).

Despite all these advantages, rail transit consumes a lot of energy annually, with increasing demand from the expanded urban area. Researchers have applied various algorithms to obtain the travel strategies offline and assumed that the train would follow the optimized travel plan in practice. Scheepmaker et al. (2017) summarized five methods to improve train energy efficiency, three of which are related to train speed trajectory and timetable, namely: 1) Energy-efficient train control (EETC), which aims to minimize the energy consumption by optimizing train speed trajectory under specific environmental conditions such as travel time, travel distance, speed limit, track slope and so forth; 2) Energy-efficient Train Timetabling (EETT), which is to discover the timetable that could maximize the efficiency of EETC; and 3) Adjusting the timetable to match passenger demand and train headway to reduce the movement of empty seats. The previous research outcomes are able to obtain the most energy-efficient timetable and speed trajectory of trains according to environmental information such as travel weight, travel distance, track slope, and speed limit.

However, rail transit suffers from the disturbances caused by the dynamic environment

(Hassanabadi, Moaveni, and Karimi 2015; H. Liu, Tian, and Li 2015), which may result in a situation that trains are not able to follow the travel plan. For example, when two adjacent trains travel on the same track, they need to keep enough distance to ensure safety. The minimum safety distance between them is generally determined by moving/fixed block systems. When the leading train needs to slow down or stop temporarily in case of emergency, the following trains need to slow down and change the travel strategy if they cannot meet the safety requirements given by the block system. In such a case, it is challenging for the zone controller to provide an optimized timetable and speed trajectory for the following trains in near real-time because finding the optimized solution is highly complicated, which leads to extra energy cost. Therefore, a near real-time re-optimization is necessary to minimize extra energy consumption as well as to pursue punctuality.

This research proposes a Multi-agent System (MAS) based rail transit control to alleviate the above problem. MAS is a method to realize distributed network control by coordinating the interaction of multiple agents, which is a decent option for realizing a complicated control system by decomposing sophisticated problems into multiple less-complex problems. The feasibility of applying multi-agent technology in the field of rail transit is increasing with the development of the Internet of Things (IoT) and Communication Based Train Control (CBTC) technology. In addition, sufficient flexibility fulfils the changeable requirements in the dynamic environment of rail transit. Some scholars try to apply Multi-agent System (MAS) in the field of rail transit due to its high flexibility and reliability (Dalapati et al. 2016; Hassanabadi, Moaveni, and Karimi 2015; Proenca and Oliveira 2004). However, the energy optimization and dynamic nature of trains are insufficiently investigated in the MAS rail control field. Furthermore, little research has comparatively investigated extra energy consumption caused by the change of dwell time and travel weight. The main objectives of this research are as follows:

1. To establish a rail transit multi-agent framework and corresponding information exchange channel structure.

2. To propose the anti-disturbance decision-making mechanism within the multi-agent framework and to realize the near real-time travel plan re-optimization.

3.To investigate the influence of train stay time and train weight variation disturbance on extra energy consumption.

The rest of this dissertation is organized as follows: Chapter 2 introduces the reviewed literature, and Chapter 3 introduces the main framework of the Multi-agent System proposed in this research, the interaction mechanism among agents, and the adopted Moving Block System (MBS). Chapter 4 verifies the effectiveness of the proposed system through four cases with numerous scenarios. Chapter 5 summarizes the main contribution, limitations of this research and raises future work.

Chapter 2: Literature Review

The main purpose of this research is to propose an innovative multi-agent control method for rail transit, which can optimize the speed trajectory of each interfered train in near real-time with the objective of minimizing energy consumption when rail transit is disturbed. Therefore, the train control system is reviewed first to provide a reference for the designed multi-agent system. Furthermore, the research on Energy-efficient Train Control, Energy-efficient Train Timetabling, and Energy-aimed Train Timetable Rescheduling is reviewed to discover appropriate optimization algorithms. Finally, the multi-agent system and the state of the art of its application in rail transit is reviewed to pursue improvements compared to previous research.

2.1 Train Control System

The control system of rail transit has achieved continuous development with the advance of engineering and communication technology, while engineers and researchers have been contributing to realize fully automated train control by using multiple systems. Research related to Automatic Train Control (ATC) has been developed for many years, and it is changing from the traditional track-based train control (TBTC) systems to communication-based train control (CBTC) systems (Bu et al. 2013).

Figure 1 shows a schematic diagram of a widely adopted CBTC system, which consists of a ground side control system and an onboard side control system. The ground side control system includes Data Storage Unit (DSU), Data Communication System (DCS), Automatic Train Supervision (ATS), Area Controller, Zone Controller (ZC) and Computer-Based Interlocking (CBI). The onboard side consists of Automatic Train Operation (ATO) and Automatic Train Protection (ATP) systems. The onboard side system collects the speed and position information of each train by using sensors and transmits the data to the Zone Controller, which then computes the safe moving distance of each train based on the block principle (such as MBS) and sends the corresponding moving authorization to the onboard side.

The stability of the ground side control system is essential in such a structure since the information exchange among trains depends on the guidelines from the ground side system during the control process. The corresponding regional controller is responsible for coordinating and authorization of multiple trains when they enter a particular area simultaneously. However, the controller is not able to provide optimized travel strategies for the trains in near real-time once a disturbance occurs in such condition because the mathematical model of timetable and trajectory for multiple trains are complex and challenging to solve (Howlett 2000; Khmelnitsky 2000; Liu and Golovitcher 2003). To guarantee safety in such a situation, the controller computes the movement authority for the trains based on the block system. A train will have to brake if the distance between it and its leading train reaches the minimum distance specified by blocking theory, which causes unnecessary kinetic energy waste.

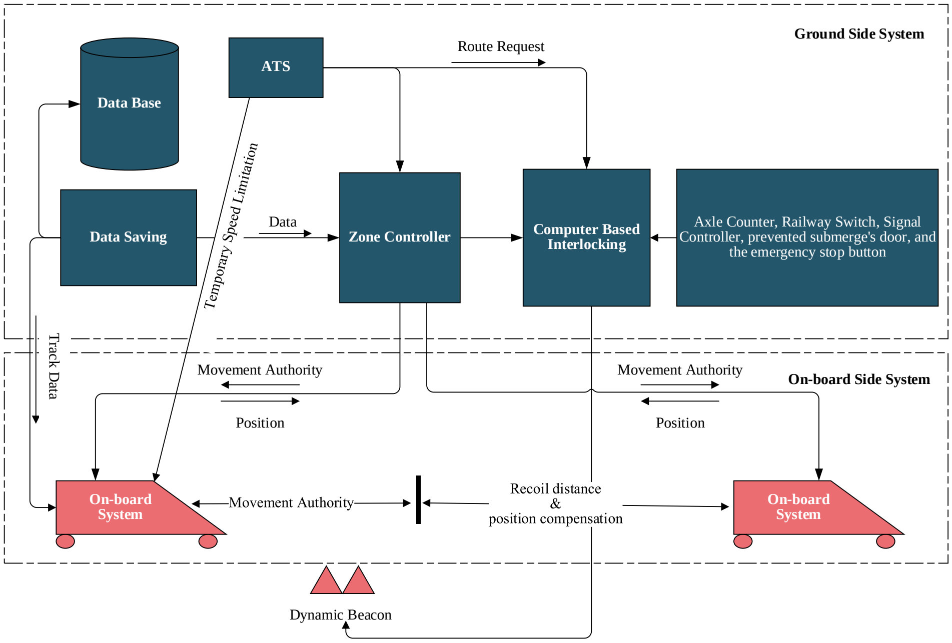


Figure 1: Rail Transit Control System (Huang 2014)

2.1.1 Energy-efficient Train Control

The designed purpose of the Train Control System is to ensure efficient information transmission in rail transit control, while the Energy-efficient Train Control (EETC) research aims to discover the train speed trajectory that requires minimum energy consumption (Scheepmaker et al. 2017).

The research of EETC could be traced back to 1968 in Ichikawa (1968), and some other publications with similar periods, such as Kokotovic and Singh (1972). The fundamental theory for the optimal train control strategy was developed by the Scheduling and Control Group (SCG) during a railway research program started in 1982 at the South Australia University, which could be found in the research of Howlett, Milroy and Pudney (1994). In addition, the first comprehensive analysis of a flat track is given by Asnis et al. (1985), who assumed the acceleration as the control variable and employed Pontryagin's Maximum

Principle (PMP) to find the necessary conditions for the optimal driving strategy. The next milestone occurred in 1989, Benjamin et al. (1989) claimed that the driving strategies of a typical diesel-electric locomotive were controlled by a throttle that could only take a finite number of positions. Those positions gave a constant fuel supply rate respectively and therefore, the power supply to the wheels was also constrained by those positions. Howlett and Pudney showed that a train (with a distributed mass on a track) running on a track with continuous changing gradient was able to be treated as a point mass train, and any driving strategies of continuous control could be approximated by a strategy with discrete control (Howlett et al. 1994). In order to determine the coast-accelerate-brake point by combining the factors of energy-efficient, punctuality and riding comfort, Chang and Sim (1997) applied the Genetic Algorithm (GA) (belongs to the evolutionary algorithm) to derive the driving strategy in 1997.

Research about the train speed trajectory optimization has been increasing with a continuous improvement in both optimization effect and calculation speed. Generally, the algorithms for optimizing the train speed trajectory with the objective of minimizing energy consumption can be divided into two categories, which are mathematical programming and heuristic algorithms. Some recent publications based on mathematical programming are selected and listed in Table 1.

Table 1: Research for Energy-efficient Train Control Based on Mathematical Programming

Main Algorithm/Theory	Publication	Multiple/Single Train(s)
PMP	(Howlett 2000)	Single Train
	(Khmelnitsky 2000)	Single Train
	(Liu and Golovitcher 2003)	Single Train
	(Albrecht et al. 2016a, 2016b)	Single Train

Sequential Quadratic Programming	(Miyatake and Matsuda 2009)	Single Train
Bellman-ford Algorithm	(Lu et al. 2014)	Single Train
Kuhn-Tucker Conditions	(Li and Lo 2014b)	Multiple Trains
Pseudospectral Method and MILP	(Y. Wang et al. 2013)	Single Train
MILP	(Lu et al. 2016)	Single Train
MILP & PMP	(Tan et al. 2018)	Single/Adaptive
Pseudospectral method	(Wang and Goverde 2016)	Single Train
	(Wang and Goverde 2017)	Multiple Trains
Dynamic Programming	(Haahr, Pisinger, and Sabbaghian 2017a)	Single Train
Monte Carlo Simulation	(Tian et al. 2017)	Multiple Trains

As shown in Table 1, Howlett (2000) raised that it was probably not reasonable to assume that the acceleration is a uniformly bounded control. He studied the optimal control under continuous and discrete control by using PMP with time as the independent variable. The key equation for determining the optimal switching point is established according to the necessary conditions. In order to further determine a detailed program for minimizing the energy consumption in traction and brake applications, Khmelnisky (2000) investigated this problem and considered the variable gradients and velocity limits. Additionally, an analytic solution for the sequence of operation change points and optimal controls of trains are offered by Liu and Golovitcher et al. (2003). Albrecht et al. (2016a, 2016b) summarized the key principles of optimal train control and discussed the optimization control problems of trains in different aspects. Their research proved that the control strategy leading to minimal energy cost always exists and is unique among the problems of a single train running between two stations. Miyatake and Masuda (2009) proposed a speed trajectory optimization method by considering the charge and discharge of the on-board storage device based on sequential quadratic programming. Wang et al. (Y. Wang et al. 2013) formulated the optimization problem of train speed trajectory optimization by taking the travel comfort into account and solved it by pseudo-spectral method and MILP algorithm. Lu et al. (2014) employed the Berman-Ford algorithm to model train braking speed trajectory in a discrete manner and conducted optimization analysis by PMP, and the paper demonstrates a high degree of consistency between the analysis and optimization results. Li and Lo (2014b) proposed an integrated method including passenger flow prediction, timetable, and speed trajectory optimization to realize dynamic train scheduling and reduce energy consumption for the metro system. However, their method ignored the coasting phase in the train speed trajectory to reduce the computational complexity. Lu et al. (2016) integrated the nonlinear constraints caused by gradient change in braking route into a MILP model and solved some speed trajectory optimization problems. Wang and Goverde (2016) solved the multi-stage optimal control model of trains with and without delay by pseudo-spectral algorithm. Afterwards, they proposed a model which formulated the multi-objective, multi-train and multi-phase optimization problem by using pseudo-spectral method, and the optimization objectives include energy consumption and delay (Wang and Goverde 2017b). Haahr et al. (2017) proposed a method to optimize the velocity trajectory through Dynamic Programming algorithm, in which the search space is reduced through event-based decomposition and thus the computation speed is accelerated. Zhao et al. (2017) proposed an integrated model and solved it by genetic and brute force methods. The model adjusted the timetable and speed trajectory simultaneously to minimize energy consumption. Based on Monte Carlo simulation, Tian et al. (2017) proposed a comprehensive optimization method, combining train operation and power flow.

Besides the mathematical planning algorithm, scholars also employed heuristic algorithms to solve the train speed trajectory optimization problem. Compared with the planning algorithm, the heuristic algorithm model is easier to build for some complex problems and could provide a feasible solution in an acceptable time (Apter 2018; Pearl 1984). However, the trade-off between the viable and optimal solutions obtained by the heuristic algorithm is hard to determine. Heuristic algorithms are widely adopted for multi-train optimization problems. Some recent publications focusing on solving EETC problems by using heuristic algorithms are listed in Table 2.

Table 2: Research for Energy-efficient Train Control based on Heuristic Algorithm

Main Algorithm/Theory	Publication	Multiple/Single Train(s)
Genetic Algorithm (GA)	(Y. V. Bocharnikov et al. 2007)	Single Train
Genetic Algorithm, Ant Colony optimization and Dynamic Programming	(Lu et al. 2013a)	Single Train
Genetic Algorithm	(Li and Lo 2014a)	Multiple Trains
Brute Force, Ant Colony and Genetic Algorithm	(Zhao et al. 2015)	Multiple Trains
Genetic Algorithm	(Yang et al. 2016)	Multiple Trains
Genetic Algorithm	(Liu, Xun, and Bin 2017)	Single Train
Genetic Algorithm and Brute Force	(Zhao et al. 2017b)	Multiple Trains

As shown in the table, the Genetic Algorithm is used in most related research due to its advantages, such as the ability to discover the global optimum solution. Bocharnikov et al. (2007) employed the Genetic Algorithm to optimize the energy consumption of a single train and explored the relationship between journey time and energy consumption. Lu et al. (2013b) established a distance-time-speed model and realized the optimization of train speed trajectory in discrete search space by using the combination of Genetic Algorithm, Ant Colony Algorithm and Dynamic Programming. Li and Lo (2014b) established an optimization model aiming at minimizing net energy and solved it by Genetic Algorithm with considering the timetable and train control. Therefore the model is suitable for resolving the problem of multi-train optimal control with regenerative braking. Zhao et al. (2015) established a multi-objective model integrating delay and energy consumption and solved the multi-objective model based on Brute Force Algorithm, Ant Colony Algorithm and Genetic Algorithm. Yang et al. (2016b) proposed a model based on Genetic Algorithm to optimize the train timetable and speed trajectory by solving a two-phase stochastic model with uncertain train mass. The model assumes that the optimal speed trajectory of the train is composed of maximum acceleration, coasting and deceleration to reduce the computation complexity. Liu et al. (2016). proposed a model to optimize the train speed trajectory with regenerative braking through Genetic Algorithm. The net energy consumption is reduced for the regenerative braking was considered.

2.1.2 Energy-efficient Train Timetabling

The objective of EETT research is the same as the EETC, which is to reduce the total energy consumption, but EETT focuses on discovering the optimized timetable for one or multiple trains on single railway lines or in a network. It is expected that the total energy consumption will be minimized after each train adopts the EETC driving strategy between each pair of consecutive stations with the optimized timetable. Mills and Peerkins (1991) began to explore EETT to solve the traffic problem of freight trains in Australia. As some tracks of the Australian railway network have trains running in two directions simultaneously, it is necessary to reschedule the trains when there is a conflict. They proposed a method to solve the meet-pass problem based on a discrete heuristic algorithm and re-optimized the speed trajectory based on a nonlinear optimization model. The testing between Port Augusta and Tarcoola in Australia showed 6% of energy savings, and the computing of resolving the problem and optimising speed trajectory took 3.3 minutes and

21.5 minutes, respectively, on a HP9000/340 workstation. The research of EETT gradually increased after 2000 with a continuous improvement of computing power. Similar to EETC, this section selects some recent EETT researches and classifies them according to the applied algorithm types. The researches based on Mathematical Programming are summarized in Table 3.

Table 3: Research for Energy-efficient Train Timetabling Based on Mathematical Programming

Main Algorithm and Theory	Publication	Multiple/Single Trains	Tested Station Scope
Dynamic Programming	(T. Albrecht and Oettich 2002)	Single Train	10 Stations
Nonlinear Mathematical Programming, and ϵ-constraint method	(Ghoseiri, Szidarovszky, and Asgharpour 2004)	Multiple Train	Varying numbers of stations
Mathematical Programming and DC power flow model	(Pena-Alcaraz et al. 2012)	Single Train	36 Stations
Dynamic Programming	(Binder and Albrecht 2013)	Single Train	7 Stations
Pseudospectral method	(Wang and Goverde 2016b)	Multiple Trains	7 Stations

Albrecht and Oettich (2002) used a simulation model to calculate the energy utilization of each discrete running time between two consecutive stations and then used a dynamic programming algorithm to find the optimal timetable. The model is able to optimize the total running time of the train on the line. In addition, the model will also attempt to increase the possibility that passengers can still catch the next train in the case of train delays. The model suggests reducing waiting time and increasing train running time to reduce energy consumption. Tests on the railway system in Dresden show that the system reduces energy consumption by 15-20%.

Ghoseiri et al. (2004) considered a train operation network model including single and double track, multiple trains, and stations. They established a nonlinear mathematical programming model and solved the model by LINGO (a commercial solver). The established model has two optimization objectives: minimizing fuel consumption and total running time. As the decrease of running time will increase energy consumption, it is necessary to weigh the two optimization objectives. In their research, the Pareto curve of the trade-off between energy consumption and running time is determined by the ϵ-constraint method. Then the appropriate timetable is selected from the Pareto curve according to the distance during the process of multi-objective optimization. The test results in an artificial example show that energy consumption increases with the decrease of travel time. Hence, the relative characteristics of the two optimization objectives make it impossible to get the only optimal driving strategy in this model.

Peña-Alcaraz et al. determined the timetable of the metro system based on a mathematical programming model in 2012 and optimized the net energy consumption of trains by considering regenerative braking. This research focuses on maximizing the utilization efficiency of energy generated by regenerative braking by synchronizing regenerative braking and acceleration of adjacent trains. The adopted model simulates the synchronization of acceleration and braking through the power flow model. It is worth noting that the focus of this study is not the speed trajectory of a single train; hence EETC is not considered in their model. The research reports that the simulation of the Madrid metro system shows 7% average *ESP* on the premise of ensuring passenger service (Pena-Alcaraz et al. 2012). Binder and Albrecht (2013) explored train timetabling and energy-efficient operation. They developed a dynamic planning algorithm, which fixed the running time between two main stations and adjusted the arrival and departure time of the intermediate station to reduce energy consumption. Optimization objectives include reducing energy consumption and reducing the delay of arriving at each station. The randomness of the dwell time change is considered in this model, and the test between seven stations in Germany shows that this model can save energy by 4.3% and 12.9%. The fluctuation of energy-saving percentage depends on the weight setting between optimization objectives (Binder and Albrecht 2013).

Wang and Goverde (2016c) used the pseudospectral method to solve the train timetable optimization problem under various constraints. This research focuses on optimizing the timetable and speed trajectory of the following train when the leading train is delayed. The case study shows a 50-kilometre-long line, and the optimization objective consists of reducing the delay of the following train and the total energy consumption. The results show that the following train could reduce delays and total energy consumption if the leading train can accurately provide forecast information.

Table 4 lists some publications on EETT based on the heuristic algorithm, and it can be found that genetic algorithm is widely used in this field. Albrecht considered adjusting running time to synchronize the acceleration and regenerative braking of adjacent trains so as to improve the utilization of the energy generated by regenerative braking in 2004. The model minimizes the total energy consumption and peak power by finding the optimal adjustment of running time. The case study shows that the method can save 4% energy and reduce 17% sum of 15-min-average power compared with optimizing the running time of a single train while the dwell time is unchanged (Albrecht 2004).

Table 4: Research for Energy-efficient Train Timetabling Based on Heuristic Algorithm

Main Algorithm and Theory	Publication	Multiple/Single Trains	Applied Station Scope
Genetic Algorithm	(Albrecht 2004)	Single Train	16 Stations
Genetic Algorithm	(Ding et al. 2011)	Single Train	6 Stations
Genetic Algorithm, Simulation, Fuzzy Linear Programming	(Cucala et al. 2012)	Single Train	6 Stations
Genetic Algorithm	(Yang et al. 2013)	Multiple Trains	14 Stations
Genetic Algorithm	(Li and Lo 2014a)	Multiple Trains	14 Stations
Genetic Algorithm	(Yang et al. 2014)	Multiple Trains	14 Stations
Genetic Algorithm	(Yang et al. 2015)	Multiple Trains	14 Stations

Ding (2011) considered acceleration, coasting, and braking in driving state and established a two-level iterative optimization model to determine the best timetable and energy-saving driving strategy of subway lines. The genetic algorithm is adopted to solve the model, and the results showed that the model could reduce energy consumption by up to 19.1%. Cucala et al. (2012) simulated the uncertain delay of rail transit by using fuzzy numbers and punctuality constraints and established a two-objective model to reduce energy consumption and delay. They find the optimal timetable by allocating the assignable time in the trip. The test results of the study during the journey from Madrid to Barcelona show that compared with the commercial timetable, the energy consumption can be saved by 5.25% when there is no delay, and the energy consumption can be reduced by 6.67% if delay exists.

Yang et al. (2013) described the synchronization process by a mathematical model and then discovered the optimal synchronization timetable by genetic algorithm. The energy consumption is indirectly reduced by maximizing the time overlap between the acceleration and braking of neighbouring trains. The case study shows the model improves the overlap of acceleration and regenerative braking of neighbouring trains by 15.2% during off-peak hours and 22.1% during peak hours. Yang et al. (2014) further explored the model by integrating the waiting time of passengers into consideration. The uncertainty of dwell time is ignored in the model, and the genetic algorithm is employed again to provide solutions. This study proposes that the model can reduce 8.9% of energy and 3.2% of passenger waiting time when all renewable energy can be fully utilized. Yang et al. (2015) extended the time span to one day and established a model considering all trains in the same track section. On the premise that all trains run according to the optimal speed trajectory, all trains arrive and leave synchronously through model coordination to improve the utilization efficiency of renewable energy. The scheduling problem is first described by a mixed-integer programming model, and then it is solved by the genetic algorithm. Compared with the algorithm in 2013, the research improves the utilization rate of renewable energy by 36.2% and reduces energy consumption by 4.3%.

Li and Luo (2014) raised a model to reduce the net energy consumption of trains by assuming that the trains have a constant acceleration rate, decision rate, and running resistance. The model considers the optimization of timetable and speed trajectory simultaneously. Similar to other scholars' research, the timetabling part tries to improve the utilization efficiency of energy generated by regenerative braking through regenerative braking of synchronous trains and acceleration of adjacent trains. The model relies on the genetic algorithm to solve the problem, and the test results show that when the headway between trains is 90 seconds, the energy consumption can be reduced by 25%, but the percentage of energy-saving decreases with the increase of headway.

2.1.3 Energy-aimed Train Timetable Rescheduling

Due to the considerable computational complexity of the timetabling, most of the previous methods complete the optimization process offline. However, a train may not be able to continue the pre-optimized timetable if a disturbance occurs during the operation process, which leads to extra energy consumption (Zhu and Goverde 2020). Most Train Timetable Rescheduling research focuses on minimizing the total delay time of all trains or passengers (Dalapati et al. 2016; Ortega, Pozo, and Puerto 2018). To reduce the unnecessary energy loss caused by disturbance, researchers began to explore the Energy-aimed Train Timetable Scheduling (ETTR) in recent years.

Gong et al. (2014) proposed a timetable adjustment method to reduce the total energy consumption caused by interference on the whole trip. In their approach, a train speeded up in the following section if it spent extra dwell time at a station to ensure that it could arrive at the next station on time. Therefore, the gap between the adopted timetable and the planned one is reduced. This method leads to an increase in the energy consumption of the catch-up section, whereas the total energy consumption will be controlled at an acceptable level due to the offline timetable been highly optimized. Afterwards, Yang et al. proposed a near real-time timetable optimization method based on deep reinforcement learning (Yang et al. 2019). This method is able to re-organize the timetable for all the affected trains after a disturbance occurs. However, deep reinforcement learning algorithms require tremendous iterations, which results in high training costs. Besides, the optimization in rail transit needs to consider plenty of parameters because of the high requirements on safety, travel comfort, parking accuracy, etc. Thus it is difficult to ensure that the algorithm based solely on the neural network can provide reasonable decisions under any circumstances.

2.2 Multi-agent System

2.2.1 Agent

The first widely adopted definition of the agent is given by Wooldridge and Jennings (1995):" An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to achieve its delegated objectives ". Weiss adopted this definition (2012) and showed a schematic diagram of an agent (Figure 2). The figure shows that the interaction process between the agent and the environment is divided into three steps: collecting information from the environment, making decisions based on the information, and executing actions that may have impacts on the environment. Russell and Norvig show different types of agents that have these three steps, and they claimed that agent is something that can act with five characteristics: autonomous operation, environmental perception, long-term persistence, adaptation to change, creation and pursuit of goals (Russell and Norvig 1996). However, the interaction with the environment is abstract and covers a wide range. Some objects have similar functions to this definition, such as an object in object-oriented programming languages and Expert System (Weiss 2012).

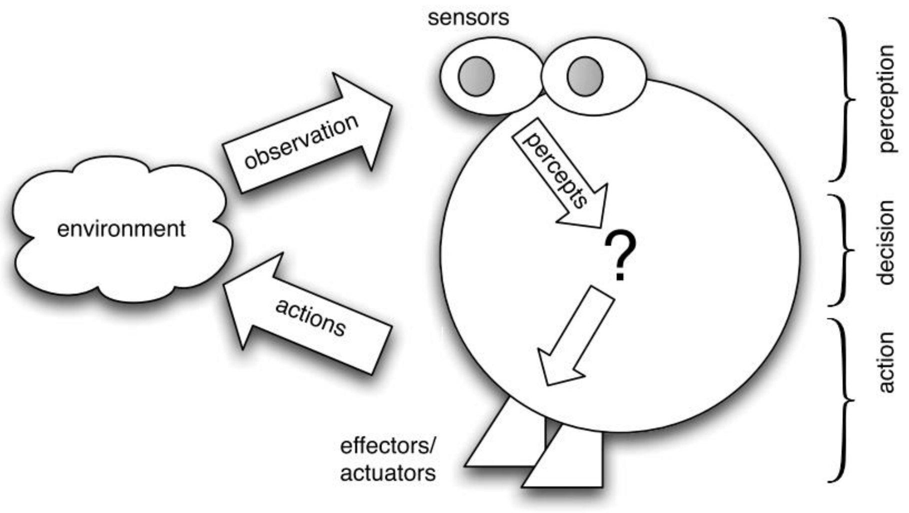


Figure 2:Demonstration of *agent* (Weiss 2012)

The main difference between an *agent* and an *object* in object-oriented programming languages is that normally an *agent* has higher autonomy. Both *agent* and *object* in object-oriented programming languages, such as JAVA, Python, obtain information from the environment, make decisions based on internal logic, and provide feedback to the environment. However, engineers need to declare the property for each variable as public or private within an *object* during the programming process. Private variables or methods can only be accessed and used by the *object* that owns them, while public variables are able to be called by other objects. Once a public method is defined in an object, the *object* will not be able to refuse other objects to use the method. On the contrary, the interactions between agents are more complicated than objects. It is unnecessary for an *agent* to perform an action when it receives a request from another *agent* because the profit of different agents might have conflicts in a sophisticated environment. Furthermore, agents have a higher ability to cope with environmental changes. An *object* will initiate a decision-making process and assumes that the environment would not change during the computing process when it receives the calling from another object. However, there is the possibility of new changes in the environment during the process of decision execution, which invalids the result about to be generated by the object. For an agent, however, it is able to continue receiving updates from the environment. The *agent* will receive a new event triggering code if necessary so as to decide whether to continue the current action.

2.2.2 MAS Structure

MAS is a method to realise distributed network control by coordinating multiple agents (Ge, Yang, and Han 2017), which is usually adopted to solve complex problems that are challenging or impossible to solve by a single agent. Agents in MAS interact with each other to pursue the designed goals through cooperation or competition. Compared with single-agent systems, an MAS has three advantages: 1. It can transform a complex problem into several simple problems and reduce the difficulty of building internal logics and mathematical models (Balaji and Srinivasan 2010). 2. A MAS system has better flexibility and extensibility than a Single-agent System. The editions of logic, algorithm or neural network model in a deployed MAS system are easier to be carried out than a single agent controlled one under the same sophisticated engineering field (Weiss 2012). For example, engineers could embed a Recurrent Neural Network (RNN) into agents responsible for time-series prediction and Back Propagation Neural Network (BPNN) into agents responsible for classification. 3. In a fluent communication environment, multiple agents can be collaborated in parallel processing to solve complex models and speed up the computation process (Rousset et al. 2016).

The MAS system has three kinds of structures: centralised, decentralised, and hybrid (Zhang and Hammad 2012). Figure 3 shows the comparison between Single-agent Control and the three structures of MAS control.

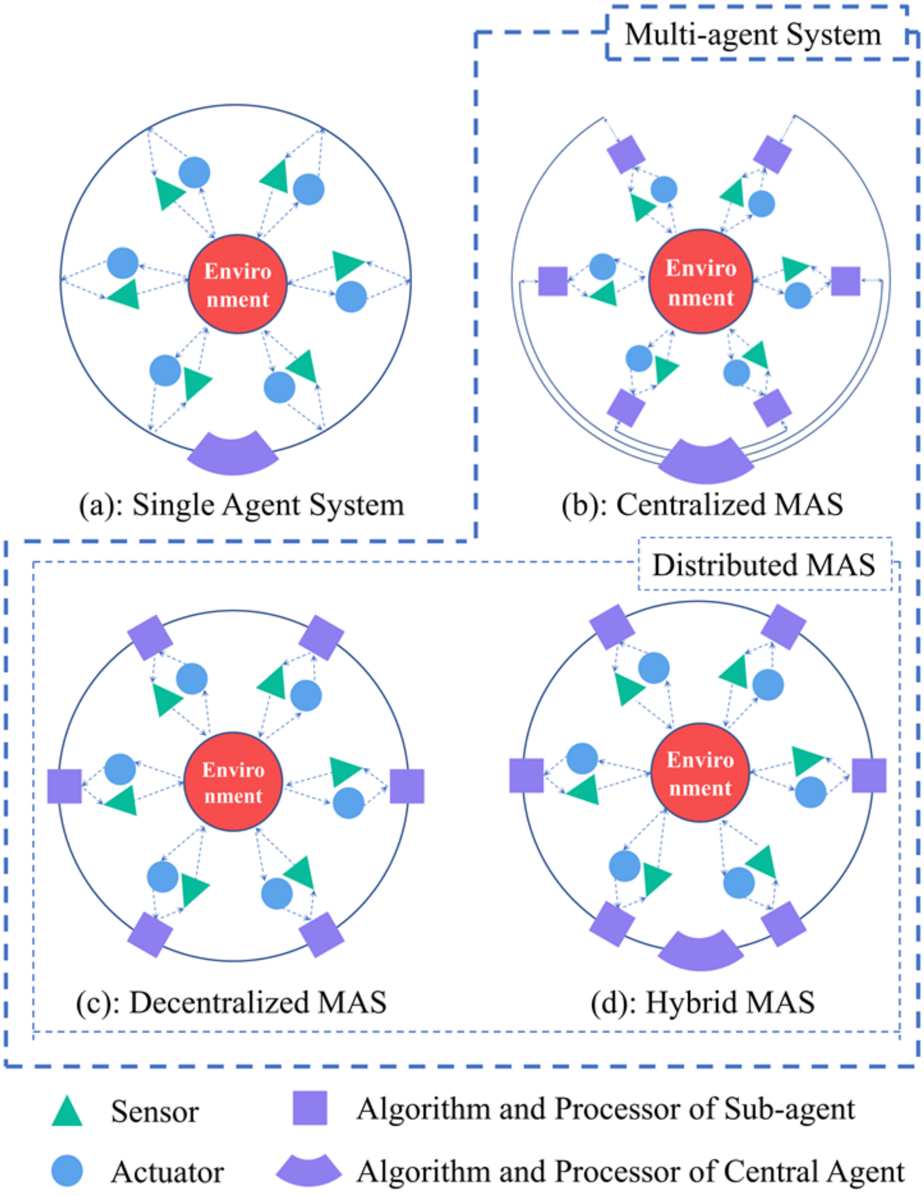


Figure 3: Different Types of Agent-Based Systems

The main differences between Single-agent System and MAS are that Single-agent System only includes one processing node (Figure 3 (a)), while the MAS system includes multiple processing nodes (Figure 3 (b-d)) (Weiss 2012). The centralised structure includes a plurality of subagents and a central agent (see Figure 3 (b)). Subagents collect information from the environment and influence the environment through sensors and controllers, respectively (Q. Liu et al. 2015). A subagent will send a request to the central agent and wait for the reply (command) from the central agent when it encounters a task that is hard to be handled independently. There is no communication channel that directly connects subagents. If the interests of two (or more) subagents conflict, the central agent will send a command to the involved subagents and guides the actions according to the global objective. The involved subagents then execute the received commands, even if the action will reduce their own interests. The central agent has global information of involved subagents in this framework, so it is easier for it to discover the global optimal solution when there are conflicts among subagents. However, the central agent needs to face considerable computational pressure, and the stability of the structure will be greatly affected if the central agent is overloaded.

The Decentralised structure does not include a central agent, as shown in Figure 3 (c), the subagents exchange information among the directly connected channels. The involved subagents will negotiate between themselves and gain profits in the environment through cooperation or competition if there are conflicts. Typically the MAS system with this structure needs to declare the benefit priority of different subagents in specific conflict events to ensure rapid conflict handling. This framework has good stability and will not cause the whole system paralyses due to the crash of a single subagent. However, it is more challenging to get the global optimal solution than a centralised structure because each subagent only collects local information.

The third framework is in Hybrid form, as shown in Figure 3 (d), which includes at least one central agent and several subagents. The information in this MAS structure can be directly exchanged among subagents and through the central agent. Usually, the subagents attempt to solve conflicts by themselves. The central agent only participates in coordinating the benefits of subagents if there is a conflict that the authority or computing power of subagents is insufficient to solve. Therefore, this system is similar to the decentralised structure for general conflicts. This kind of system has a higher ability to deal with complex problems due to the existence of the central agent, which is adopted in the proposed system.

2.2.3 Interaction Triggering Mechanism between Agents

The triggering mechanism of exchange information among agents in a MAS system has a significant influence on the efficiency of the system. Inadequate communication and collaboration between agents will reduce the system's ability to cope with complex changes in the environment, while excessive interaction leads to unnecessary waste of communication and computing resources. In general, the Triggering Mechanism for agents to exchange information can be divided into two categories: Time Triggering Mechanism (TTM) and Event Triggering Mechanism (ETM).

Each agent exchanges information with the pre-set time interval in TTM, which consists of three communication types, which are periodic communication, variable time communication and random communication. TTM is easier to deploy compared with ETM. Communication logic based on TTM is simple, and the relevant deployment is less challenging. However, the designer needs to set the communication frequency higher than required to ensure the correct operation of the system and improve stability. Frequent data exchange will result in the waste of computing and communication resources (Zhang et al. 2014).

For ETM, a series of events that demand communication is required to be determined in advance. When a pre-set event occurs, the agent that senses the event will initiate contact and send information to other agents that will be affected. Thus ETM is able to save resources compared with the TTM because agents only exchange data when an event occurs. However, the possible events for ETM should be analysed and set comprehensively and accurately. Otherwise, it may lead to accidents caused by insufficient collaboration among agents. Especially, accidents in rail transit directly endanger safety. Table 3 shows some literature based on TTM and ETM.

Table 5:Relevant MAS & Control Network Publications Classified According to Triggering Mechanism and Configurations

Publication	TTM	ETM
Centralized Configuration	(Hui et al. 2014; Wang and Han 2015, 2015)	(Jia et al. 2014; Peng and Han 2013; Peng, Han, and Yue 2013; Xun et al. 2013; Yue, Tian, and Han 2013; Zhang and Han 2014)
Distributed/Decentralized Configuration	(Ding, Han, and Guo 2013; Z. Wang et al. 2013; You, Li, and Xie 2013)	(Guinaldo et al. 2014; Guo, Ding, and Han 2014; De Persis, Sailer, and Wirth 2013; Zhang et al. 2014)

2.3 Multi-agent System in Rail Transit

The application of MAS technology in rail transit started in the 20th century. Burckert et al. described a system with a global agent and mobile agents, which transmits schedules in a dynamic environment (Burckert, Fischer, and Vierke 1998). Agents coordinate and control resources through communication in the prototype system. Afterwards, Linde and Fisher (1999) proposed a mobile agent system that can be planned and monitored (Lind and Fischer 1999).

Proenca and Olivier proposed a MAS system for rail transit that combines control function and learning function. The control function is realized by three Agents (supervisor, train and station), which is responsible for meeting the basic requirements of train safety, while the learning layer improves the rules of train driving strategy according to the short legs of driving data (Proenca and Oliveira 2004). Verma and Pattanaik (2014) demonstrated a MAS based system that adopts the MBS to ensure safety. The system implements a simplified sub-target mobile licensing model, but the cooperation and interaction among agents have not been demonstrated. Hassanabadi et al. (2015) proposed a rail transit control framework based on the MAS system, which is composed of Station Agent, Train agent and Central Control agent. A train under the control of the system will transmit the information to the following train if it encounters an emergency and needs to stop temporarily. The corresponding following train would take braking once the distance between the two trains cannot meet the safety requirements. Although this mechanism could ensure the driving safety of every train under the control of the system, the energy optimization method is not considered to reduce unnecessary energy consumption when conflicts occur.

Chapter 3: Proposed Methodology

This chapter first demonstrates the control framework of the proposed MAS system, and then introduces the dynamic interaction mechanism of the agents within the framework. The adopted MBS (which is used for ensuring safety) is introduced after the interaction mechanism section.

3.1 Control Framework

Part of the proposed system is shown in Figure 4. The system includes three types of agents, namely Central Agent, Train Agent, and Station Agent. There are four tracks in the figure, where trains on Tracks 1 and 2 travel from left to right and trains on the other two tracks travel in the opposite direction. $Train(i\_n)$ is used to represent any running train, where n means the train is running on the n^{th} track and i represents the train sequence on the track. For instance, $Train(1\_2)$ is the first train running on Track 2, and $Train(2\_2)$ is the following train of it. Figure 4 shows the moment where five trains are running on Track 1 from left to right. $Train((i-1)\_1)$, $Train(i\_1)$ and $Train(i\_2)$ are running between Station D and Station E, and $Train((i+1)\ \_\ 1)$ and $Train((i+1)\ \_\ 2)$ are not reach to Station D yet.

Figure 4 demonstrates the communication structures of trajectory optimization and timetable optimization, and the communication process is demonstrated in the interaction mechanism section. There are two approaches in which subagents (train or station agents) exchange information. The first approach is to exchange information with the connected subagents directly from the connected channel when the system is functioning properly. The central agent is used as the communication medium in the second approach, which will be activated if a subagent does not receive the desired information from another subagent or encounters a problem that cannot be resolved independently. The involved subagent sends the application to the central agent in such conditions, and then the central agent delivers the received message to the subagent that needs the information and waits for feedback. The feedback will be transmitted to the subagent that sent the application once it is received. Theoretically, the proposed communication ensures that the system is at least as stable as the system in use, for the second communication method is similar to the currently adopted one.

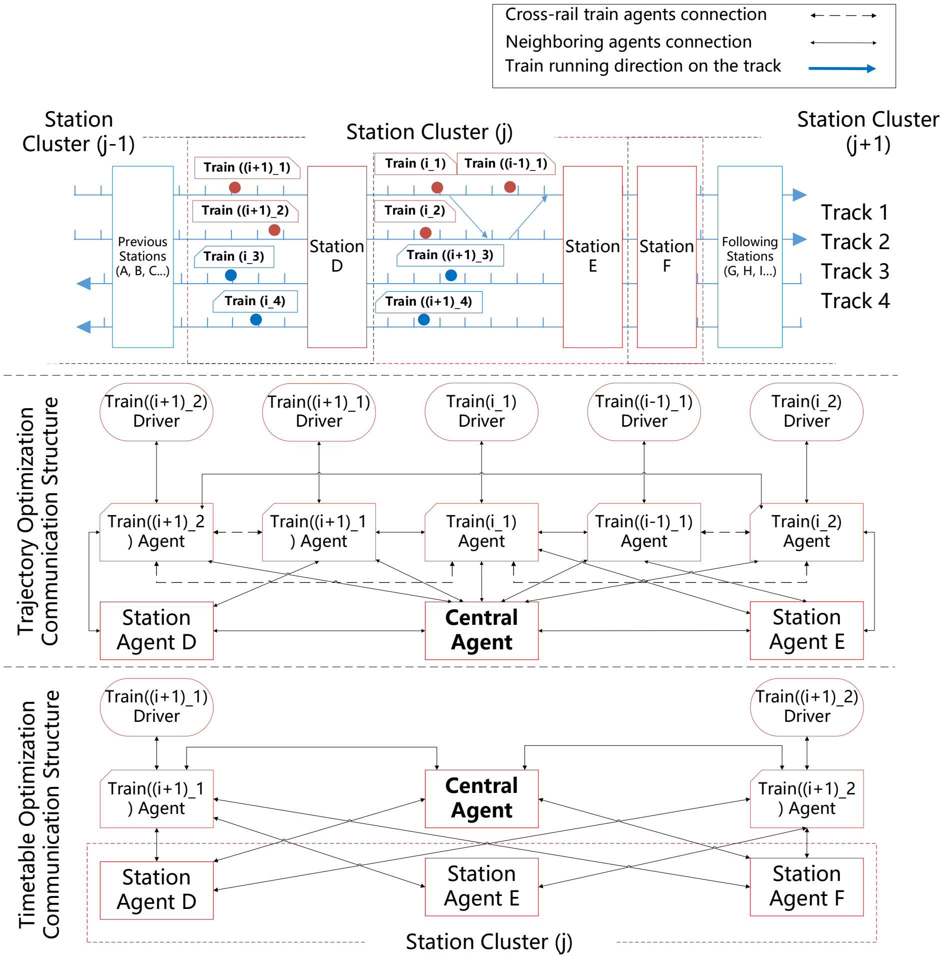


Figure 4: Part of the Proposed MAS System (Guo et al. 2021)

3.2 Train Agent

This section includes two parts: Trajectory Optimization and Timetable Optimization. Trajectory Optimization is based on the MILP algorithm, and the constraint conditions of the optimization model are presented in the first section. Timetable Optimization is based on a trained Deep Neural Network (DNN). The corresponding section introduces the data generation process for training the optimization model and the optimization effect after training.

3.2.1 Trajectory Optimization

The task of the train agent is to optimize the timetable and speed trajectory based on the received information. A Mixed-Integer Linear Programming (MILP) based model is adopted to optimise the speed trajectory, which is introduced in this section.

The journey D of a train between any two points is equally divided into several segments with a distance of Δd. Thus the relationship between d and Δd is represented by Eq. (1):

$$D = \sum_{i-1}^{N} \Delta d \qquad \text{Eq. (1)}$$

The Davis' equation is used to calculate the driving resistance of trains in each Δd:

$$F_{i,drag} = A + Bv_i + Cv_{i,avg}^2 \qquad \text{Eq. (2)}$$

The train speed will not have drastically variation when Δd is short. Thus the speed change within each Δd is able to be approximately linearized. Assume the speed of a train is vi when the train first enters the segment Δd, while the speed is vi+1 at the moment the train left. Then the average speed of the train within the segment can be approximated as:

$$v_{i,avg} = 0.5 * (v_i + v_{i+1}) \qquad \text{Eq. (3)}$$

By including a series of nonnegative variables of special ordered sets type 2 (SOS2), a nonlinear function can be represented by piecewise linearity. This SOS2 can have up to two adjacent non-negative variables, and the sum of all variables is limited to 1. Set a small constant δ, which represents the effective step size from V_{min} to V_{max}, and $K = (V_{\max} - V_{\min})/\delta, K \in \mathbb{N}$. Thus the decision variables v_i^2 could be expressed by:

$$v_i^2 = \sum_{k=1}^{K} (V_{\min} + (k-1)\delta)^2 \cdot \alpha_i^k \qquad \text{Eq. (4)}$$

Set v_i' as an approximation of speed v_i, then it could approximately be obtained as:

$$v_i' = \sum_{k=1}^{K} (V_{\min} + (k-1)\delta) \cdot \alpha_i^k \qquad \text{Eq. (5)}$$

$$\sum_{k=1}^{K} \alpha_i^k = 1 \qquad \text{Eq. (6)}$$

$$0 \leq \alpha_i^k \leq 1, k = 1,2,\cdots,K \qquad \text{Eq. (7)}$$

In order to control the value from α_i^1 to α_i^K to satisfy SOS2, set a set of variable λ_i^1 to λ_i^{K-1}, where:

$$\lambda_i^k \subseteq \{0,1\} \qquad \text{Eq.}(8)$$

$$\sum_{k=1}^{K-1} \lambda_i^k = 1 \qquad \text{Eq. (9)}$$

$$-\alpha_i^k - \alpha_i^{k+1} + \lambda_i^k \leq 0 \qquad \text{Eq.}(10)$$

An approximation of the average velocity of $v_{i,avg}$ for each distance segment can be approximately calculated by using Eq. (13). The variable β_i^k is used to obtain another set of SOS2, which is used to find the approximation of $v_{i,avg}^2$ and $1/v_{i,avg}$ within each distance segment, as shown in Eq. (11) to Eq. (15):

$$\sum_{k=1}^{K} \beta_i^k = 1 \qquad \text{Eq. (11)}$$

$$0 \leq \beta_i^k \leq 1, k = 1,2,\cdots,K \qquad \text{Eq. (12)}$$

$$v_{i,avg} \approx v'_{i,avg} = \frac{v'_i + v'_{i+1}}{2} = \sum_{k=1}^{K} (V_{\min} + (k-1)\delta) \cdot \beta_i^k \qquad \text{Eq. (13)}$$

$$v_{i,avg}^2 \approx v'^2_{i,avg} = \sum_{k=1}^{K} (V_{\min} + (k-1)\delta)^2 \cdot \beta_i^k \qquad \text{Eq. (14)}$$

$$\frac{1}{v_{i,\text{avg}}} \approx \frac{1}{v'_{i,\text{avg}}} = \sum_{k=1}^{K} \frac{\beta_i^k}{V_{\min} + (k-1)\delta} \qquad \text{Eq. (15)}$$

Similarly, in order to control the value from β_i^1 to β_i^K to satisfy SOS2, set a set of variable γ_i^1 to γ_i^{K-1}, where:

$$\gamma_i^k \subseteq \{0,1\} \qquad \text{Eq.}\,(16)$$

$$\sum_{k=1}^{K-1} \gamma_i^k = 1 \qquad \text{Eq. (17)}$$

$$-\alpha_i^k - \alpha_i^{k+1} + \gamma_i^k \leq 0 \qquad \text{Eq.}\,(18)$$

In order to ensure passenger travel comfort, it is stipulated that the maximum acceleration and deceleration of trains are a_{max} and $a_{d,max}$, respectively, and the acceleration (deceleration) of a train should be within its limit range. Therefore:

$$-a_{d,max} \leq a_i = \frac{v_{i+1}^2 - v_i^2}{2\Delta d} \leq a_{max} \qquad \text{Eq. (19)}$$

The time for a train to pass a Δd is equal to the distance divided by the average speed, thus:

$$\Delta t_i = \frac{\Delta d}{v_{i,avg}} \qquad \text{Eq. (20)}$$

Furthermore, the speed of trains on unidirectional tracks is non-negative, thus:

$$0 \leq v_i^2 \leq V_{max}^2 \qquad \text{Eq. (21)}$$

The energy loss during the transmission process is not less than 0, which is represented by:

$$\begin{gathered} E_i \eta_t - \frac{1}{2} M(v_{i+1}^2 - v_i^2) - F_{i,drag} \mathit{\Delta} d - Mg \mathit{\Delta} h_i \geq 0 \\ \frac{E_i}{\eta_b} - \frac{1}{2} M(v_{i+1}'2 - v_i'^2) - F_{i,drag} \mathit{\Delta} d - Mg \mathit{\Delta} h_i \geq 0 \end{gathered} \qquad \text{Eq. (22)}$$

The kinetic energy reduced by the maximum braking force is greater than or equal to the maximum kinetic energy reduction, while the kinetic energy increased by the maximum traction force is less than or equal to the maximum kinetic energy increase. Therefore, there are constraints of Eq. (23) to Eq. (26):

$$-F_{i,drag} \mathit{\Delta} d - Mg \mathit{\Delta} h_i - \frac{1}{2} M(v_{i+1}^2 - v_i^2) \leq F_{b,max} \mathit{\Delta} d \qquad \text{Eq. (23)}$$

$$F_{i,drag} \mathit{\Delta} d + Mg \mathit{\Delta} h_i + \frac{1}{2} M(v_{i+1}^2 - v_i^2) \leq F_{t,max} \mathit{\Delta} d \qquad \text{Eq. (24)}$$

$$-F_{i,drag} \mathit{\Delta} d - Mg \mathit{\Delta} h_i - \frac{1}{2} M(v_{i+1}^2 - v_i^2) \leq P_{b,max} \mathit{\Delta} t_i \qquad \text{Eq. (25)}$$

$$F_{i,drag} \mathit{\Delta} d + Mg \mathit{\Delta} h_i - \frac{1}{2} M(v_{i+1}^2 - v_i^2) \leq P_{t,max} \mathit{\Delta} t_i \qquad \text{Eq. (26)}$$

The total travel time of a train should not exceed the specified maximum travel time, thus:

$$T_{max} \geq \sum_{i=1}^{N} \Delta t_i = \sum_{i=1}^{N} \frac{\Delta d}{v_{i,ave}} \qquad \text{Eq. (27)}$$

The objective function of the model is represented in Eq. (28) when the optimization objective is energy:

$$\begin{gathered} \text{Minimize:} \sum_{i=1}^{N} E_i(v_i^2) \\ \text{Subject to Eq. (1) to Eq. (27)} \end{gathered} \qquad \text{Eq. (28)}$$

The objective function of the model is represented in Eq. (29) when the optimization objective is time:

$$\text{Minimize:} \sum_{i=1}^{N} t_i(v_i^2) \qquad \text{Eq. (29)}$$
$$\text{Subject to Eq. (1) to Eq. (26)}$$

The speed trajectory within each section can be obtained by solving the model, and the whole speed trajectory within D is obtained by connecting the speed trajectory within each section.

3.2.2 Timetable Optimization

The total time for a train to travel in a cluster consists of running time and dwell time at each station, set the total time, running time and dwell time for a train to travel from $Station\ (n)$ to $Station(n+x)$ are $T_{n,n+x}^{total}$, $T_{n,n+x}^{running}$, $T_{n,n+x}^{dwell}$, respectively. Thus:

$$T_{n,n+x}^{total} = T_{n,n+x}^{running} + T_{n,n+x}^{dwell} \qquad \text{Eq. (30)}$$

Set the given time for a train to travel from $Station(n)$ to $Station(n+1)$ is $T_{n,n+1}^{travel}$, then the total travel time for a train from $Station(n)$ to $Station(n+x)$ would be:

$$T_{n,n+x}^{running} = \sum_{i=n}^{n+x-1} T_{i,i+1}^{running} \qquad \text{Eq. (31)}$$

Similarly, set the shortest time required for a train to run (exclude the dwell time) from $Station(n)$ to $Station(n+1)$ is $T_{n,n+1}^{min}$, then the shortest time demand from $Station(n)$ to $Station(n+x)$ would be:

$$T_{n,n+x}^{min} = \sum_{i=n}^{n+x-1} T_{i,i+1}^{min} \qquad \text{Eq. (32)}$$

The total given time in the predetermined timetable is longer than the shortest time demand for the train to travel from the first station to the last one in a cluster so as to ensure that the train is still able to arrive at the next station on time when the train spends extra dwell time at one station during rush hour. Thus:

$$T_{n,n+x}^{running} > T_{n,n+x}^{min} \qquad \text{Eq. (33)}$$

Energy-aimed Train Timetable Rescheduling Among Three Stations

Timetable optimization in each cluster is to find the most reasonable time allocation strategy, which provides minimum energy consumption. The ratio of the least time spent by the train in the two sections to the running time can be obtained by Eq. (34) for a cluster consists of three stations ($Station\ n$, $Station(n+1)$ and $Staion\ (n+2)$):

$$P_{n,n+1}^{min} = \frac{T_{n,n+1}^{min}}{T_{n,n+2}^{running}}$$
$$P_{n+1,n+2}^{min} = \frac{T_{n+1,n+2}^{min}}{T_{n,n+2}^{running}} \qquad \text{Eq. (34)}$$

The redundant time ($T_{n,n+2}^{running} - T_{n,n+2}^{min}$) is evenly divided into N^* equal parts, then the allocation of $(N^* + 1)$ time allocation possibilities can be obtained based on Eq. (35) and Eq. (36). The increase in N^* results in a larger number of samples in this step and higher prediction accuracy but reduces the speed of obtaining data. N^* is set to 20 in this research to get the balance between computation speed and accuracy. $\mathbf{MP}_{n,n+2}$ represents a series, in which each element corresponds to a time distribution of the train travelling from $Station(n)$ to $Station(n+2)$. The time distribution corresponding to the j^{th} element can be obtained by the Eq. (37) and Eq. (38):

$$Interval_{n,n+2} = \frac{1 - P_{n,n+1}^{min} - P_{n+1,n+2}^{min}}{N^*} \qquad \text{Eq. (35)}$$

$$\mathbf{MP}_{n,n+2} = P_{n,n+1}^{min} : Interval_{n,n+2} : \left(1 - P_{n+1,n+2}^{min}\right) \qquad \text{Eq. (36)}$$

$$\mathbf{T}_{n,n+1}(j) = \mathbf{MP}_{n,n+2}(j) * T_{n,n+2}^{running} \qquad \text{Eq. (37)}$$

$$\mathbf{T}_{n+1,n+2}(j) = T_{n,n+2}^{running} - \mathbf{T}_{n,n+1}(j) \qquad \text{Eq. (38)}$$

The total energy consumption of train running $Station(n)$ to $Station(n+2)$ can be expressed by Eq. (39):

$$\mathbf{E}_{n,n+2}(j) = \mathbf{E}_{n,n+1}(j) + \mathbf{E}_{n+1,n+2}(j) \qquad \text{Eq. (39)}$$

Where, $\mathbf{E}_{n,n+1}(j)$ and $\mathbf{E}_{n+1,n+2}(j)$ are computed by Eq. (28). The relationship between $P_{n,n+1}$ and the $\mathbf{E}_{n,n+2}(j)$ is shown in Figure 5. Polynomials corresponding to the calculated data and the corresponding lowest points ($P_{n,n+1}^{min}$) can be obtained by the polynomial fitting. In this study, fitting from 2^{nd} power to 8^{th} power is adopted (seven equations are obtained). The lowest points of the seven polynomials are selected to re-execute Eq. (39), in which the time allocation giving the lowest energy consumption result is selected as the optimal timetable for this set of data.

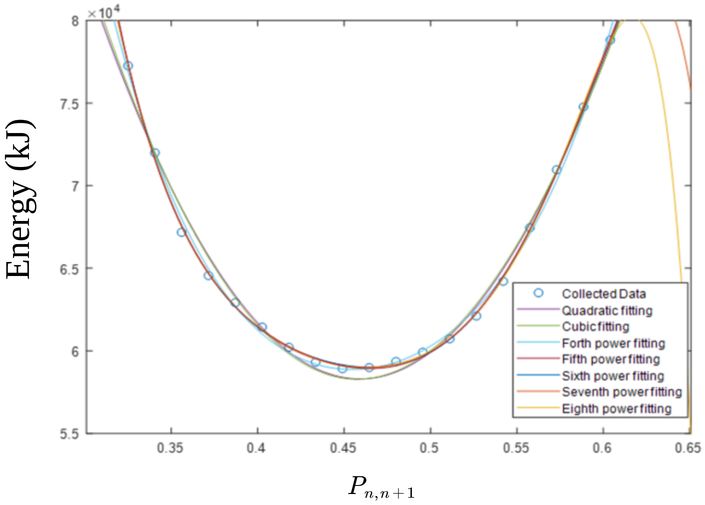


Figure 5: Example of a relationship between $P_{n,n+1}$ and total energy consumption

A total of 126,672 data points were obtained by changing the parameters such as travel weight, travel time and travel distance. The distributions of the parameters are summarized as follows:

- $Mass_{n,n+1}$ and $Mass_{n+1,n+2}$: ranges from 180 t to 240 t with 20 t as interval.
- $Dis_{n,n+1}$ and $Dis_{n+1,n+2}$: ranges from 1200 m to 6000 m with 200 m as interval.
- $T_{n,n+2}^{running}$: ranges from $\frac{Dis_{n,n+1}+Dis_{n+1,n+2}}{15m/s}$ to $\frac{Dis_{n,n+1}+Dis_{n+1,n+2}}{10m/s}$ with 20 m/s as interval.

Among these data, 70% of them are used to train the DNN, 15% are selected as verification data, and the remaining 15% are used as test data. Figure 6 shows the structure of the neural network, the input data are summarized in Table 6, the output data is $P_{n,n+1}^{min}$ for each group, and Figure 7 shows the mean squared error for the training, validation and test data sets.

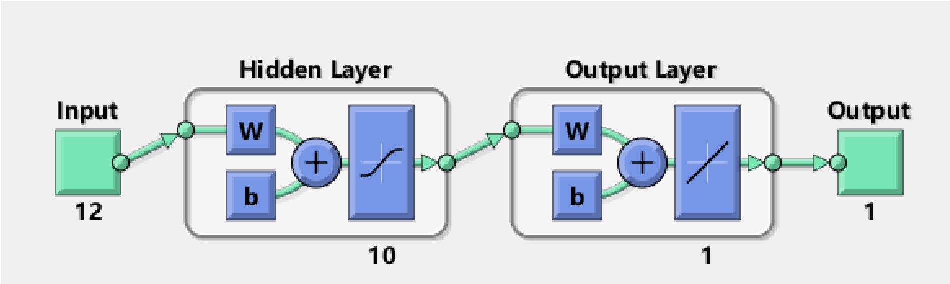


Figure 6: DNN structure

Table 6: Training input data for the DNN

$T_{n,n+2}^{running}$	$Mass_{n,n+1}$	$Mass_{n+1,n+2}$
$\frac{Mass_{n,n+1}}{Mass_{n+1,n+2}}$	$\frac{Mass_{n,n+1}}{Mass_{n,n+1} + Mass_{n+1,n+2}}$	$\frac{Mass_{n+1,n+2}}{Mass_{n,n+1} + Mass_{n+1,n+2}}$
$Dis_{n,n+1}$	$Dis_{n+1,n+2}$	$Dis_{n,n+1} + Dis_{n+1,n+2}$
$\frac{Dis_{n,n+1}}{Dis_{n+1,n+2}}$	$\frac{Dis_{n+1,n+2}}{Dis_{n,n+1} + Dis_{n+1,n+2}}$	$\frac{Dis_{n+1,n+2}}{Dis_{n,n+1} + Dis_{n+1,n+2}}$

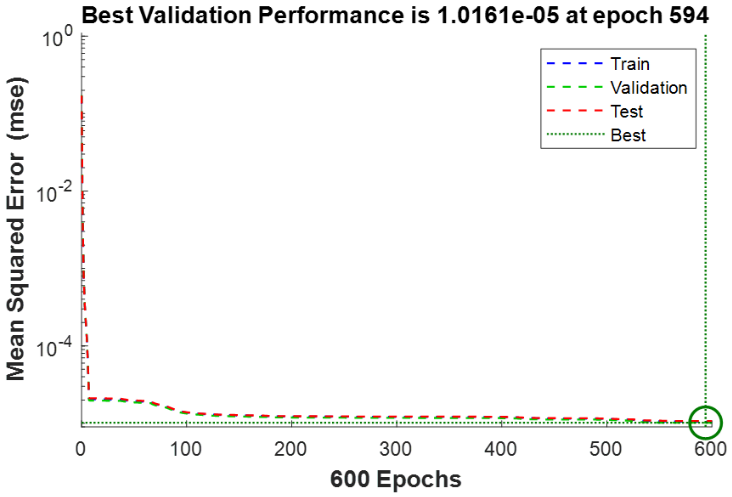


Figure 7: Mean squared error of the data sets

The coefficient of determination (R^2) is adopted to verify the prediction accuracy, which could be obtained as:

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} \quad \text{Eq. (40)}$$

where $SS_{Regression}$ is the sum of squares of the errors between the predicted value and the real value of all sample points. SS_{Total} represents the sum of squares of the difference between the average value and the real value of all sample points. Consider a data set that has n samples, for the i^{th} sample, the real value is y_i, the predicted value is y_i', and the average value of all samples is $\bar{y}$, then:

$$SS_{Regression} = \sum_{i=1}^{n}(y_i - y_i')^2 \qquad \text{Eq.}\,(41)$$

$$SS_{Total} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \qquad \text{Eq.}\,(42)$$

Thus:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad \text{Eq.}\,(43)$$

The closer the value of R^2 is to 1, the more accurate the prediction is. The predicted value is completely equal to the actual value for each sample when R^2 is equal to 1.

After training, the R^2 of the test data set is 0.9988, which is very close to 1, which shows that the trained model gives excellent results in the test data set and can find highly optimized solutions among the three stations. The mean square errors of training, verification and test data sets are all close to zero, as shown in Figure 6. It should be claimed that the distribution of data sets used in training and testing are relatively close. In practical application, if the applied cluster is obviously different from the training set data, extra training should be conducted for the applied cluster to avoid inaccurate prediction.

Energy-aimed Train Timetable Rescheduling Among More Stations

According to the results of the previous step, it can be known that using the trained DNN is able to provide highly accurate results, thus the schedule optimization of a multi-station cluster can be based on the above conclusions. For example, for any cluster with four stations and three sections, the time needs to be distributed among the three sections, so there is the following relationship:

$$\begin{aligned} T_{n,n+3}^{running} &= T_{n,n+1}^{running} + T_{n+1,n+2}^{running} + T_{n+2,n+3}^{running} \\ &= (P_{n,n+1}^{running} + P_{n+1,n+2}^{running} + P_{n+2,n+3}^{running}) * T_{n,n+3}^{running} \end{aligned} \qquad \text{Eq. (44)}$$

In the above relational expression, the principle similar to Eq. (35) and Eq. (36) can be adapted to obtain all possible proportions of all time allocated to the third section, as shown in Eq. (45) and Eq. (46):

$$Interval = (1 - P_{n,n+1}^{min} - P_{n+1,n+2}^{min} - P_{n+2,n+3}^{min})/N^* \qquad \text{Eq. (45)}$$

$$\mathbf{MP}_{n+2,n+3} = P_{n,n+1}^{min} + P_{n+1,n+2}^{min} : Interval : \left(1 - P_{n,n+1}^{min} - P_{n+1,n+2}^{min}\right) \qquad \text{Eq. (46)}$$

Benefited from the very high prediction accuracy among three stations, it is reasonable to assume that the trained neural network is able to provide an optimized $P_{n,n+1}$ and $P_{n+1,n+2}$ for any element in $\mathbf{MP}_{n+2,n+3}$. Thus a two-parameter optimization can be transformed into a one-parameter optimization by directly referring to the neural network prediction in the previous step. Considering every time distribution in the present research requires dozens of times of high complexity optimization, this transformation significantly reduces the generation time of data points. The relationship between $P_{n+2,n+3}$ and total energy consumption in each combination is obtained with the speed trajectory optimization model, and the lowest point is found by polynomial regression. Finally, the optimal time distribution in three stations could be determined. By repeatedly applying similar principles, theoretically, this method is able to obtain the optimal schedule distribution among a larger number of stations. It is worth noting that with the increase of the number of stations in a cluster, the errors of each neural network may be superimposed, resulting in reducing the overall prediction accuracy. Such errors can be alleviated by increasing the value of N^*, thus increasing the quantity and accuracy of training data.

3.3 Station Agent

This chapter demonstrates the control framework, workflow, and data collection process of the Station Agent. Three kinds of anti-disturbance mechanisms are presented in the first section, which is applied in the scenarios of different prediction accuracy. The second section introduces the process of data collection and model training.

3.3.1 Framework and Workflow

The summarized interactive framework of station agents and train agents is shown in Figure 8. As mentioned in the Proposed Multi-agent System section, station agents are separated into several clusters. Each cluster has an independent database, saves the data of card swiping, the number of passengers that are waiting on the platform, and the corresponding train dwell time collected from different types of sensors. In addition to the time-varying data, the data that does not change with time (distance between stations, track slop and curve, etc.) are also saved in the database. The affected train timetable and speed trajectory will be optimized once a disturbance is detected (or predicted). The whole process needs the cooperation of three technologies: 1. Disturbance monitoring /prediction. 2. Timetable optimization. 3. Train speed trajectory optimization. The prediction and monitoring of disturbances caused by heavy passenger flow are completed by the station agent, which is introduced in the remains of this section, and the other two processes are demonstrated in the Train agent section.

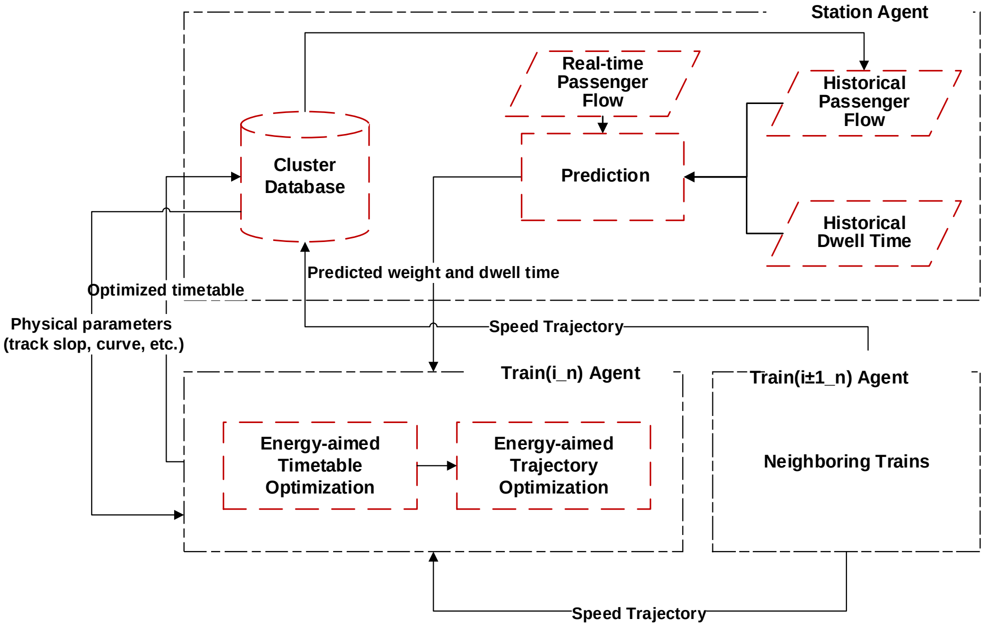


Figure 8: Control Framework of the Proposed System

Disturbances caused by heavy passenger flow consist of two types: extra dwell time and train travel weight change. The disturbance of dwell time change can be obtained by detection or prediction, while the weight change can only be obtained by prediction. Based on the detection and prediction technique, three mechanisms of coping with extra dwell time disturbance are raised. The three mechanisms are demonstrated and compared with the example shown in Figure 9. There are three trains running on Truck n with the position shown in the figure. $Train(i\_n)$ will arrive at the first station in Cluster (j) (Station D). According to the prediction, Station E will have a delay of $T_{pre}^{delay}(E)$ seconds, and the actual delay will be $T_{act}^{delay}(E)$ seconds. According to the planning timetable, the given total time of $Train\ (i\_n)$ from $Station\ D$ to $Station\ G$ is $T_{D,G}^{total}$, and the total time from $Station\ E$ to $Station\ G$ is $T_{E,G}^{total}$.

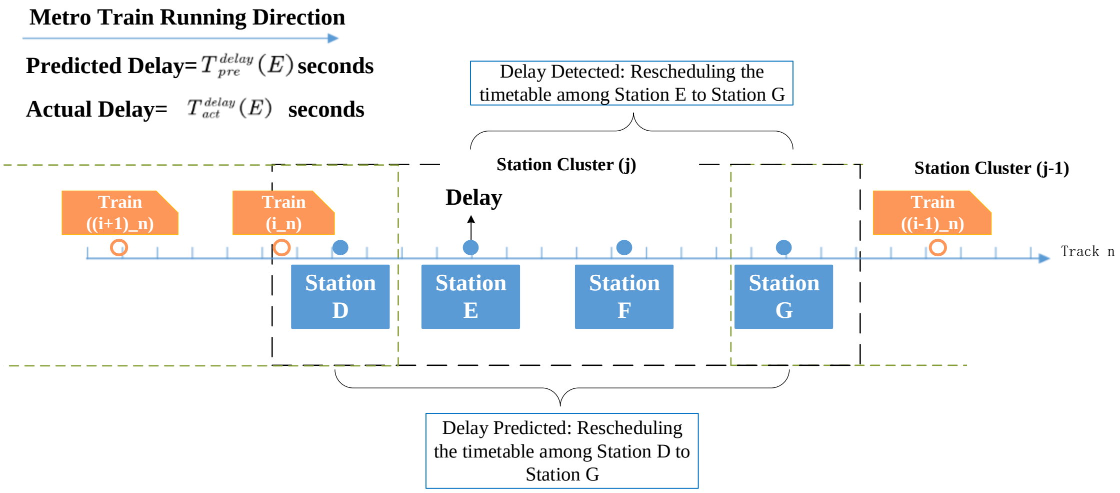

Metro Train Running Direction
Predicted Delay= $T_{pre}^{delay}(E)$ seconds
Actual Delay= $T_{act}^{delay}(E)$ seconds
Delay Detected: Rescheduling the timetable among Station E to Station G
Station Cluster (j)
Station Cluster (j-1)
Delay
Train ((i+1)_n)
Train (i_n)
Train ((i-1)_n)
Track n
Station D
Station E
Station F
Station G
Delay Predicted: Rescheduling the timetable among Station D to Station G

Figure 9: Extending Dwell Time at Station E

In the first mechanism, Train(i_n) agent allocate the total time of $T_{D,G}^{total} - T_{pre}^{delay}(E)$ seconds to the journey from Station D to Station G in the most reasonable way to reduce the extra energy cost caused by the delay. The advantage of this mechanism is that it can optimize the timetable before the occurrence of delay duration, thus the time between Station D and Station E is adjusted along with the other sections. On the premise of high prediction accuracy, this method provides an optimal solution that minimizes the extra energy consumption caused by the delay within the cluster. However, the train will travel at speed beyond the necessary speed in the process from Station D to Station E if the actual delay is less than the predicted one, which eventually causes a negative effect on the optimization performance.

Instead of $T_{pre}^{delay}(E)$, Train(i_n) agent adopt the $T_{act}^{delay}(E)$ in the second mechanism. The total Delay time $T_{act}^{delay}(E)$ is detected by the sensor when the train leaves Station E. Thus Train(i_n) agent assigns a total time of $T_{E,G}^{total} - T_{act}^{delay}(E)$ from Station E to Station G. The performance of this mechanism is more stable than the first one due to the delay is detected by sensors directly. However, this mechanism provides less energy-saving potential, for the travel time between Station D and Station E is not adjusted.

The third mechanism combines the characteristics of the previous two. The value of $T_{pre}^{delay}(E)$ is multiplied by a positive empirical coefficient γ, as shown in Eq. (47):

$$\begin{gathered} T_{pre}^{delay}(E)' = T_{pre}^{delay}(E) * \gamma \\ where, \gamma \in [0,1] \end{gathered} \qquad \text{Eq. (47)}$$

The existence of γ is to ensure the value of $T_{pre}^{delay}(E)'$ less than $T_{act}^{delay}(E)$. Train(i_n) agent first optimize the timetable among Station D to Station G with a total time of $T_{D,G}^{total} - T_{pre}^{delay}(E)'$. Then Train(i_n) follows the optimized timetable travel from Station D to Station E. The actual delay time at Station E ($T_{act}^{delay}(E)$) will be obtained when Train(i_n) left Station E, and the train optimizes the timetable again. The total energy consumption from Station D to Station G of this mechanism is less than the second one since the speed trajectory of the train from Station D to Station E. Furthermore, due to $T_{pre}^{delay}(E)' \leq T_{act}^{delay}(E)$, the risk of reducing the energy-saving due to insufficient prediction accuracy is avoided. The value of γ increases with the increase of the reliability of prediction results. γ equals zero when the prediction result is totally unreliable, and the mechanism in the third class is the same as that in the second mechanism. With the accumulation of data, the prediction accuracy is increasing. When gamma is equal to 1, the speed trajectory of the train will be the same as the first mechanism.

3.3.2 Data for Dwell Time and Travel Weight Change Prediction

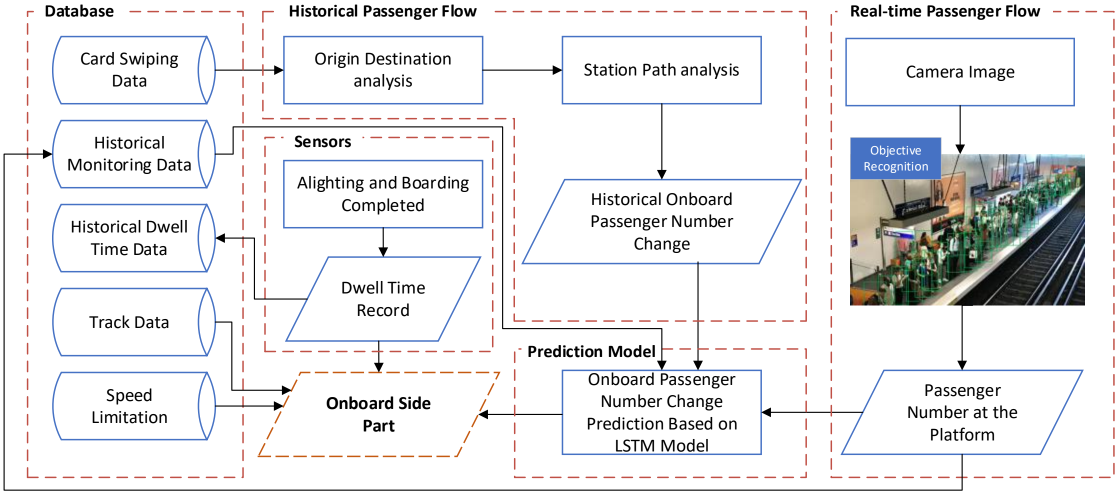
Passenger flow is the main influencing factor of train stay time and travel weight change at a station. Therefore, in order to realize the prediction of dwell time and travel weight, the first step is to predict the passenger flow. Previous studies on forecasting passenger flow mainly focused on the time interval of more than ten minutes (Bi et al. 2019; Li et al. 2019; Liu, Chen and Zhu (2018), this may be due to the highly random passenger flow in a short time, which is unfavourable to the prediction accuracy. However, the departure interval of rush time is shorter than 10 minutes for some rail transit systems, thus it is necessary to improve the prediction accuracy of the system. A prediction system combining offline and online platforms is proposed, which is expected to improve the short time interval prediction accuracy, as shown in Figure 10.

In the offline platform, the historical card swiping data of passengers can be processed based on the origin-destination (OD) analysis method. Suppose there are m stations on a certain line, and a train is heading from $Station(1)$ to $Station(m)$. If there is no transfer station, the number of passengers on the train when the train runs between $Station(n)$ and $Station\ (n+1)$ can be calculated by Eq. (48):

$$\sum_{i=1}^{n}\left(\mathbf{P_b}(i)-\mathbf{P_a}(i)\right) \qquad \text{Eq. (48)}$$

where, $\mathbf{P_b}(i)$ and $\mathbf{P_a}(i)$ represent the boarding and alighting passengers at $Station(i)$, respectively. The number of passengers can be estimated by the shortest path analysis for the scenario with the transfer station, which is not discussed here due to it is beyond the scope of this research.

The online platform is designed to collect the data of the second dimension used for training the LSTM neural network, that is, the number of people on each train and the number of people waiting on the platform. The number of waiting for passengers on the platform corresponding to the riding direction will be estimated and recorded through the collaboration of the monitoring system and object recognition algorithm. The current object recognition technology is able to count the number of passengers shown in pictures (or videos) in near real-time (such as you only look once, YOLO). As shown in the figure, these collected near real-time data provide another dimension of information for training neural networks. Because the dwell time and travel weight change of trains are directly related to the number of passengers on both sides, it is expected to increase the information of this dimension to improve the accuracy of the final prediction. In addition, the per capita weight is used to times the onboard passenger number to estimate the travel weight change of trains.



Database
Card Swiping Data
Historical Monitoring Data
Historical Dwell Time Data
Track Data
Speed Limitation
Historical Passenger Flow
Origin Destination analysis
Station Path analysis
Historical Onboard Passenger Number Change
Sensors
Alighting and Boarding Completed
Dwell Time Record
Onboard Side Part
Prediction Model
Onboard Passenger Number Change Prediction Based on LSTM Model
Real-time Passenger Flow
Camera Image
Objective Recognition
Passenger Number at the Platform

Figure 10: Work Flow of Station agent

The data are saved to the database as historical monitoring data and also sent to the LSTM prediction model as an additional input factor. Assume there are M trains passing through a non-transfer station. Set the timestamp when the $\mathrm{m^{th}}$ $(\mathrm{m} \in (0, \mathrm{M}])$ train arrive at the station is $\mathrm{t_m}$, the number of on-board passengers on the train when the train left the station is $\mathrm{NO_m}$, the predicted number of passenger on-board for the $\mathrm{m^{th}}$ train is $\widehat{\mathrm{NO_m}}$, the dwell time is DW_m, the predicted dwell time is $\widehat{DW_m}$, and the number of corresponding waiting passengers is $\mathrm{NW_m}$, the cumulative number of passengers that swiping cards from the moment that $(\mathrm{m}-1)^{\mathrm{th}}$ train leaving to $(\mathrm{m})^{\mathrm{th}}$ train leaving is $\mathrm{S_m}$, then:

$$\widehat{\mathrm{NO_m}}, \widehat{\mathrm{Dw_m}} = \mathrm{LSTM}\left(\mathrm{t_m}, \mathrm{NW_m}, \mathrm{S_m}, \widehat{\mathrm{NO_{m-1}}}\right) \qquad \mathrm{Eq.}\,(49)$$

The process is demonstrated in Figure 11.

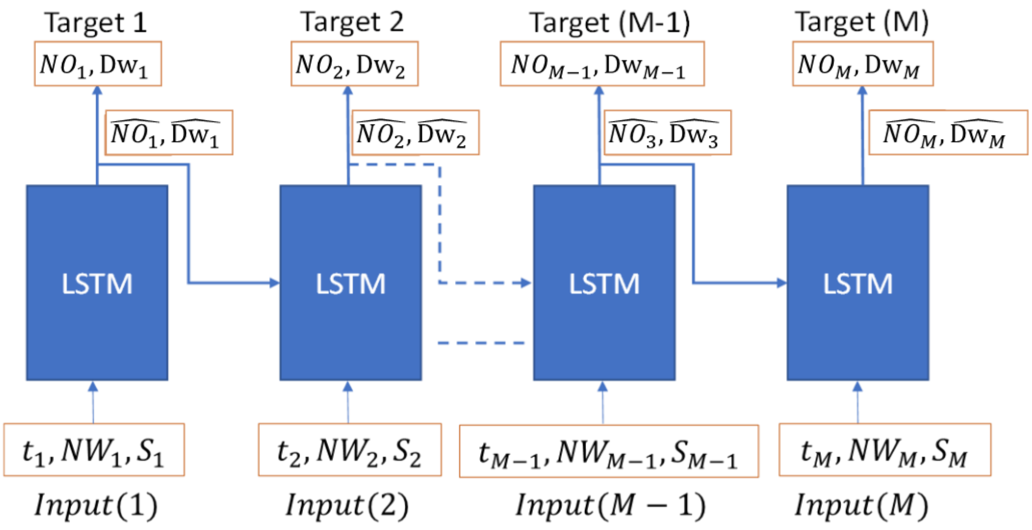

Target 1
NO_1, Dw_1
$\widehat{NO_1}, \widehat{\mathrm{Dw}_1}$
Target 2
NO_2, Dw_2
$\widehat{NO_2}, \widehat{\mathrm{Dw}_2}$
Target (M-1)
$NO_{M-1}, \mathrm{Dw}_{M-1}$
$\widehat{NO_3}, \widehat{\mathrm{Dw}_3}$
Target (M)
NO_M, Dw_M
$\widehat{NO_M}, \widehat{\mathrm{Dw}_M}$
LSTM
LSTM
LSTM
LSTM
t_1, NW_1, S_1
$Input(1)$
t_2, NW_2, S_2
$Input(2)$
$t_{M-1}, NW_{M-1}, S_{M-1}$
$Input(M-1)$
t_M, NW_M, S_M
$Input(M)$

Figure 11: Prediction of the number of on-board passengers and dwell time

In order to avoid the accuracy decrease caused by the large difference among different seasons, working days and holidays, etc, it is recommended to separate different types of data into multiple data sets and train an independent model for each of them to improve the accuracy.

3.4 Central Agent

The interaction diagram between the central agent and other agents is demonstrated in Figure 12. The central agent has sensors, a decision-making model and actuators. One sub-agent will send a signal to the central agent when it experiences a conflict with others, or the desired feedback information is not received on time. The signal will first be received by the sensors, which transfer to the decision-making model afterwards. The event code and relevant information will be analysed within the decision-making model, and then the model gives the instruction accordingly. The general rule for the decision-making process is reducing energy cost, providing safety, punctuality, and passenger comfort. The data relating to the solved conflict will be saved in the database for future analysis. The other agents have to follow the order from the central agent, even if the action will reduce their profit, for the central agent has a higher priority.

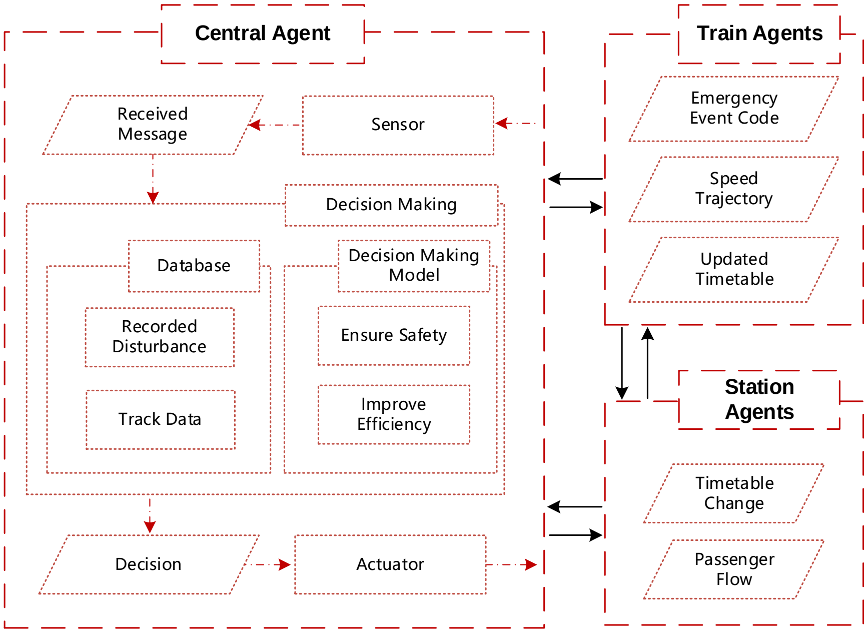


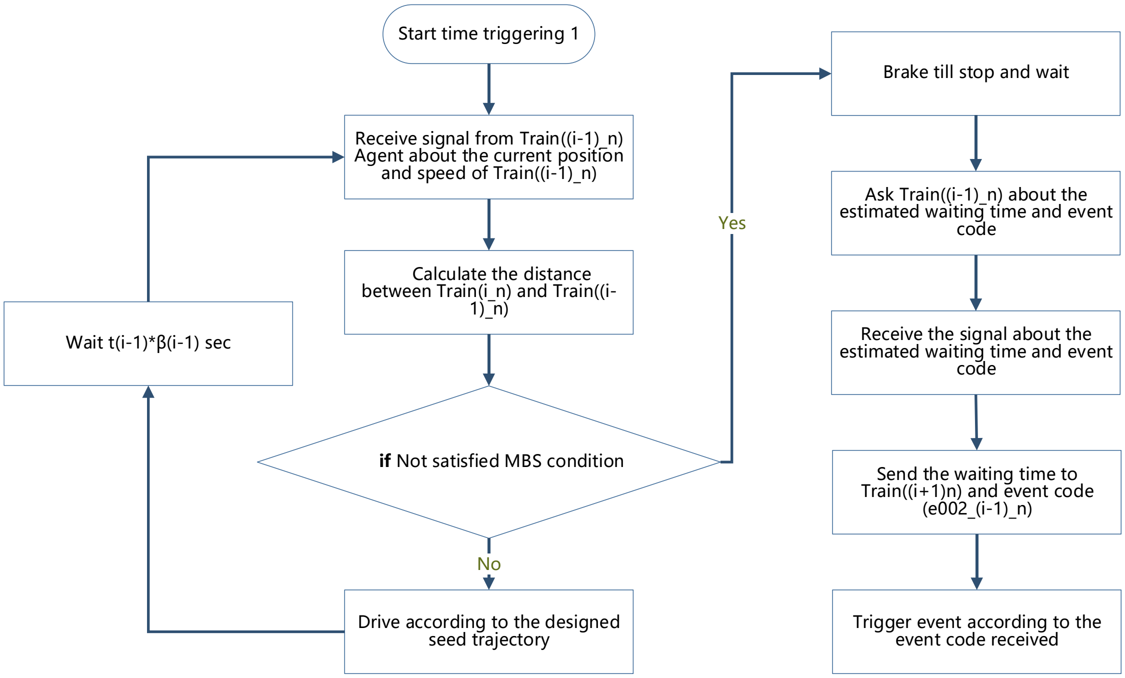
Figure 12: Interaction between central agent and other types of agents

3.5 Interaction Mechanism

According to the literature review, the adoption of high communication frequency in a multi-agent system leads to the waste of computing and communication resources. In contrast, the lower frequency may lead to the failure to meet the engineering requirements. Thus a hybrid mechanism combining TTM and ETM is adopted to ensure that the communication frequency satisfies the engineering requirements while reducing the communication frequency.

3.5.1 Time Triggering Conditions

The time triggering conditions between Train(i_n) agent with other agents are shown in Figure 13. Train(i_n) agent communicates with $Train((i-1)\_n)$ every $t(i-1)*\beta(i-1)$ seconds and with $Train((i+1)\_n)$ every $t(i)*\beta(i)$ seconds to update the environment conditions. The $t(i)$ and $t(i-1)$ is determined by the distance between trains, and the smaller distance requires a smaller communication interval between these agents. The value of $\beta(i)$, $\beta(i-1)$, and $\beta(i+1)$ is 1 when the trains are running properly, which will be decreased when unexpected situations occur to increase the communication frequency.


Start time triggering 1
Receive signal from Train((i-1)_n) Agent about the current position and speed of Train((i-1)_n)
Calculate the distance between Train(i_n) and Train((i-1)_n)
if Not satisfied MBS condition
No
Drive according to the designed seed trajectory
Wait t(i-1)*β(i-1) sec
Yes
Brake till stop and wait
Ask Train((i-1)_n) about the estimated waiting time and event code
Receive the signal about the estimated waiting time and event code
Send the waiting time to Train((i+1)n) and event code (e002_(i-1)_n)
Trigger event according to the event code received

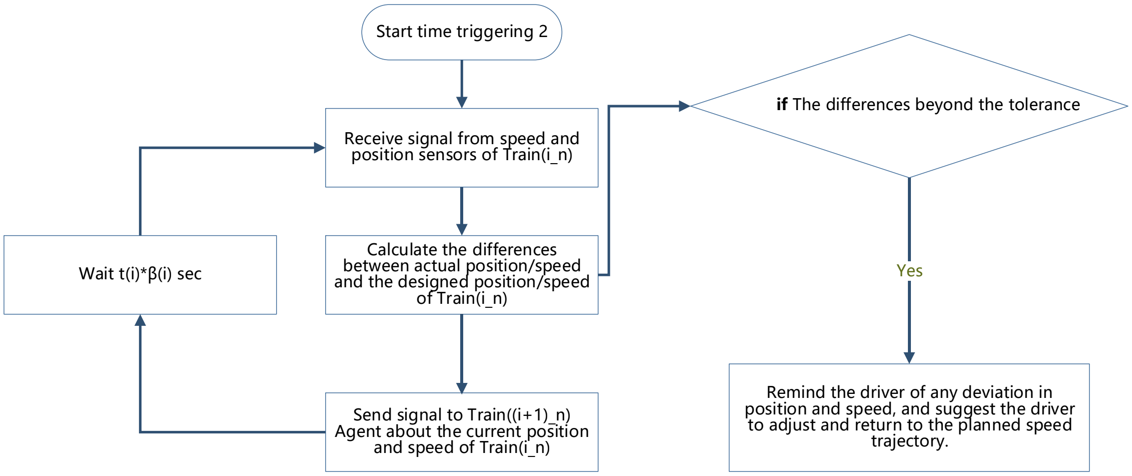

Start time triggering 2
Receive signal from speed and position sensors of Train(i_n)
Calculate the differences between actual position/speed and the designed position/speed of Train(i_n)
Send signal to Train((i+1)_n) Agent about the current position and speed of Train(i_n)
Wait t(i)*β(i) sec
if The differences beyond the tolerance
Yes
Remind the driver of any deviation in position and speed, and suggest the driver to adjust and return to the planned speed trajectory.

Figure 13: Time Triggering pseudocode for Train(i_n)

The time triggering conditions and pseudocode for station agents is demonstrated in Figure 14 by selecting Station agent E as an example.

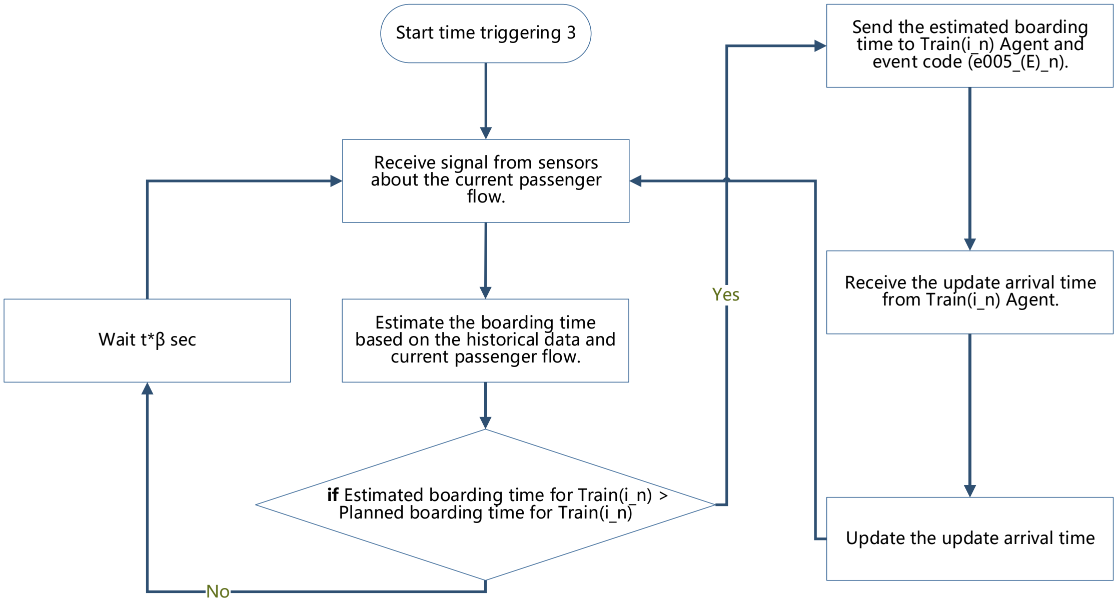

Start time triggering 3
Receive signal from sensors about the current passenger flow.
Estimate the boarding time based on the historical data and current passenger flow.
if Estimated boarding time for Train(i_n) > Planned boarding time for Train(i_n)
No
Yes
Wait t*β sec
Send the estimated boarding time to Train(i_n) Agent and event code (e005_(E)_n).
Receive the update arrival time from Train(i_n) Agent.
Update the update arrival time

Figure 14: Time Triggering pseudocode for Station E

3.5.2 Event Triggering Conditions

Besides time triggering mechanisms, each agent needs extra communication when the system meets some unexpected situations. The communication mechanism between them is triggered by some event code in such cases. Table 7 summarizes those codes and the corresponding events. The rest of this section will explain each event in detail. The processes of those mechanisms are shown from Figure 15 to Figure 20.

Table 7: Summary of event code and relevant situation

Event Code	Represented Situation	Send From	Received By
e001_(i)_n	Train(i_n) has to stop for a period of time due to some unexpected reasons	Train(i_n) Driver	Train(i_n) Agent
e002_(i-1)_n	Train((i-1)_n) adopted a new speed trajectory	Train((i-1)_n) Agent/Central Agent	Train(i_n) Agent/Central Agent
e003_(i)_n	Train(i_n) is not able to switch track, speed trajectory optimization should be processed	Train(i_n) Agent	Train(i_n) Agent
e004_(i)_n	Train(i_n) is not able to follow the original speed trajectory. The Central agent needs to check whether Train(i_n) can switch tracks.	Train(i_n) Agent	Central Agent
e005_(E)_n	The estimated boarding time is longer than the boarding time given in the planned timetable.	Station agent E	Train((i+1)_n)

Event triggering with code e001_(i)_n:

If the Train(i_n) Driver encounters an emergency in the process of travelling (such as line fault) and needs to stop, the driver will send a signal to Train(i_n) agent with an event code: “e001_(i)_n“. The signal includes the estimated duration for the temporary parking. The agent will then proceed with the process shown in Figure 15, by which to generate an optimized new speed trajectory.

In the process of speed trajectory optimization, safety, passenger comfort, and parking accuracy are taken as constraints. Optimization objectives are punctuality and energy consumption. Usually, giving a longer time for a train to run between two stations will reduce energy consumption. Therefore, for delayed trains, the two optimization objectives of punctuality and energy consumption are contradictory. For rail transit systems, punctuality is usually prior to saving energy. Therefore, the proposed method takes punctuality as a higher priority than energy saving. As shown in Figure 15, when Train(i_n) agent receives the event code “e001_(i)_n“, the agent first optimizes the speed trajectory by minimizing the travel time. If the possible earliest arrival time is earlier than the planned arrival time, Train(i_n) agent will optimize the speed trajectory by minimizing the energy consumption with the planned arrival time as constraints. Otherwise, the possible earliest arrival time will be adopted as the constraints instead.

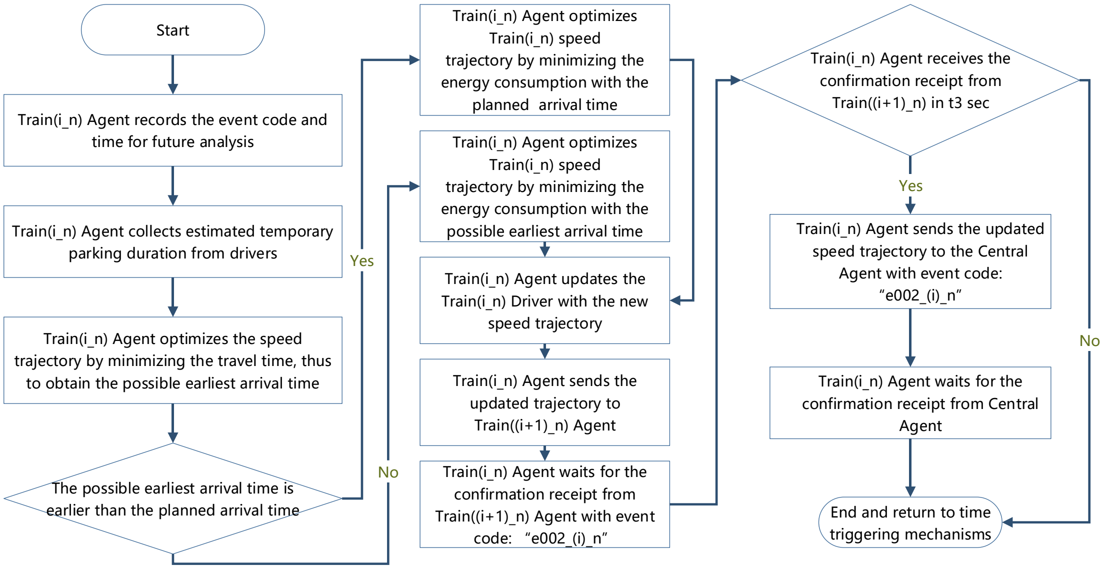

Start
Train(i_n) Agent records the event code and time for future analysis
Train(i_n) Agent collects estimated temporary parking duration from drivers
Train(i_n) Agent optimizes the speed trajectory by minimizing the travel time, thus to obtain the possible earliest arrival time
The possible earliest arrival time is earlier than the planned arrival time
Yes
No
Train(i_n) Agent optimizes Train(i_n) speed trajectory by minimizing the energy consumption with the planned arrival time
Train(i_n) Agent optimizes Train(i_n) speed trajectory by minimizing the energy consumption with the possible earliest arrival time
Train(i_n) Agent updates the Train(i_n) Driver with the new speed trajectory
Train(i_n) Agent sends the updated trajectory to Train((i+1)_n) Agent
Train(i_n) Agent waits for the confirmation receipt from Train((i+1)_n) Agent with event code: "e002_(i)_n"
Train(i_n) Agent receives the confirmation receipt from Train((i+1)_n) in t3 sec
Yes
No
Train(i_n) Agent sends the updated speed trajectory to the Central Agent with event code: "e002_(i)_n"
Train(i_n) Agent waits for the confirmation receipt from Central Agent
End and return to time triggering mechanisms

Figure 15: Process for Train(i_n) when event code "e001_(i)_n" is received

Event triggering with code e002_(i-1)_n :

Train(i_n) only needs to travel according to its predefined speed trajectory when Train((i-1)_n) travels properly (the difference between the speed and position of the train at the corresponding time and the original plan is within the allowable error range, and the Central agent or driver does not send any instructions to change the driving plan). However, if the Train(i-1_n) encounters an emergency in the process of travelling (such as line fault) and needs to stop, the Train((i-1)_n) agent is triggered by this event to send signals to the Train(i_n) Agent. Then Train(i_n) agent needs to make decisions and necessary optimization according to the new speed trajectory of Train((i-1)_n). The process for the corresponding decision and optimization process is shown in Figure 16. During the process, Train(i_n) agent first determines whether the original speed trajectory of Train(i_n) still meets the safety requirements of MBS shown in Eq. (50) and Eq. (51).

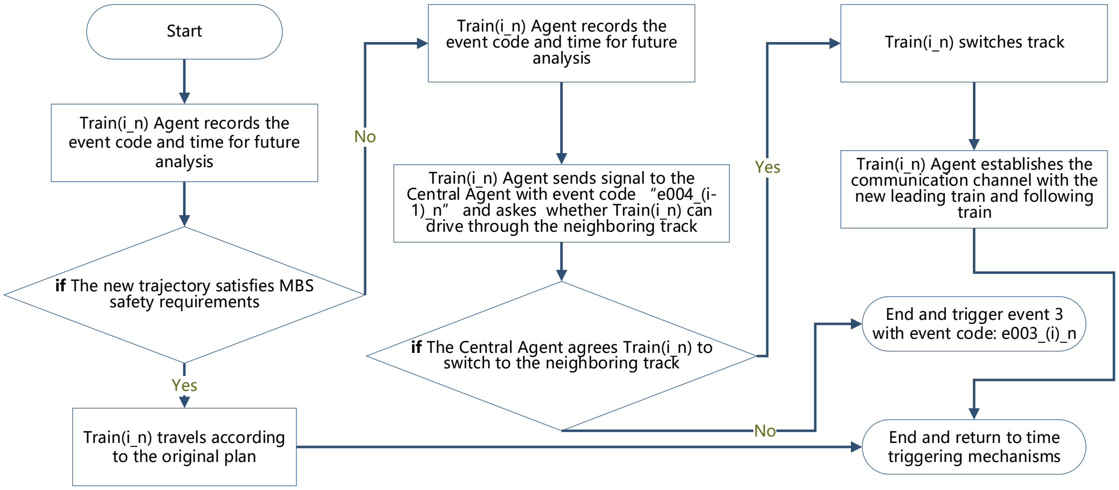

Start
Train(i_n) Agent records the event code and time for future analysis
if The new trajectory satisfies MBS safety requirements
No
Yes
Train(i_n) travels according to the original plan
Train(i_n) Agent records the event code and time for future analysis
Train(i_n) Agent sends signal to the Central Agent with event code "e004_(i-1)_n" and askes whether Train(i_n) can drive through the neighboring track
if The Central Agent agrees Train(i_n) to switch to the neighboring track
Yes
No
Train(i_n) switches track
Train(i_n) Agent establishes the communication channel with the new leading train and following train
End and trigger event 3 with event code: e003_(i)_n
End and return to time triggering mechanisms

Figure 16: Process for Train(i_n) when event code “e002_(i-1)_n “ is received

If the distance between Train(i_n) and Train((i-1)_n) meet the MBS requirement, it indicates that the original speed trajectory of Train(i_n) still meets the safety requirements. Considering that the original speed trajectory was obtained through highly offline optimization, thus Train(i_n) would follow the original speed trajectory and schedule without further optimization. If not, Train(i_n) needs to adjust the original driving strategy to meet the safety requirements. There are two possibilities for adjustment. The first is to switch to a parallel runway in the same direction. The advantage of this method is that the original speed trajectory can be maintained as much as possible to avoid train delays and energy waste. Train(i_n) communicates with the Train((i±1)_(n±1)) Agents to apply the track switch. If it is found in this step that the switching will not affect the normal running of trains in adjacent tracks in the same direction, and the track position can be switched, Train(i_n) agent will reach an agreement with the corresponding train agents in adjacent tracks and switch track to avoid losses caused by the re-planning of speed trajectory. If running through another track will affect the normal running of Trains on that track, the other re-planning method will be adopted instead. Train(i_n) will then maintain the current track and adjust the speed trajectory. Then Train(i_n) agent will trigger event 3 with event code “e003_(i)_n“ to optimize the speed trajectory of Train(i_n) based on the collected data.

Event triggering with code e003_(i-1)_n:

Based on the collected data about the new speed trajectory of Train((i-1)_n), Train(i_n) agent first computes the critical moment in this event. The critical moment has three characteristics: 1) the critical moment occurs between the time the Train ((i-1)_n) starts and completes re-acceleration. 2) The speed of Train((i-1)_n) equals Train(i_n) at the moment, which is known as critical speed. 3) MBS requirement is critically satisfied, i.e., Eq. (50) and Eq. (51) are critically satisfied. As a result, the critical speed is the highest allowable speed at the moment, and kinetic energy can be saved. The critical velocity is marked in Figure 26 to help understand this concept. Train(i_n) agent then optimizes the speed trajectory before and after Train(i_n) achieves the critical speed by using MILP algorithm with constraints given in the Train agent section and sends the event code “e002_(i)_n“ to Train((i+1)_n) agent and waits for the receipt from it. It reflects a potential failure in the communication channel between Train((i+1)_n) agent and Train(i_n) agent if the confirmation is not received in time. In that case, Train(i_n) agent sends event code “e002_(i)_n“ to the Central Agent, which takes the responsibility to transfer the information among these two train agents. This procedure is demonstrated in Figure 17.

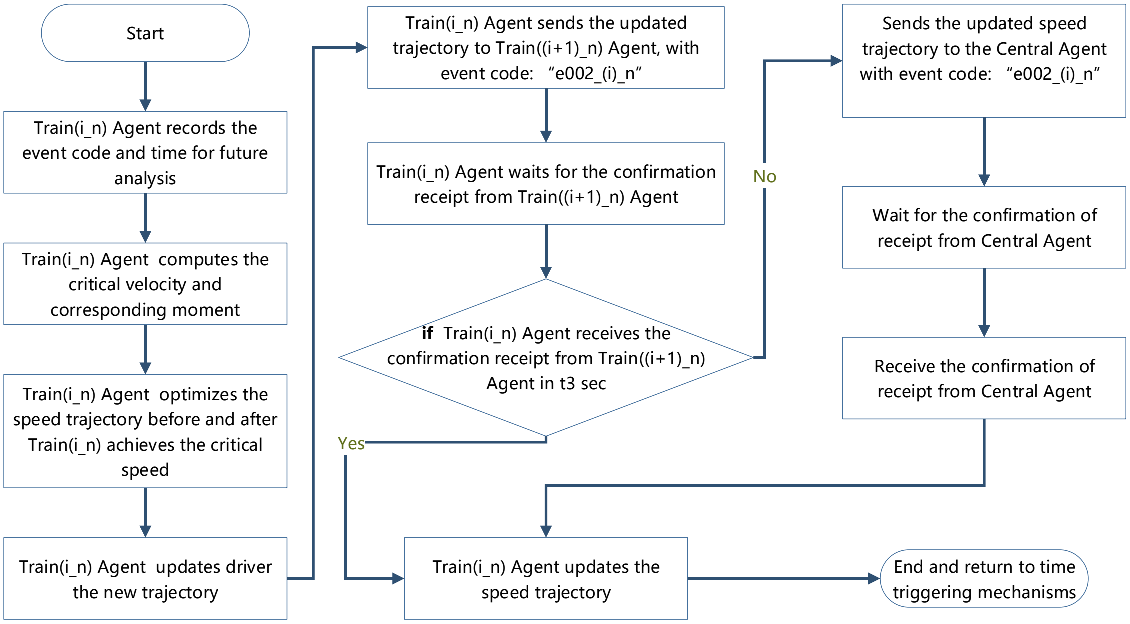

Start
Train(i_n) Agent records the event code and time for future analysis
Train(i_n) Agent computes the critical velocity and corresponding moment
Train(i_n) Agent optimizes the speed trajectory before and after Train(i_n) achieves the critical speed
Train(i_n) Agent updates driver the new trajectory
Train(i_n) Agent sends the updated trajectory to Train((i+1)_n) Agent, with event code: "e002_(i)_n"
Train(i_n) Agent waits for the confirmation receipt from Train((i+1)_n) Agent
if Train(i_n) Agent receives the confirmation receipt from Train((i+1)_n) Agent in t3 sec
No
Yes
Sends the updated speed trajectory to the Central Agent with event code: "e002_(i)_n"
Wait for the confirmation of receipt from Central Agent
Receive the confirmation of receipt from Central Agent
Train(i_n) Agent updates the speed trajectory
End and return to time triggering mechanisms

Figure 17: Process for Train(i_n) when event code "e003_(i-1)_n " is received

Event triggering with code e002_(i)_n :

When the Train(i_n) agent changes the driving strategy and sends a new speed trajectory to the Train((i+1)_n) Agent, the Train(i_n) agent waits for a receipt from the Train((i+1)_n) agent to determine that the Train((i+1)_n) agent successfully received the information sent by Train((i-1)_n). So as to avoid the potential risk caused by the failure of information delivery. If the Train(i_n) agent does not receive the expected information within the specified time, it will send the corresponding event code to the Central agent and request the Central agent to assist the communication (as shown in Figure 16 and Figure 18). The process is shown in Figure 18.

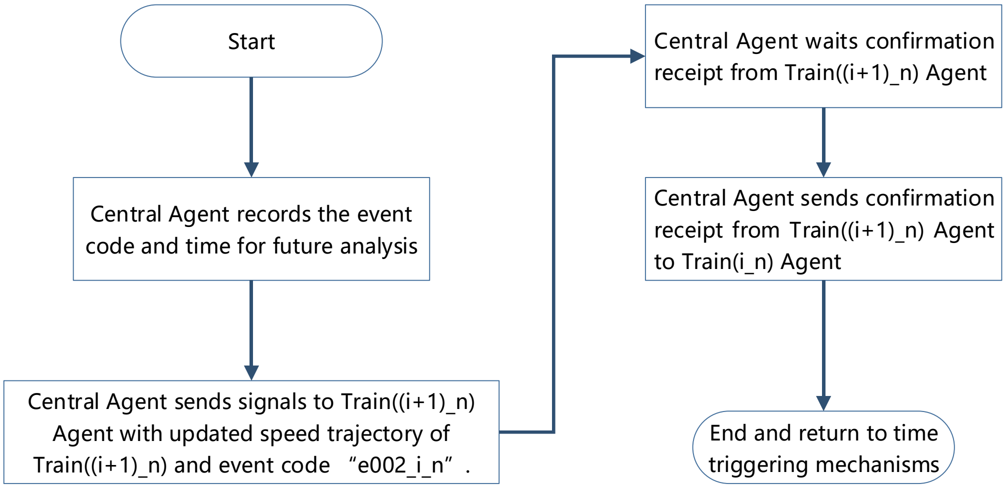

Start
Central Agent records the event code and time for future analysis
Central Agent sends signals to Train((i+1)_n) Agent with updated speed trajectory of Train((i+1)_n) and event code "e002_i_n".
Central Agent waits confirmation receipt from Train((i+1)_n) Agent
Central Agent sends confirmation receipt from Train((i+1)_n) Agent to Train(i_n) Agent
End and return to time triggering mechanisms

Figure 18: Process for the Central agent when event code “e002_(i)_n“ is received

Event triggering with code e004_(i)_n:

When the Central agent receives a signal from Train(i_n) agent with event code “e004_(i)_n“, the Central agent needs to detect whether the trains running on the neighbouring track meets the condition of switching tracks. The Central agent first compares the speed trajectories of Train(i_n) and the trains travelling on the adjacent tracks. If those trajectories meet MBS requirements, then the switching won’t affect the driving plan of the trains running on neighbouring tracks. Thus, the Central agent will send a signal to Train(i_n) Agent, allowing Train(i_n) to switch to an adjacent track to travel. However, if the Central agent finds that the MBS requirements are not satisfied by those trajectories, the Center agent informs Train(i_n) agent that Train(i_n) cannot switch the track. The corresponding process is shown in Figure 19.

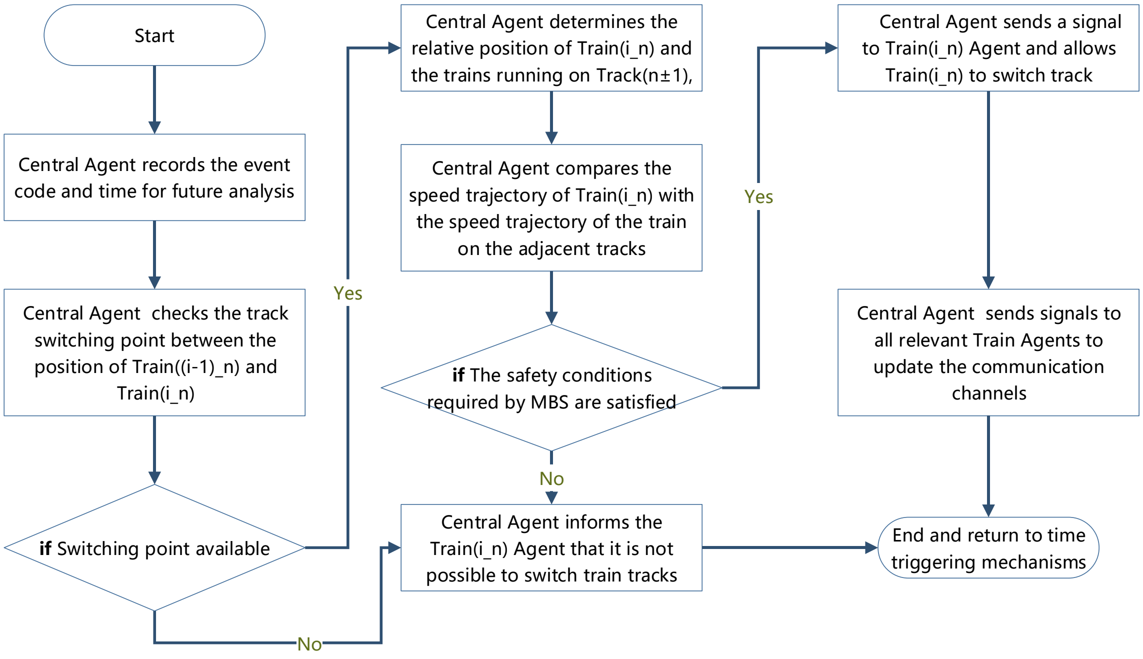

Start
Central Agent records the event code and time for future analysis
Central Agent checks the track switching point between the position of Train((i-1)_n) and Train(i_n)
if Switching point available
Yes
No
Central Agent determines the relative position of Train(i_n) and the trains running on Track(n±1),
Central Agent compares the speed trajectory of Train(i_n) with the speed trajectory of the train on the adjacent tracks
if The safety conditions required by MBS are satisfied
Yes
No
Central Agent informs the Train(i_n) Agent that it is not possible to switch train tracks
Central Agent sends a signal to Train(i_n) Agent and allows Train(i_n) to switch track
Central Agent sends signals to all relevant Train Agents to update the communication channels
End and return to time triggering mechanisms

Figure 19: Process for the Central agent when event code “e004_(i)_n“ is received

Event triggering with code e005_(E)_n:

Based on the literature review, it is found that computational demand for timetable optimization is higher than that for train speed trajectory optimization, which usually relies on an offline optimization process. One constraint has to be fulfilled during the optimization, which is that the given running time between any two stations has to be longer than the possible minimum time that the train can achieve. When the actual boarding time in a station exceeds the planned boarding time, the train is still able to arrive at the next station on time by adjusting the speed trajectory with a higher average speed.

As shown in Figure 4, if the actual departure time of Train ((i+1)_n) at Station E is later than the planned departure time due to heavy passenger flow, the train will need to speed up to accomplish a “catch up task“ to ensure punctuality when arriving at Station F. As a result, more energy is consumed due to the higher average speed. In the proposed system, a timetable optimization is applied to minimize energy consumption among Stations D, E and F while ensuring punctuality at Station F. The corresponding procedure is shown in Figure 20, and the corresponding optimization algorithm can be found in section 3.2.2.

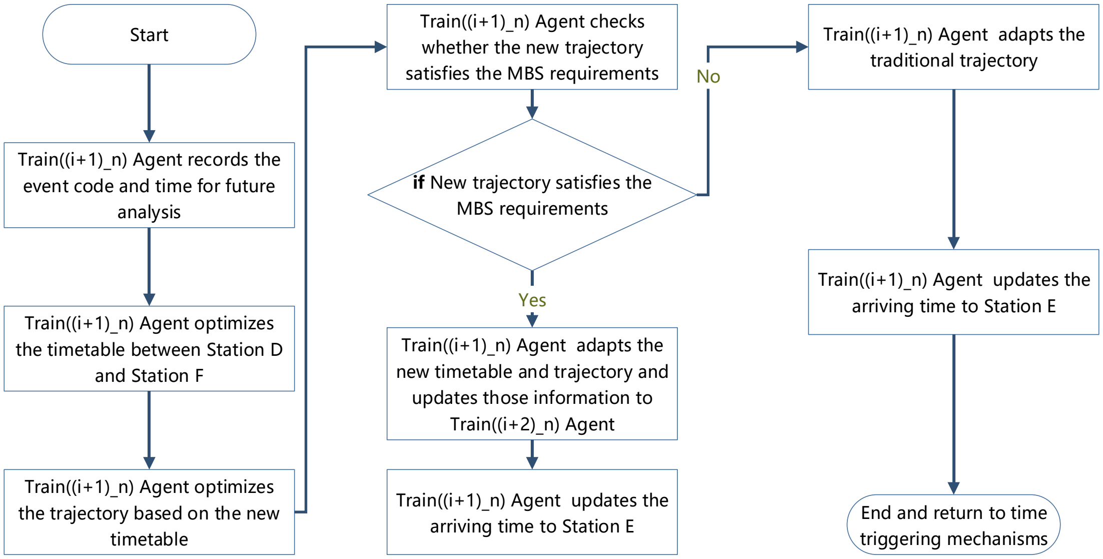

Start
Train((i+1)_n) Agent records the event code and time for future analysis
Train((i+1)_n) Agent optimizes the timetable between Station D and Station F
Train((i+1)_n) Agent optimizes the trajectory based on the new timetable
Train((i+1)_n) Agent checks whether the new trajectory satisfies the MBS requirements
if New trajectory satisfies the MBS requirements
Yes
No
Train((i+1)_n) Agent adapts the new timetable and trajectory and updates those information to Train((i+2)_n) Agent
Train((i+1)_n) Agent updates the arriving time to Station E
Train((i+1)_n) Agent adapts the traditional trajectory
Train((i+1)_n) Agent updates the arriving time to Station E
End and return to time triggering mechanisms

Figure 20: Process for the Train((i+1)_n) agent when event code “e005_(E)_n“ is received

3.5.3 MBS Based Safety Constraints

The block system is widely adopted to ensure safety in rail transit, which consists of Fix Block System (FBS) and Moving Block System (MBS) (Lu, Song, and Li 2007). The MBS is adopted in this research, for it guarantees a higher control efficiency by providing smaller spacing. The principle of the adopted MBS is shown in Figure 21. Any two neighbouring trains within such a system should guarantee that the following train should be able to fully stop outside of a pre-set safety distance through service braking whenever the leading train takes the emergency braking to stop. Considering that two trains with different braking abilities may simultaneously run on the same track, two rules are set to further improve safety.

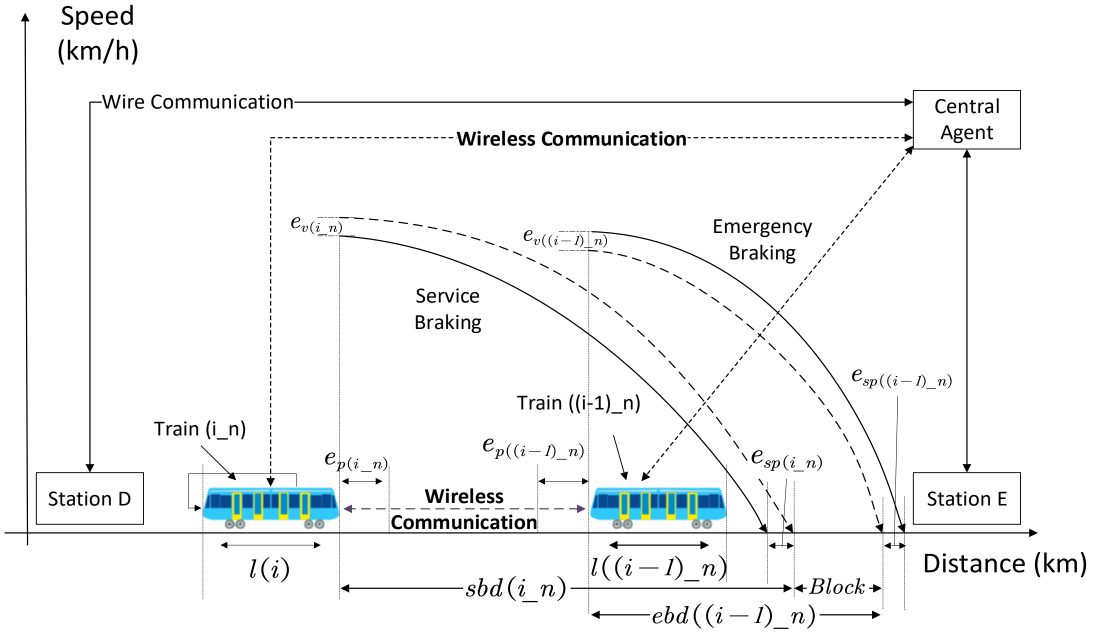


Figure 21: Principle of the Adopted Safety Constraints (MBS Based)

The first rule is set to avoid potential risk caused by inaccuracy sensors. All errors are assumed to cause the predicted train spacing to be larger than the actual one. Thus the distance between any two adjacent trains when they entirely stop is always greater than or equal to the pre-set block if an emergency situation occurs. As shown in Figure 21, at the moment $sbd(i\_n)$ and $ebd((i-1)\_n)$ are the service braking distance of Train(i_n) and the emergency braking distance of Train((i-1)_n), respectively. The errors of running position, speed, complete stop position are represented by e_p, e_v, and e_{sp}.

The other rule is set to avoid the risk caused by the difference in speed limit and braking capacity of different train models. It is activated when the emergency braking distance of the leading train is larger than the service braking distance of the following train. The value of $ebd((i-1)_n) - sbd(i\_n)$ is assumed to be 0 in such a case (see Eq. (50) and Eq. (51) and (6) for details). SBD(i_n) and EBD((i-1)_n) are two arrays, which respectively represents the service braking distance of train(i_n) and the emergency braking distance of train(i-1)_n) during the travel duration, and S_{safe} is the pre-set safety distance.

$$\min\left(\mathrm{TP(i-1\_n)-TP(i\_n)}+\left(\mathrm{EBD(i-1\_n)-SBD(i\_n)}\right)^{-}\right)\geq S_{safe} \qquad \text{Eq. (50)}$$

$$\left(EBD(i-1\_n)-SBD(i\_n)\right)^{-}=\begin{cases}0 & \text{if} > 0\\ EBD(i-1\_n)-SBD(i\_n) & \text{if} \leq 0\end{cases} \qquad \text{Eq. (51)}$$

Chapter 4: Performance Evaluation and Discussion

This chapter verifies the energy-saving effectiveness of the proposed system by simulating four types of disturbances. The first case is the situation that a train in the system needs to stop temporarily when it encounters an unexpected situation, and this temporary stop will cause the following train to fail to follow the planned speed trajectory. The purpose of this case is to test the energy-saving performance of the proposed system when suddenly encountering a disturbance of temporary parking of a train. The second and the third case study tests the extra dwell time and train travel weight change disturbance in a station caused by large passenger flow, respectively. The purpose of these two cases is to test the energy-saving performance of the proposed system under the situation of only facing delay disturbance and travel weight change disturbance, respectively. Furthermore, results of the third case are also adopted to compare with the second case to determine which kind of disturbance will cause more energy consumption. The final case tests the system with the combination disturbance of delay and travel weight change to discover whether the proposed system is able to complete the optimization quickly when it encounters a combined disturbance. Each case includes numerous scenarios to prove that the proposed system can effectively save energy and meet the needs of near real-time optimization in practice.

Matlab is the coding platform to realize the agent, in which the neural network toolbox and CPLEX optimizer are adopted, respectively. The adopted hardware during the test is i7 8700 CPU, 16G RAM, and RTX 2060 GPU.

4.1 Case Studies

4.1.1 Case Study 1: Disturbance of Unexpected Braking

The purpose of this case is to test the energy-saving performance of the following train in the proposed system when a train makes a temporary emergency stop due to an unexpected situation. The Scenario is shown in Figure 22, where three trains are travelling from Station A to Station B with a travel distance of 3.65 km. Train(1_1) is the leading train, followed by Train(2_1) and Train(3_1). The weight of each train is 230 tonnes, and the maximum speed limit in the section is 30 m/s. Each train is constrained to meet MBS requirements with a safety distance of 500 metres. The departure times of the three trains are t=0 (s), t=100 (s) and t=200 (s), respectively, and the planned travel time between Station A and Station B is 250 seconds for each train. For the first 1500m, all trains travel according to the offline optimized timetable and speed trajectory, and the involved Agents exchange information by following the mechanics presented in the proposed methodology section. Each leading train continually sends signals to the following train. The signal contains information about the current position, planning speed trajectory, and driving status of the leading train, as shown in the process 1~2 in Figure 23.

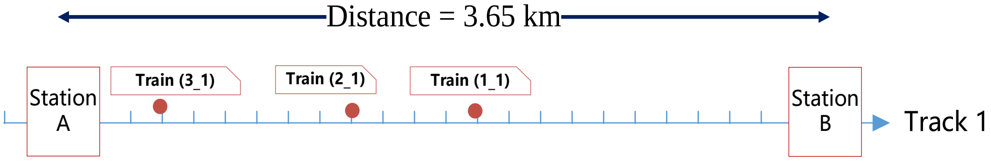


Figure 22: Distribution of Trains and Stations in Case 1

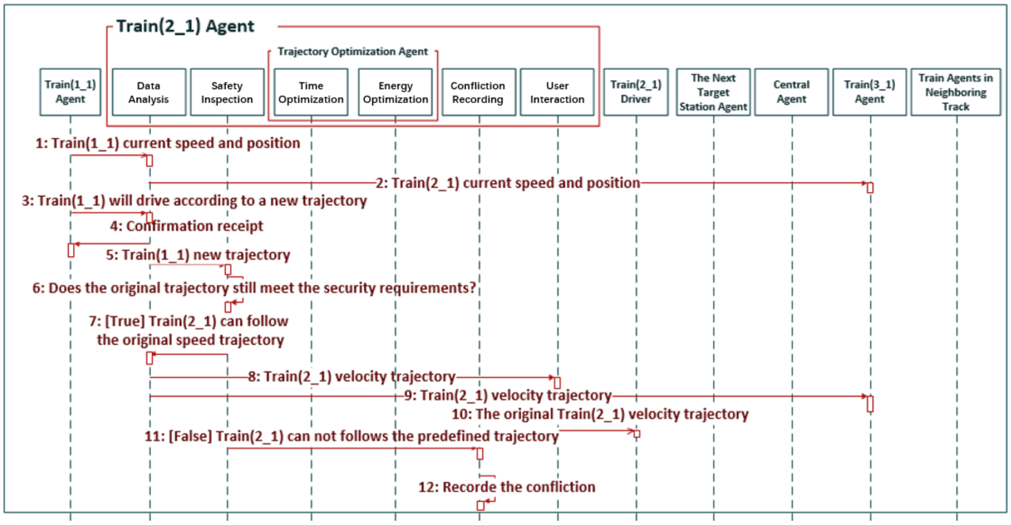

Train(2_1) Agent
Trajectory Optimization Agent
Train(1_1) Agent
Data Analysis
Safety Inspection
Time Optimization
Energy Optimization
Confliction Recording
User Interaction
Train(2_1) Driver
The Next Target Station Agent
Central Agent
Train(3_1) Agent
Train Agents in Neighboring Track
1: Train(1_1) current speed and position
2: Train(2_1) current speed and position
3: Train(1_1) will drive according to a new trajectory
4: Confirmation receipt
5: Train(1_1) new trajectory
6: Does the original trajectory still meet the security requirements?
7: [True] Train(2_1) can follow the original speed trajectory
8: Train(2_1) velocity trajectory
9: Train(2_1) velocity trajectory
10: The original Train(2_1) velocity trajectory
11: [False] Train(2_1) can not follows the predefined trajectory
12: Recorde the confliction

Figure 23: Sequence Diagram for the Multi-agent Interaction: Part 1

At t=100 seconds, the driver of Train(1_1) detects an emergency and estimates that a temporary stop of 80 seconds will be required to deal with the situation. The driver transmits the corresponding information to the Train(1_1) Agent, and the Train(1_1) agent re-optimizes the speed trajectory according to the built-in algorithm and updates the new trajectory to the Train(2_1) Agent. After receiving the signal from Train(1_1) Agent, Train(2_1) agent discovered that the planned speed trajectory of Train(2_1) unable to satisfy the constraints of MBS. Therefore, it starts to send a signal to the Central agent consulting if it can switch to the adjacent trajectory. Then the Central agent rejected the request due to there are no adjacent tracks in this scenario. The above process is illustrated in Figure 24.

In order to meet the safety requirements of MBS, Train(2_1) must re-optimize the speed trajectory by the MILP algorithm with the constraints given by the new trajectory of Train(1_1) and MBS condition, and Train(2_1) will be delayed due to this change. The trajectory optimization process is shown in Figure 25.

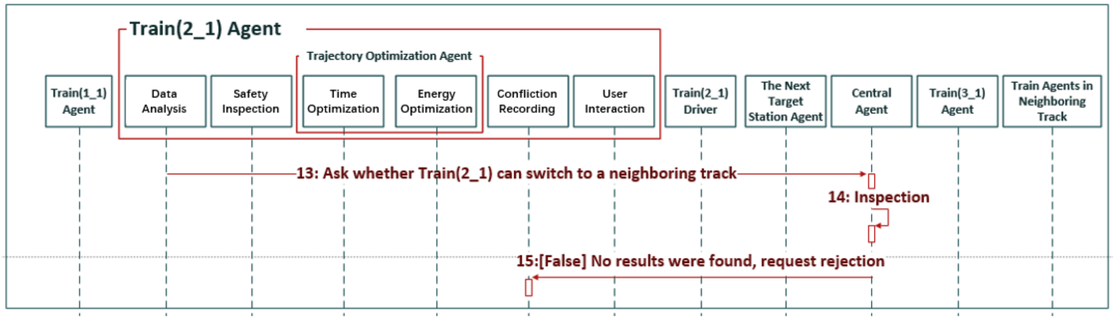

Train(2_1) Agent
Trajectory Optimization Agent
Train(1_1) Agent
Data Analysis
Safety Inspection
Time Optimization
Energy Optimization
Confliction Recording
User Interaction
Train(2_1) Driver
The Next Target Station Agent
Central Agent
Train(3_1) Agent
Train Agents in Neighboring Track
13: Ask whether Train(2_1) can switch to a neighboring track
14: Inspection
15:[False] No results were found, request rejection

Figure 24: Sequence diagram for the Multi-agent Interaction: Part 2

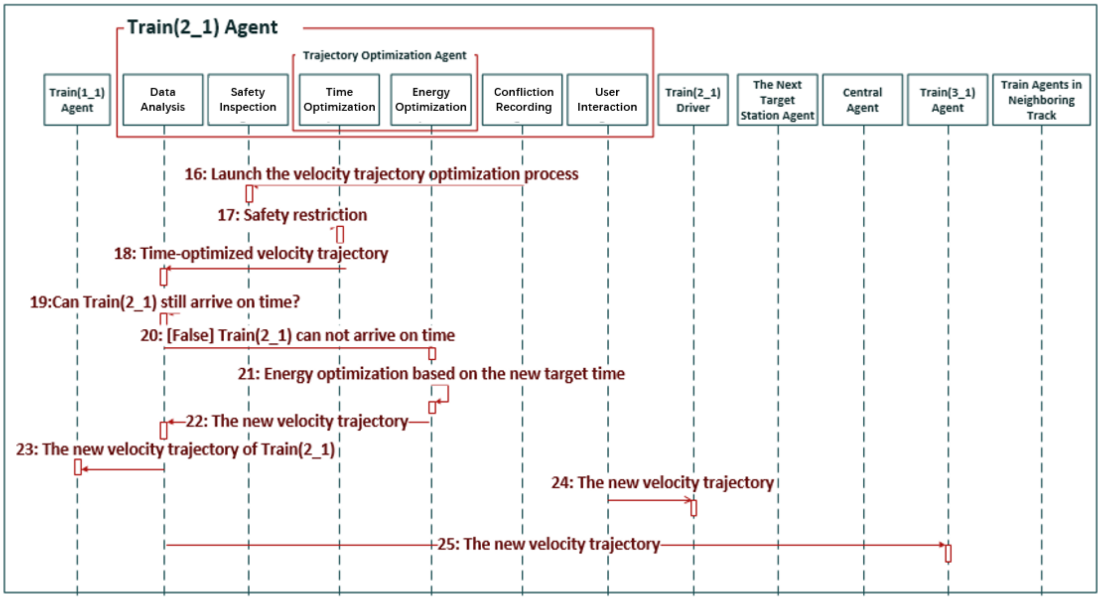
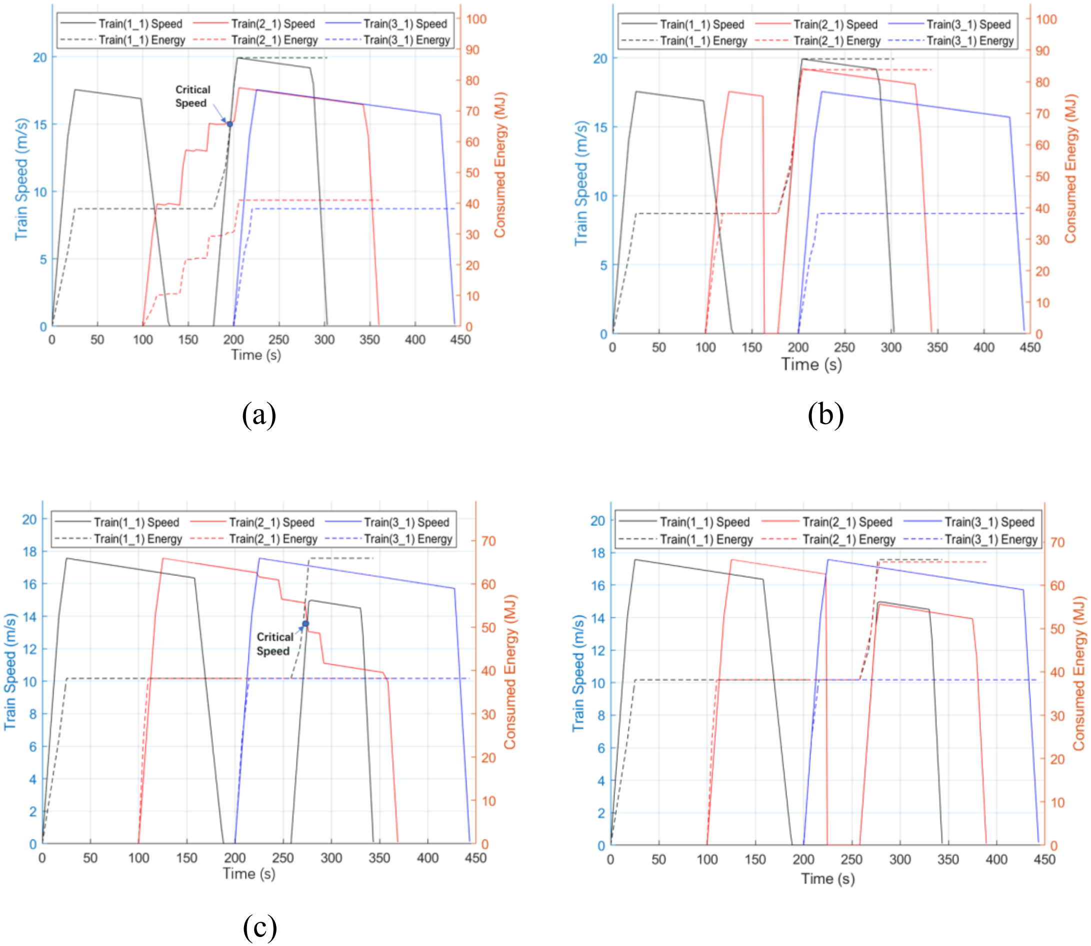

Train(2_1) Agent
Trajectory Optimization Agent
Train(1_1) Agent
Data Analysis
Safety Inspection
Time Optimization
Energy Optimization
Confliction Recording
User Interaction
Train(2_1) Driver
The Next Target Station Agent
Central Agent
Train(3_1) Agent
Train Agents in Neighboring Track
16: Launch the velocity trajectory optimization process
17: Safety restriction
18: Time-optimized velocity trajectory
19:Can Train(2_1) still arrive on time?
20: [False] Train(2_1) can not arrive on time
21: Energy optimization based on the new target time
22: The new velocity trajectory
23: The new velocity trajectory of Train(2_1)
24: The new velocity trajectory
25: The new velocity trajectory

Figure 25: Sequence Diagram for the Multi-agent Interaction: Part 3

Train(2_1) then run based on the new speed trajectory. Train(1_1) fixes the problem at t=175 (s) and accelerates, which gives Train(2_1) a further movement authority. At that moment, both trains are in a state of delay, so the agents of both trains take time as the prior optimization objective. Firstly, the required shortest time to arrive at Station B is simulated based on the environment and train model. The target arriving time is the latest time between the environment-allowed earliest arriving time and the scheduled time. Finally, both trains arrive at the target station by following the new speed trajectory. The corresponding travel speed and total energy consumption of the three trains at each moment during this process are shown in Figure 26 (a).

It should be noted that the speed trajectory of Train(3_1) is not affected. However, the Train(2_1) agent has passed its new speed trajectory to Train(3_1) agent when it finds that Train(2_1) is not able to run according to the predefined speed trajectory. When the Train(3_1) agent receives the information from Train(2_1) Agent, it found that the predefined speed trajectory could still meet the MBS constraints. Therefore, Train 3 does not need to take the re-optimization procedure because its speed trajectory has been optimized offline.

Figure 26 (b) shows a process that ensures safety without energy optimization, and some scholars have proposed MAS systems where the train agent would adopt a similar driving strategy in a similar scenario (Hassanabadi et al. 2015). In the control group, the Train Agents exchange signals with neighbouring agents, and the signal includes corresponding train position and speed but no detailed speed trajectory. Therefore when the same situation occurs Train(2_1) agent does not re-optimize when Train(1_1) took braking but continues to follow the original speed trajectory. Train(2_1) brakes to slow down and stops outside the block until the distance and speed between the two trains fails to meet the MBS requirements. Train(2_1) starts to accelerate after the MBS condition is satisfied again. The two trains arrive at the destination station subsequently, during which Train(2_1) wastes a lot of kinetic energy due to the brake. Thus the total energy consumption is higher than Figure 26 (a). A comparison of Figure 26 (a) and (b) shows that Train(2_1) consumes 45MJ (53%) less energy than the control group in the system provided.



(a)

(b)

(c)

(d)

Figure 26: Train Speed Trajectory and Energy Consumption in Case 1

Figure 26 (c) and (d) show another situation where the driver of Train(1_1) discovers the emergency at t=160 seconds, and Train(1_1) needs to stop temporarily for 100 seconds to deal with it. At this point, the Train (2_1) has already completed the acceleration phase, so the Train(2_1) agent provides a driving strategy to slow down early to ensure a safe distance before violating the MBS constraints, but this process tries to avoid wasting kinetic energy by slowing down too much. The interaction process between the Train Agents is similar to the previous scenario and is not repeated here. Comparing Figure 26

(c) and (d), it can be seen that Train (2_1) saves 28.5 MJ of energy. Represents the actual energy consumption as E_{act}, and the optimized energy consumption as E_{opt}, thus the Energy Saving Percentage (ESP) can be calculated as:

$$ESP = \frac{E_{act} - E_{opt}}{E_{act}} \times 100\% \qquad \text{Eq. (52)}$$

Table 7 summarizes the planned, actual and optimized energy consumption of the three trains in the Figure 24 (c) and (d).

Table 8: Planned, Actual and Optimized Energy Consumption of the Three Trains

	Planned Energy Cost E_{plan}	Actual Energy Cost E_{act}	Optimized Energy Cost E_{opt}	Energy Saving Percentages ESP
Train 1_1	38	66	66	0%
Train 2_1	38	66.5	38	42.86%
Train 3_1	38	38	38	0%

In order to verify the robustness of the provided system, the proposed system is tested in numerous similar scenarios. Different scenarios were obtained by adjusting two parameters, travel distance and the temporary parking time of Train(1_1). The parameters included in the tests are shown in Table 9. The travel distance ranged from 3,650 m to 8,500 m in 50-meter intervals, which forms a total of 101 possibilities. The duration of the parking ranged from 45 seconds to 100 seconds in 5-second intervals, which forms a total of 12 possibilities. Thus a total of 101*12=1212 different scenarios were tested, and the results are summarised in Table 10. According to the results in Table 10, the proposed system has a high probability of saving energy when such disturbances occur.

Table 9: Conditions of Different Scenarios

Mass of Each Train (t)	Parking Location	Travel Distance (m)	Faulty Dealing Duration (s)
230	1500m	3650:50:8500	45:5:100

Table 10: Summary of the 1212 Scenarios Applying the MAS

Energy Saving	More than 30%	20% to 30%	15% to 20%	10% to 15%	5% to 10%	Less than 5%
Number of Scenarios	494	57	21	16	16	608
Percentage	40.76%	4.70%	1.73%	1.32%	1.32%	50.17%
Accumulative percentages	40.76%	45.46%	47.19%	48.51%	49.83%	100.00%

To further illustrate the energy-saving patterns, more analysis has been carried out. As shown in Table 11, the total travel distance does not have a significant impact on energy saving. The principle of energy saving for this case relies on the necessary speed trajectory optimization according to the estimated faulty dealing duration of Train (1_1) and the distance between the two trains to minimize the kinetic energy loss. Theoretically, as the distance increases, the primary energy consumption of the train changes from providing kinetic energy to overcoming the resistance. Therefore, the ratio of the saved energy to the total energy consumption will decrease as the travel distance increases. However, 9000m is still a short-distance journey, and most of the energy consumption is generated when the train starts to run to provide kinetic energy for the train. The phase of cruising did not appear for a short time; therefore, it was not observed that the proportion of energy-saving decreased with the increase of distance in this case study.

Table 11: Impact from Travel Distance on Energy Saving

Travel Distance	4000 to 5000	5000 to 6000	6000 to 7000	7000 to 8000	8000 to 9000
ESP	0%-70.35%	0%-76.42%	0%-80.38%	0%-82.57%	0%-82.91%
Average	16.41%	21.57%	27.22%	29.64%	35.04%
Variance	25.87	27.87	29.68	30.79	32.57

As shown in Table 12, when the time spent by Train (1_1) to deal with the faulty is between 45-65 seconds, the average energy savings is zero. This is because when the Train (1_1) has a short stopping time, the Train (2_1) can maintain its original speed trajectory without delay, and the energy consumption in these cases will be the same as the planned travel strategy. By contrast, the highest average *ESP* (82.90%) occurs when the faulty dealing duration is between 65 and 75 seconds. With the increase of the dealing fault duration, the kinetic energy loss of Train (2_1) inevitably increases because it needs to reduce the speed to satisfy the MBS requirements.

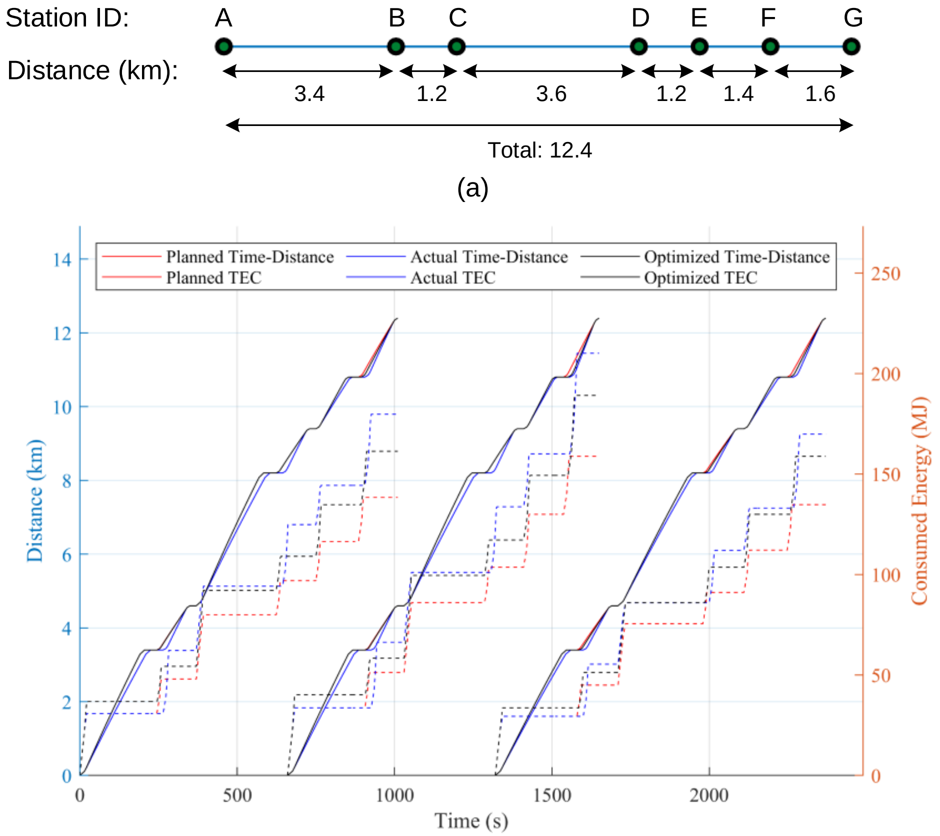
Table 12: Impact from Faulty Dealing Duration on Energy Saving

Faulty Dealing Duration (s)	45 to 55	55 to 65	65 to 75	75 to 85	85 to 95	95 to 100
ESP	0%-0%	0%-0%	0%-82.69%	0%-82.90%	0%-61.76%	0%-46.43%
Average	0%	0%	34.52%	63.91%	37.62%	20.64%
Variance	0	0	37.66	19.19	19.75	16.96

By analysing the thirty-seven scenarios where the energy-saving reaches above 30%, it is found that when the other conditions remain the same, and Trains completed the acceleration phase, a shorter faulty dealing duration (but long enough to allow Train (2_1) to change the original speed trajectory) would allow Train (2_1) to save more energy. Because in shorter durations, Train (2_1) could reserve more kinetic energy. Considering the acceleration phase of trains is relatively short; usually, the Train (2_1) has already finished the acceleration phase when it receives the signal from Train (1_1). Hence with the increasing faulty dealing duration, the average speed of Train (2_1) must be reduced to ensure a safe distance from Train (1_1). A shorter dealing duration allows a higher average speed. Hence the loss of kinetic energy of Train (2_1) would be reduced, which leads to a higher energy saving.

4.1.2 Case Study 2: Disturbance of Delay

Three trains travel from Station A to Station G according to the scheduled schedule.



(a)

(b)

Figure 27: Time-distance and Corresponding Energy Consumption of the Three Trains in Case Study 2

Figure 27 (a) shows the positions of seven stations. The abscissa of Figure 27 (b) indicates the time, the left ordinate indicates the distance, and the right ordinate indicates the consumed energy of the corresponding time. Different colour of lines represents the planned, actual and optimized travel strategy. The planned dwell time for each train in each station is twenty seconds for passengers to board and alight. Due to the large passenger flow at stations B, D and F during peak hours, every train needs to stay for extra twenty seconds when passing through these three stations in the actual group. These trains always speed up in the next section to ensure that they arrive at the next target station on time when they experience a delay, so as to reduce the gap between the actual timetable and the planned one. These trains are always able to arrive at the next station on time due to the existence of buffer time. However, extra energy is consumed within the catching-up section. A trained DNN is employed to reschedule the timetable according to the disturbance within each cluster in the optimized group.

Train 1 in Case Study 2

Case study 2 only focuses on exploring the optimized performance of the proposed system under delay disturbance, thus it is assumed that the travel weight of trains under the planned, actual and optimized group are consistent in each section. The estimated travel weights of Train 1 in the five sections are the same, which is 180t, and the dwell time at each station is 20 seconds in the planned timetable. The running time among Section B-C, D-E, F-G E and F-G in the actual group is 20 seconds shorter than that in the planned group because delay occurs in stations B, D and F. In the optimized group, however, the train agent reduces the total running time in each cluster by 20 seconds and re-schedules the remaining travel time. It can be concluded from Figure 28 and Table 13 that the estimated energy consumption of Train 1 is 138.3 MJ, the actual energy consumption is 179.71MJ, and the optimized energy consumption is 161.3 MJ. The proposed system achieves 10.24% *ESP*.

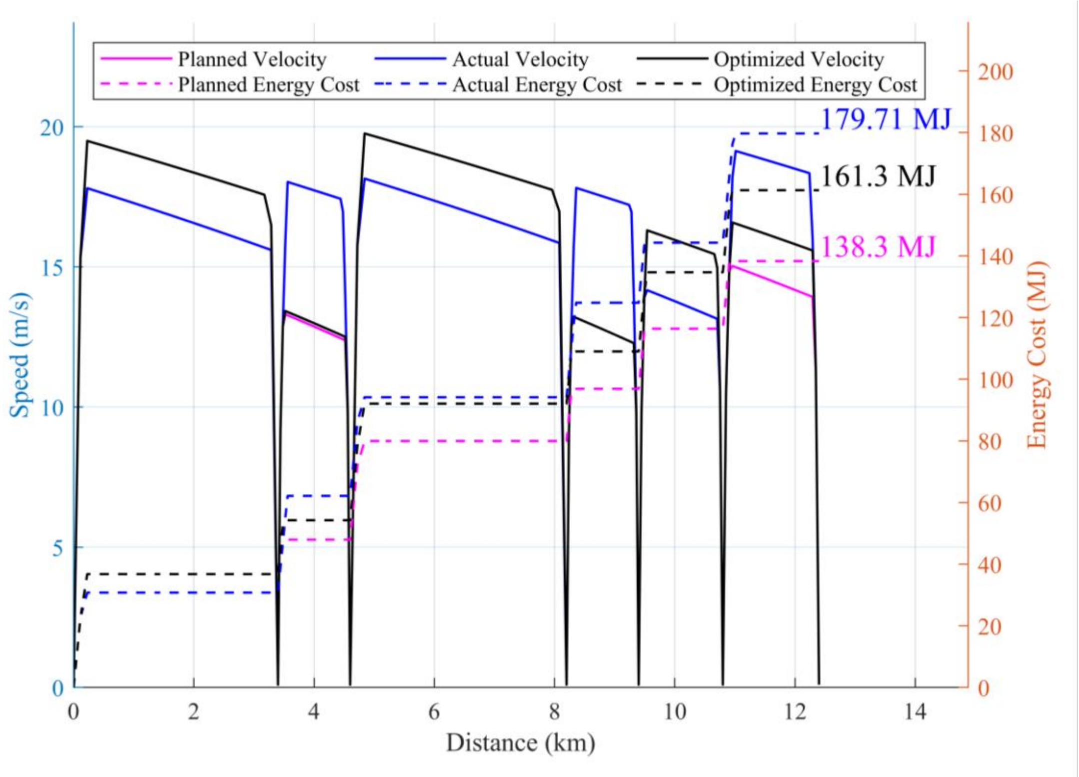


Figure 28: Distance -speed Trajectory and Corresponding Energy Consumption of Train 1

Table 13: Travel Conditions of Train 1 in Case Study 2

	Section A-B	Section B-C	Section C-D	Section D-E	Section E-F	Section F-G	Total
Planned Travel Weight (t)	180	180	180	180	180	180	1080
Planned Travel Time (s)	224.55	105.45	233.89	106.11	115.57	124.43	910
Planned Travel Energy (MJ)	30.78	17.18	31.98	16.95	19.48	21.93	138.3
Actual Travel Weight (t)	180	180	180	180	180	180	1080

Actual Travel Time (s)	224.55	85.45	233.89	86.11	115.57	104.43	850
Actual Travel Energy (MJ)	30.78	31.38	31.98	30.66	19.48	35.42	179.71
Optimized Travel Weight (t)	180	180	180	180	180	180	1080
Optimized Travel Time (s)	205.35	104.65	213.89	106.11	103.94	116.06	850
Optimized Travel Energy (MJ)	36.78	17.48	37.8	16.95	25.69	26.6	161.3

Train 2 in Case Study 2

The speed trajectory, energy consumption, and time distribution of Train 2 are shown in Figure 29 and Table 14. The estimated travel weight of Train 2 varies slightly among the five sections. The travel weight within Section A-B, C-D, and E-F in Planned, Actual and Optimized groups 200t, while the travel weight within other sections is 180t. The planned, actual and optimized energy consumption of Train 2 are 158.76 MJ, 210.04 MJ and 189.06 MJ, respectively. Compared with the actual group, the proposed system achieves 9.99% *ESP*.

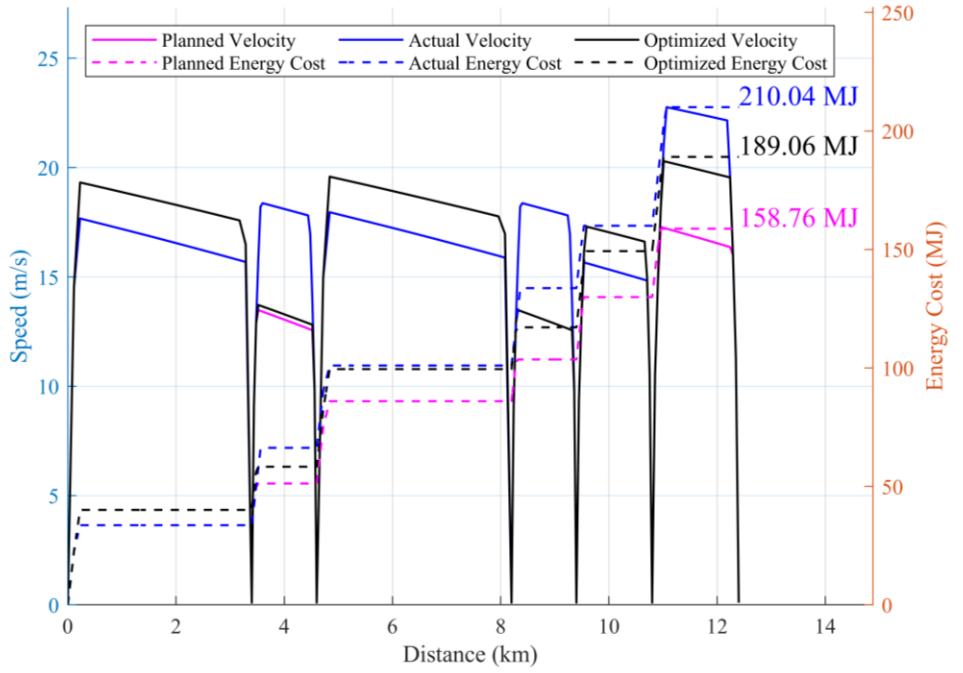


Figure 29: Distance -speed Trajectory and Corresponding Energy Consumption of Train 2

Table 14: Travel Conditions of Train 2 in Case Study 2

	Section A-B	Section B-C	Section C-D	Section D-E	Section E-F	Section F-G	Total
Planned Travel Weight (t)	200	180	200	180	200	180	1140
Planned Travel Time (s)	225.7	104.3	235.7	104.3	107.02	112.98	890
Planned Travel Energy (MJ)	33.62	17.61	34.73	17.61	26.32	28.86	158.76
Actual Travel Weight (t)	200	180	200	180	200	180	1140
Actual Travel Time (s)	225.7	84.3	235.7	84.3	107.02	92.98	830
Actual Travel Energy (MJ)	33.62	32.66	34.73	32.66	26.32	50.05	210.04

Optimized Travel Weight (t)	200	180	200	180	200	180	1140
Optimized Travel Time (s)	207.15	102.85	215.7	104.3	100.98	99.02	830
Optimized Travel Energy (MJ)	40.08	18.22	41.21	17.61	32.16	39.78	189.06

Train 3 in Case Study 2

The total travel time of Train3 in the given seven stations is 950 seconds. The corresponding speed trajectory, energy consumption and time distribution are shown in Figure 30 and Table 15, and the planned travel weight is the same as that of train 2. The planned, actual and optimized energy consumption of Train 3 is 134.72MJ, 169.83MJ and 158.76MJ respectively. Compared with the actual group, the proposed system saves 11MJ of energy and achieves 6.52% ESP.

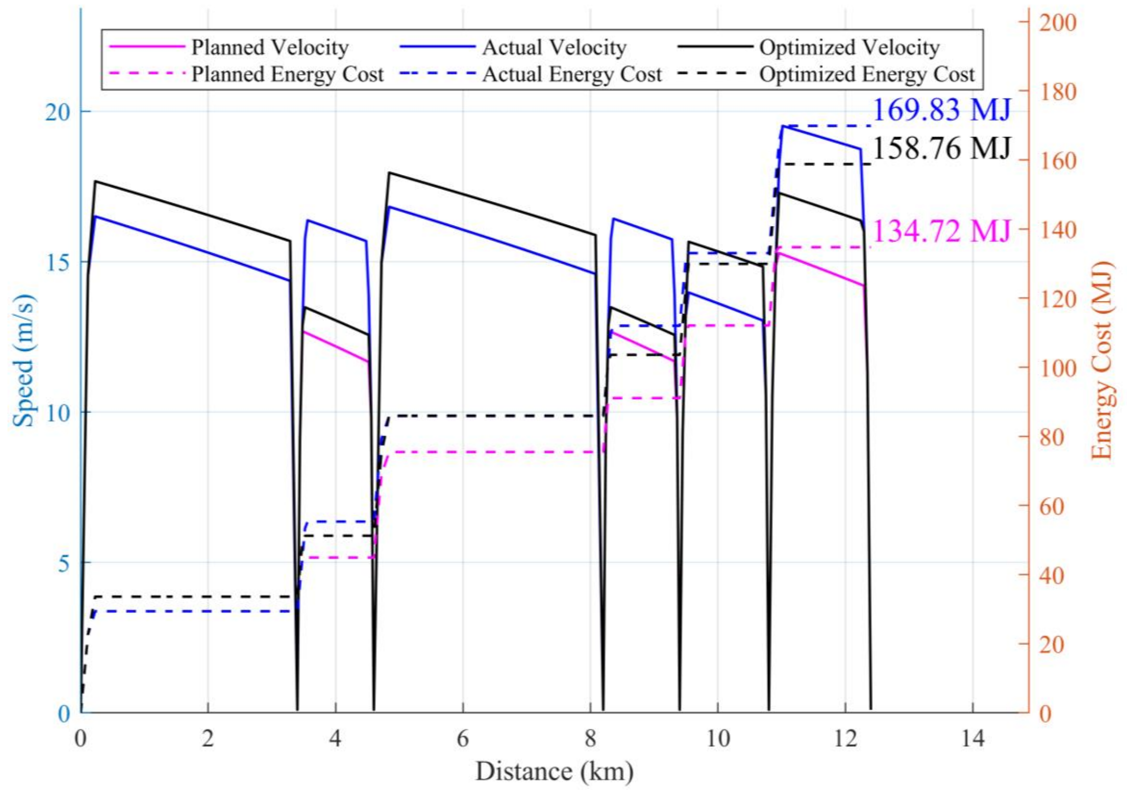


Figure 30: Distance-speed Trajectory and Corresponding Energy Consumption of Train 3

Table 15: Travel Conditions of Train 3 in Case Study 2

	Section A-B	Section B-C	Section C-D	Section D-E	Section E-F	Section F-G	Total
Planned Travel Weight (t)	200	180	200	180	200	180	1140
Planned Travel Time (s)	239.38	110.62	249.54	110.46	117.35	122.65	950
Planned Travel Energy (MJ)	29.39	15.53	30.55	15.58	21.02	22.65	134.72
Actual Travel	200	180	200	180	200	180	1140

Weight (t)							
Actual Travel Time (s)	239.38	90.62	249.54	90.46	117.35	102.65	890
Actual Travel Energy (MJ)	29.39	25.94	30.55	26.11	21.02	36.82	169.83
Optimized Travel Weight (t)	200	180	200	180	200	180	1140
Optimized Travel Time (s)	225.7	104.3	235.7	104.3	107.02	112.98	890
Optimized Travel Energy (MJ)	33.62	17.61	34.73	17.61	26.32	28.86	158.76

Numerical Analysis in Case Study 2

There are 8619 similar scenarios tested in this case. In each scenario, a train travels among three stations and a random delay occurs at the middle one.

At least one of the parameters, such as travel weight, distance, time and delay, is different in different cases. Travel weight equals the multiplication of mass by gravitational acceleration, and mass ranges from 180 tons to 240 tons. The total travel distance (D_{total}) equals the sum of all sections, and the distance of each section ranges from 1200m to 6000m. The total running ($T_{1,3}^{running}$) time ranges from $\frac{D_{total}}{15\ m/s}$ to $\frac{D_{total}}{10\ m/s}$, and the speed limit is 30m/s. The delay ranges from $10\% * T_{1,3}^{running}$ to $20\% * T_{1,3}^{running}$. The *ESP* distribution of these simulation results is shown in Table 16. Results show that all *ESP* of the tested scenarios are positive, ranging from 0.32% to 51.97%, and the average value is 10.13%. The average computational time for each scenario (schedule optimization accelerated trajectory optimization) was 4.13 seconds on a computer with an i7 8700 CPU. The computational results show that the proposed method is able to optimize the timetable and the velocity trajectory in near real-time, which satisfies the practical demands.

Table 16: Summary of the 8619 Scenarios Applying the MAS

$\boldsymbol{ESP}$	More than 30%	20% to 30%	15% to 20%	10% to 15%	5% to 10%	Less than 5%
Number of Scenarios	358	196	818	1510	2657	2672
Percentage	4.36%	2.38%	9.96%	18.39 %	32.36%	32.54%
Accumulative percentages	4.36%	6.75%	16.71%	35.10%	67.46%	100.00%

In order to discover the parameters closely related to energy saving, the relationship between different delay parameters and *ESP* is explored in this study. Table 17 shows the Pearson coefficient between these parameters and the energy savings.

Table 17: Pearson coefficients between different parameters in category 1

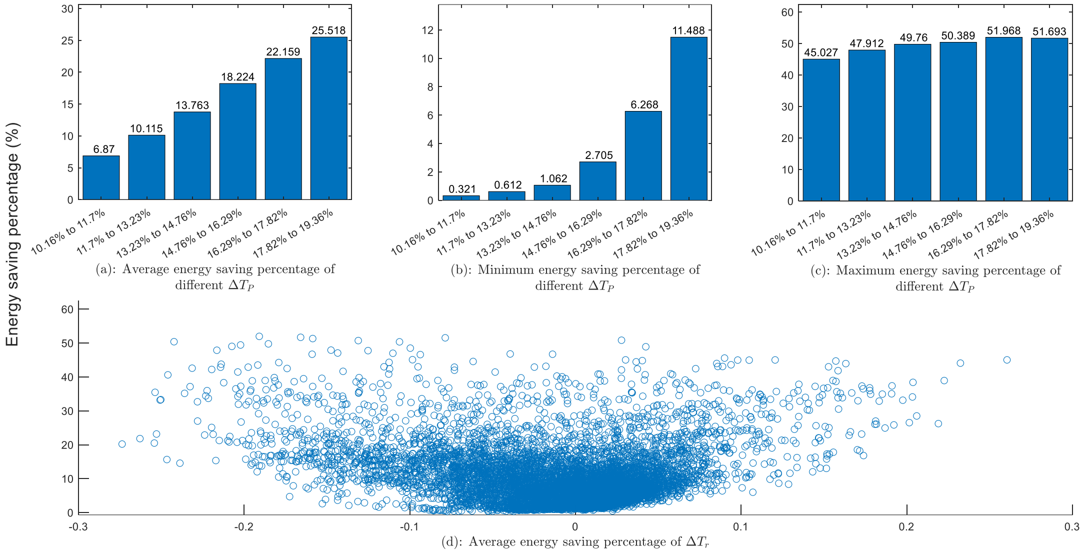
	$\Delta T\ (s)$	$\frac{D_{total}}{\Delta T}\,(m/s)$	$\lvert\Delta T_p\rvert$	$\lvert\Delta T_r\rvert$
ESP	-0.051	-0.427	**0.542**	**0.538**

A stronger correlation between two variables results in a closer absolute value of Pearson coefficient to 1. As shown in Table 17, $\left|\Delta T_p\right|$ and ΔT_r have a strong correlation with the ESP. Both $\left|\Delta T_p\right|$ and ΔT_r are dimensionless variables, which are calculated by Eq. (53) and Eq. (54), where ΔT_p represents the percentage of running time changes against the planned running time, and ΔT_r represents the distribution difference between the actual running time and the planned running time. The planned running time for the two sections in each scenario is set as $T_{plan1}^{running}$ and $T_{plan2}^{running}$, while the actual running time are set as $T_{act1}^{running}$

and $T_{act2}^{running}$.

$$\Delta T_p = \frac{\left|\left(T_{plan1}^{running} + T_{plan2}^{running}\right) - \left(T_{act1}^{running} + T_{act2}^{running}\right)\right|}{T_{plan1}^{running} + T_{plan2}^{running}} \qquad \text{Eq. (53)}$$

$$\Delta T_r = \frac{T_{plan1}^{running}}{T_{plan2}^{running}} - \frac{T_{act1}^{running}}{T_{act2}^{running}} \qquad \text{Eq. (54)}$$



(a): Average energy saving percentage of different ΔT_P

(b): Minimum energy saving percentage of different ΔT_P

(c): Maximum energy saving percentage of different ΔT_P

(d): Average energy saving percentage of ΔT_r

Figure 31: Energy-saving Percentages (ESP) of Different Delay

Figure 31 shows the relationship of ESP to related variables. In Figure 31, the ordinates of all subgraphs are ESP and the abscissa are ΔT_p. Figure 31 (a) to (c) shows the average, minimum, and maximum values of ESP for different ΔT_p. It can be discovered that the ratio of delay time to total time is positively correlated with energy savings. In Figure 31 (d), the abscissa is delta T_r, which is the difference between the ratio of the two stations' actual run-time to the planned run-time. Thus the ESP is less than 20% when ΔT_r is close to zero.

Optimization Duration:

The box diagram in Figure 32 shows the optimization duration of the tested cases. The upper and lower limit is the maximum and minimum optimization time of each group. The upper and lower boundaries of the box are the upper and lower quartile, respectively. The longest optimization time is 6.81 seconds, and the average optimization speed is 0.83 seconds in 8619 cases. Table 18 shows the optimization duration of the tested scenarios. The proposed system finishes the optimization process in 2.42 seconds for 95% of scenarios, proving the system satisfies the near real-time optimization requirements under delay disturbance.

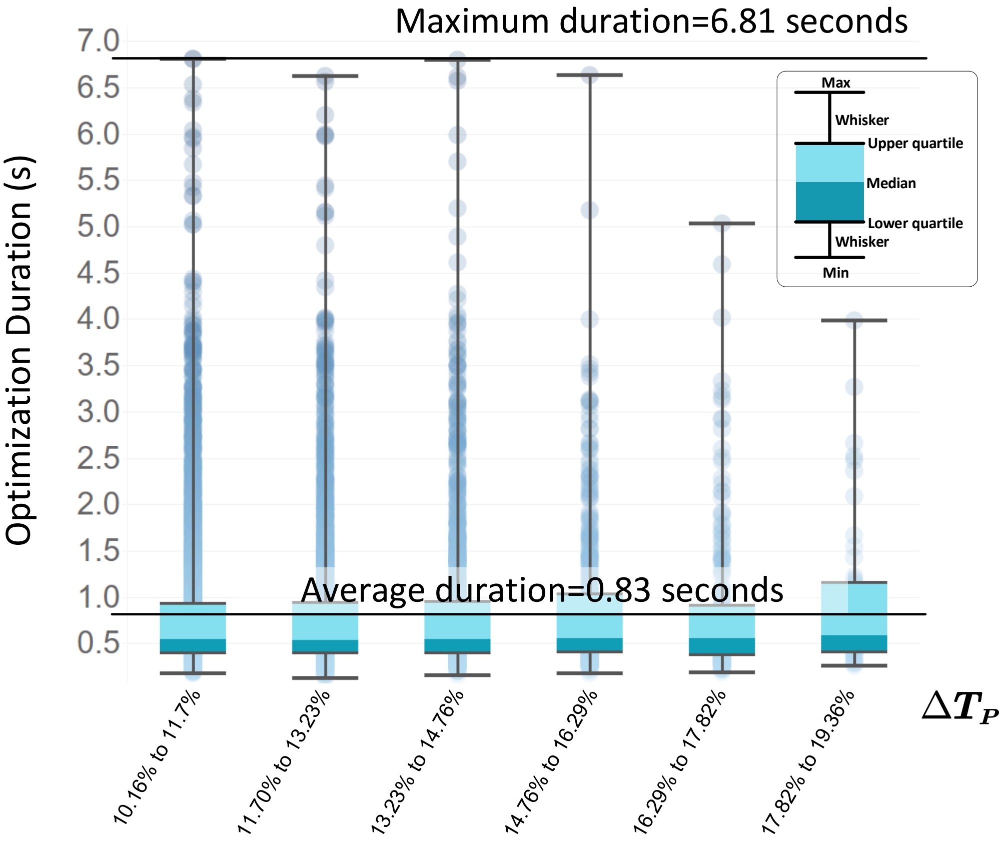


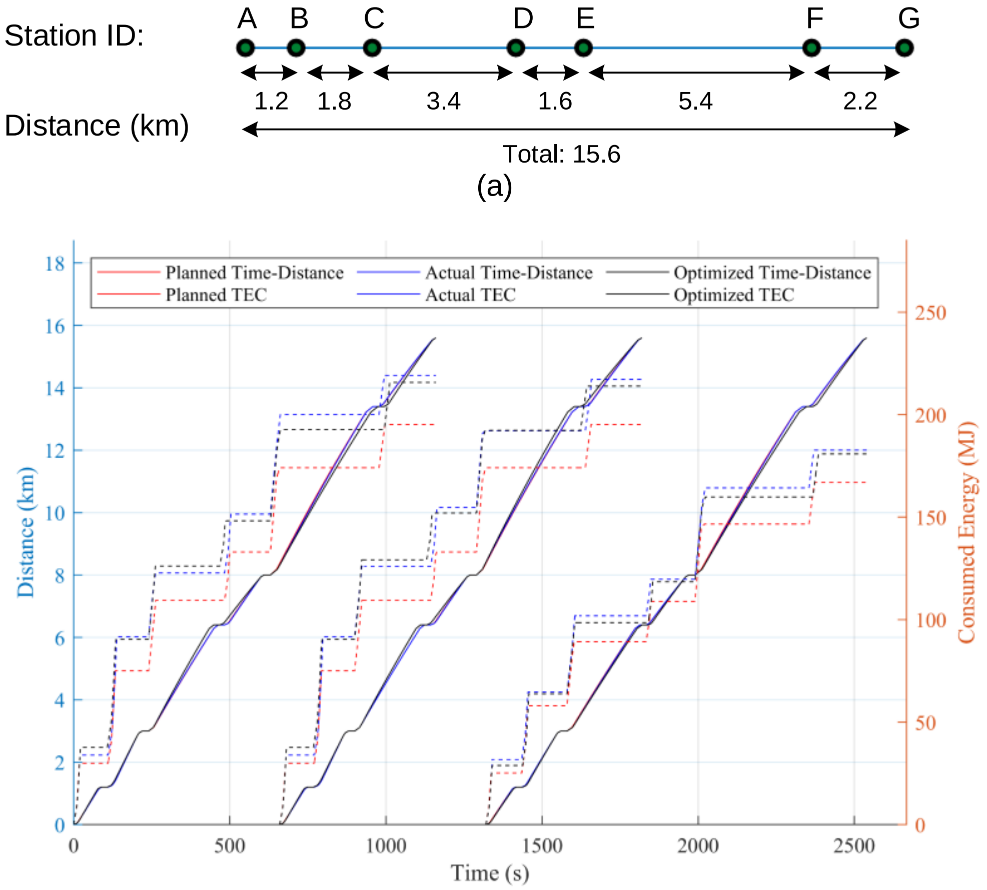
Figure 32: Optimization Duration of tested Scenarios Under Delay Disturbance

Table 18: Optimization Duration of Tested Scenarios under Delay Disturbance

Optimization Duration (s)	≤ 0.39	≤ 0.54	≤ 0.94	≤ 2.42	≤ 6.81
Satisfied Scenarios	2155	4310	6465	8189	8619
Satisfied Percentage	25%	50%	75%	95%	100%

4.1.3 Case Study 3: Disturbance of Weight Change

This Case Study tests the performance of the proposed system when the rail transit system encounters the disturbances of weight change.



(a)

(b)

Figure 33: Time-distance and Corresponding Energy Consumption of the Three Trains in Case Study 3

There are three trains in the tested example, and the positions of seven stations are shown in Figure 33 (a). The planned dwell time of the trains at each station is 20 seconds, and no delay disturbance occurs during the process. However, the actual weights of each train running within sections are assumed to be different from the plan. In the optimization group, the trained DNN reschedules the timetable in each cluster according to the predicted actual travel weight, and the MILP algorithm is employed to optimize the trajectory accordingly. Each train needs to arrive at the last station in each cluster on time to avoid a large gap between the adopted timetable and the original one. Similar to the previous case study, Figure 33 (b) shows the planned time-distance diagram of each train and the corresponding energy consumption.

Train 1 in Case Study 3

The distance-speed trajectory of Train 1 is shown in Figure 32, and the planned, actual and optimized travel time within each section is shown in Table 19. Case Study 3 focuses on exploring the optimized performance of the proposed system with only weight change disturbance; thus, the planned travel time and actual travel time of Train 1 in each section are consistent. The train agent re-schedules the timetable within each cluster by considering the weight change in the optimized group providing consistent total travel time among each cluster. As shown in Figure 32 and Table 19, the planned energy consumption of Train 1 from Station A to Station G is 195.16MJ, the actual energy consumption is 219.18MJ, and the energy consumption of the optimized group is 215.69MJ. Train 1 achieved a 1.59% ESP reduction, which is 3.49MJ .

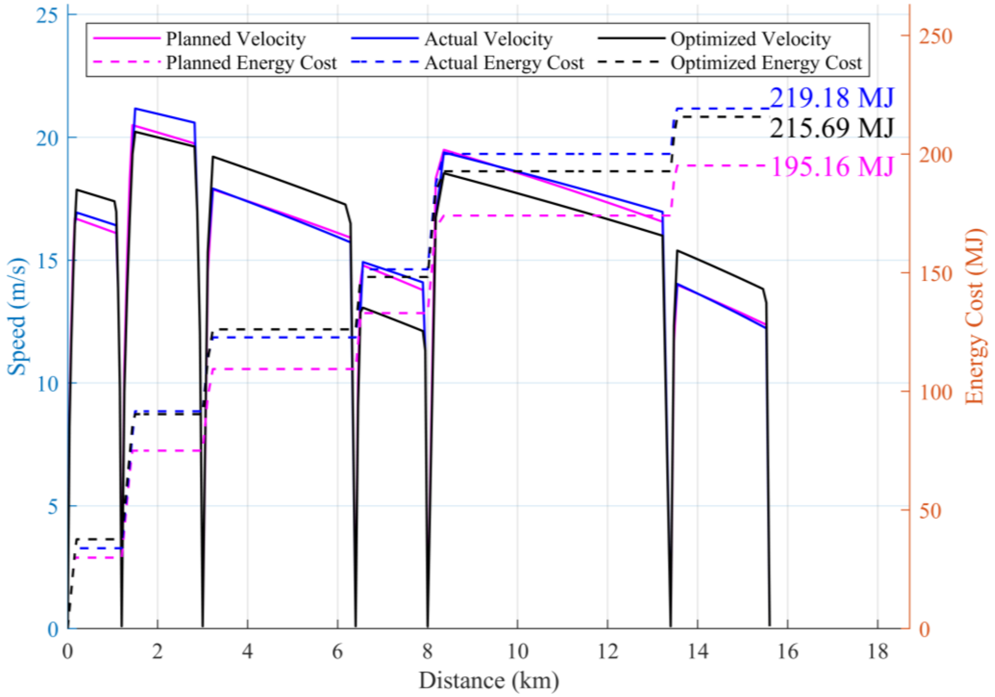


Figure 34: Distance-speed Trajectory and Corresponding Energy Consumption of Train 1

Table 19: Travel Conditions of Train 1 in Case Study 3

	Station A-B	Station B-C	Station C-D	Station D-E	Station E-F	Station F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	1200
Planned Travel Time (s)	90.43	109.57	223.34	126.66	328.32	181.68	1060
Planned Travel Energy (MJ)	29.95	45.08	34.4	23.55	41.1	21.08	**195.16**
Actual Travel Weight (t)	220	240	180	240	240	180	1300
Actual Travel Time (s)	90.43	109.57	223.34	126.66	328.32	181.68	1060
Actual Travel	33.91	57.7	31.14	28.69	48.62	19.12	**219.18**

Energy (MJ)							
Optimized Travel Weight (t)	220	240	180	240	240	180	1300
Optimized Travel Time (s)	87.65	112.35	208.54	141.46	344.7	165.3	1060
Optimized Travel Energy (MJ)	37.68	52.72	35.74	22.08	44.54	22.94	**215.69**

Train2 in Case Study 3

The total planned travel time of Train 2 between the seven stations is 1060 seconds, and the corresponding speed trajectory, energy consumption and time are shown in Figure 35 and Table 20. The planned energy consumption of Train 1 and Train 2 are the same because their travel weights and planned running time in each cluster are equal, which is 195.16MJ. The actual energy consumption and optimized energy consumption of Train 2 are 217.18MJ and 213.90MJ, respectively, which gives a 1.51% *ESP* reduction.

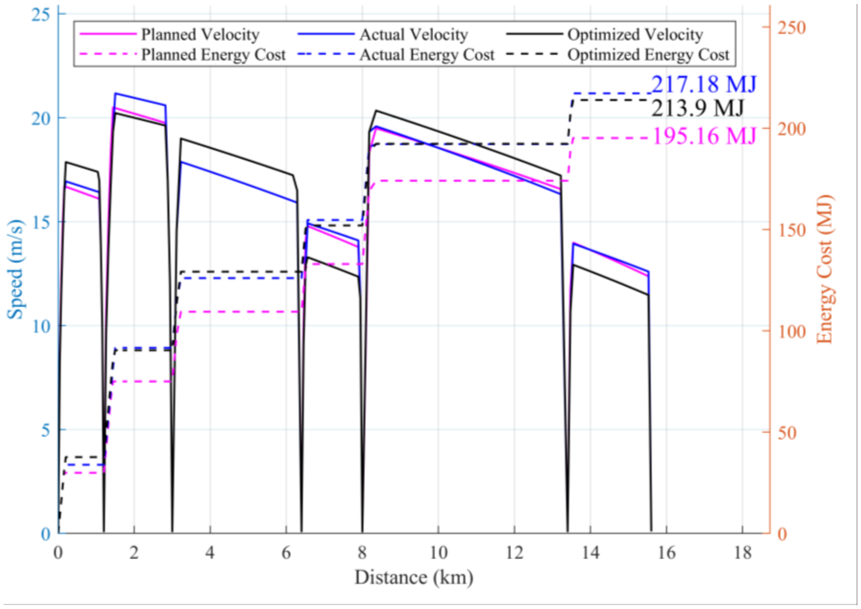


Figure 35: Distance-speed Trajectory and Corresponding Energy Consumption of Train 2

Table 20: Travel Conditions of Train 2 in Case Study 3

	Section A-B	Section B-C	Section C-D	Section D-E	Section E-F	Section F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	1200
Planned Travel Time (s)	90.43	109.57	223.34	126.66	328.32	181.68	1060
Planned Travel Energy (MJ)	29.95	45.08	34.4	23.55	41.1	21.08	**195.16**
Actual Travel Weight (t)	220	240	200	240	180	240	1320
Actual Travel Time (s)	90.43	109.57	223.34	126.66	328.32	181.68	1060
Actual Travel Energy (MJ)	33.91	57.7	34.4	28.69	37.43	25.05	**217.18**
Optimized Travel Weight (t)	220	240	200	240	180	240	1320
Optimized Travel Time (s)	87.65	112.35	210.71	139.29	313.13	196.87	1060
Optimized Travel Energy (MJ)	37.68	52.72	38.78	22.82	40.31	21.59	**213.9**

Train 3 in Case Study 3

The corresponding speed trajectory, energy consumption and travel time are shown in Figure 36 and Table 21. The planned travel duration of Train 3 between seven stations is 1120 seconds, which is slightly longer than that of Train1 and Train 2. Therefore, the planned energy consumption of Train 3 is slightly lower than that of the two leading trains, which is 166.91 MJ. The optimized energy consumption of Train 3 is 180.85MJ, which is

1.82 MJ lower than the actual energy consumption and gives 1% ESP.

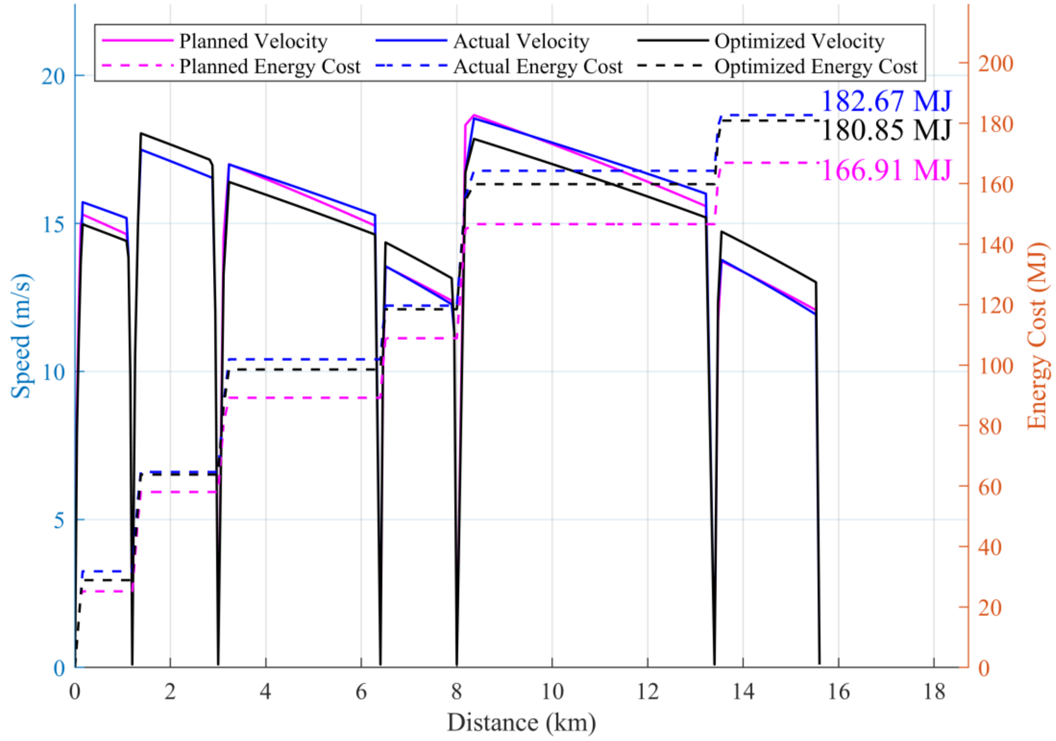


Figure 36: Distance-speed Trajectory and Corresponding Energy Consumption of Train 3

Table 21: Travel Conditions of Train 3 in Case Study 3

	Section A-B	Section B-C	Section C-D	Section D-E	Section E-F	Section F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	-
Planned Running Time (s)	94.83	125.17	233.22	136.78	344.54	185.46	1120

Planned Energy Consumption (MJ)	25.18	32.85	31.15	19.68	37.74	20.3	**166.91**
Actual Travel Weight (t)	240	200	240	180	240	180	-
Actual Running Time (s)	94.83	125.17	233.22	136.78	344.54	185.46	1120
Actual Energy Consumption (MJ)	31.79	32.85	37.25	17.77	44.58	18.42	**182.67**
Optimized Travel Weight (t)	240	200	240	180	240	180	-
Optimized Running Time (s)	97.73	122.27	240	130	357.41	172.59	1120
Optimized Energy Consumption (MJ)	28.88	34.93	34.74	19.91	41.38	21.02	**180.85**

Numerical Analysis of Case Study 3

A total of 26,537 scenarios are tested in this category. It is assumed that no delay occurs during operation in each scenario. But the weight of the actual trip is different from the original plan. It is assumed that the planned train mass is fixed at 200 tons, while the actual train mass is 180 tons to 240 tons. Other parameter settings and ranges are similar to Case Study 2. The test results show that the average, maximum and minimum ESP are 0.21%, 5.50%, and -0.009%, respectively. The value of ESP is not significant compared with the delay disturbance (the second category). Table 22 summarizes the value of ESP with different weight changes.

Table 22: Summary of the 26537 Scenarios Applying the MAS

Total Mass Difference (t)	-80.00	-60.00	-40.00	-20.00	0.00	20.00	40.00
ESP_{max} (%)	1.24	2.05	3.56	5.50	3.24	1.61	1.96
ESP_{min} (%)	-0.01	0.00	-0.01	0.00	0.10	0.00	-0.01
ESP_{ave} (%)	0.03	0.08	0.20	0.38	0.32	0.09	0.01
Number of Scenarios	1784	3557	5328	7058	3503	3564	1747
Percentage	6.72%	13.40%	20.07%	26.59%	13.20%	13.43%	6.58%
Accumulative percentages	6.72%	20.12%	40.20%	66.79%	79.99%	93.42%	100.00%

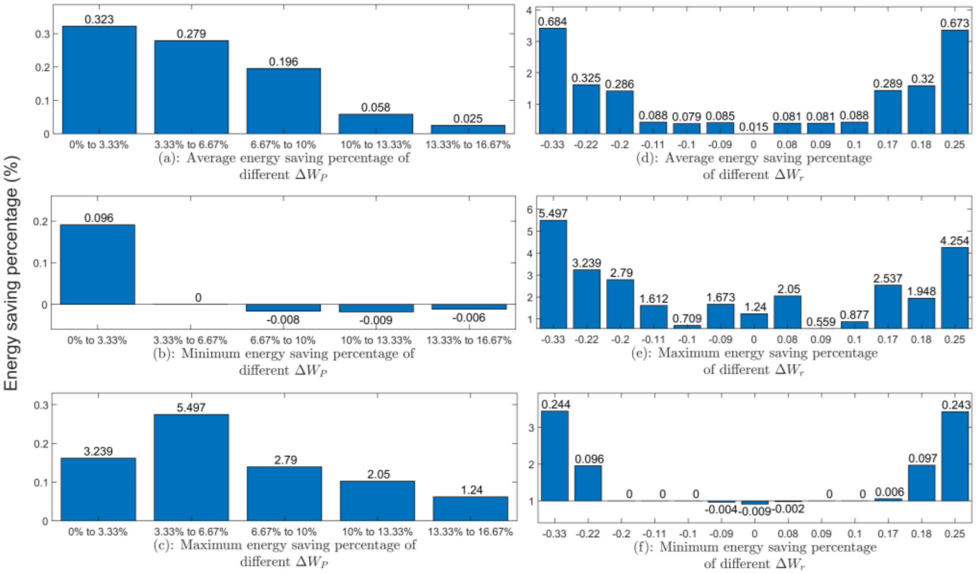
Similar to the previous category, the parameters that have a major impact on energy-saving efficiency when the weight change disturbance occurs is further analysed. The Pearson coefficient of relevant parameters is shown in Table 23. The ΔW_p and ΔW_r can be obtained by Eq. (55) and Eq. (56), where $W_{plan1}^{running}$, $W_{act1}^{running}$, $W_{plan2}^{running}$, and $W_{act2}^{running}$ respectively represent the assumed travel weight and actual running weight of the first and second sections.

$$\Delta W_p = \frac{\left|\left(W_{plan1} + W_{plan2}\right) - \left(W_{act1} + W_{act2}\right)\right|}{W_{plan1} + W_{plan2}} \qquad \text{Eq. (55)}$$

$$\Delta W_r = \frac{W_{plan1}}{W_{plan2}} - \frac{W_{act1}}{W_{act2}} \qquad \text{Eq. (56)}$$

Table 23: Pearson coefficients between different parameters of category 2

	$\Delta W\ (N)$	$\frac{D_{total}}{\Delta \mathrm{W}}\ (m/N)$	$\lvert\Delta W_p\rvert$	$\lvert\Delta W_r\rvert$
ESP	0.039	-0.15	**-0.377**	**0.759**



(a): Average energy saving percentage of different ΔW_P

(b): Minimum energy saving percentage of different ΔW_P

(c): Maximum energy saving percentage of different ΔW_P

(d): Average energy saving percentage of different ΔW_r

(e): Maximum energy saving percentage of different ΔW_r

(f): Minimum energy saving percentage of different ΔW_r

Figure 37: Relation between ESP, ΔW_p and ΔW_r

The ordinates of all subgraphs in Figure 37 are ESP, while the abscissas of Figure 37 (a) to (c) are ΔW_p. The ESP values shown in Figure 37 (a) to (c) are average, maximum and minimum values for the corresponding ΔW_p, respectively. Three subfigures show that the ESP value does not increase monotonously with the increase of ΔW_p. The abscissas of Figure 37 (d) to (f) are ΔW_r. The three subfigures show that the average, maximum, and minimum value of ESP are positively correlated with the absolute value of ΔW_r. The maximum ESP is 5.497%, and the minimum is -0.009% with the weight change disturbance. The value of ESP is very small in most scenarios compared with the scenarios with delay disturbances, which indicates that travel weight change has little influence on the distribution of timetables. This phenomenon proves that the extra energy consumption caused by weight change disturbance is less than that caused by delay. Hence, the prediction accuracy of travel weight by passenger flow will not significantly impact the energy-saving of the proposed system.

Optimization Duration in Case Study 3

The optimization duration for case study 3 with different ΔW_p is shown in Figure 38. The maximum optimization duration is 7.2 seconds, while the average optimization duration is 0.86 for this case. Furthermore, the optimization duration seems not influenced by the value of ΔW_p. The diagram shows that the proposed system is able to provide the optimized control strategy in a near real-time when the system faces disturbances of travel weight change.

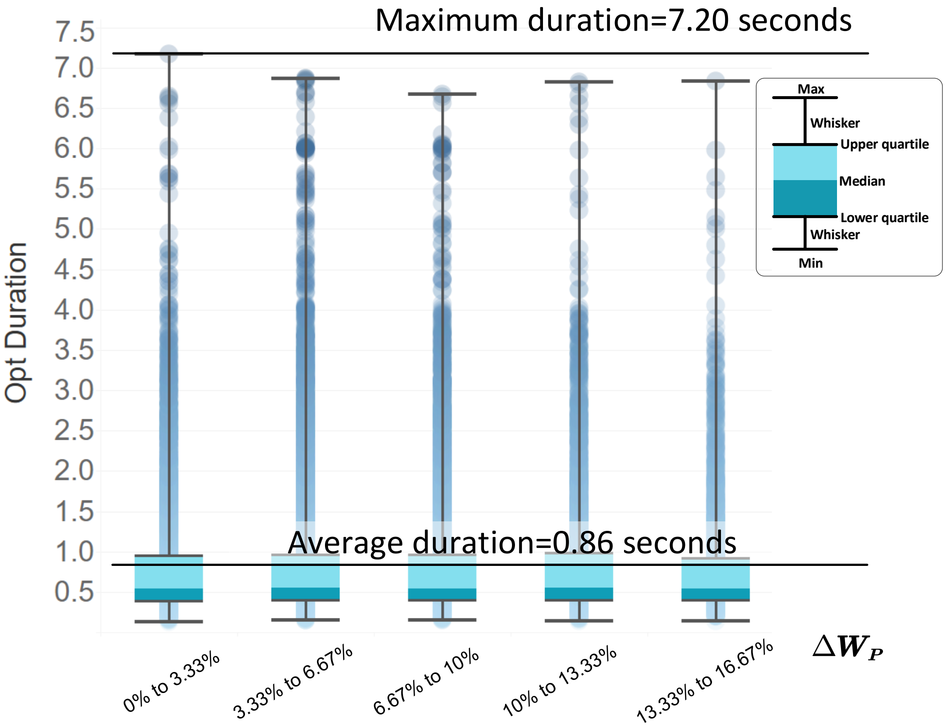


Figure 38: Optimization Duration of Tested Scenarios Under Weight Change Disturbances

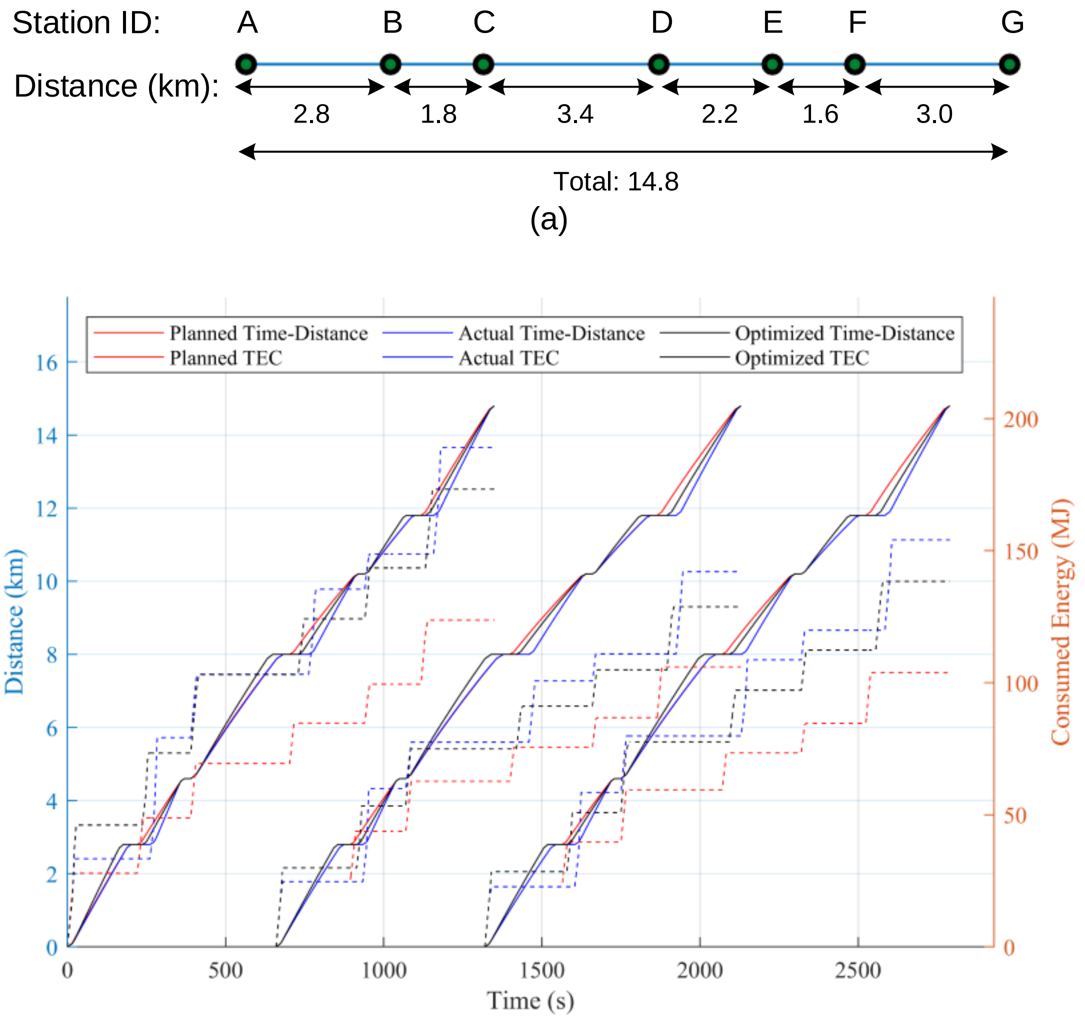
Table 24: Optimization Duration of Tested Scenarios under Mass Change Disturbance

Optimization Duration (s)	≤ 0.42	≤ 0.58	≤ 0.98	≤ 2.50	≤ 7.20

Satisfied Scenarios	6635	13269	19904	25212	26537
Satisfied Percentage	25%	50%	75%	95%	100%

4.1.4 Case Study 4: Delay and Weight Change

Three trains travel among seven stations, and both delay and mass change occur during the process. The locations of the seven stations are shown in Figure 39 (a). Random delays and travel weight changes occur during the test process, and the time and travel weight of the planned, actual, and optimized groups are demonstrated in the following descriptions.



(a)

(b)

Figure 39: Time-speed Trajectory and Corresponding Energy Consumption of the Three Trains

Train 1 in Case Study 4

The total running time of train 1 is 1250 seconds among the three clusters. The speed trajectory, energy consumption and time distribution are shown in Figure 40 and Table 25. The planned travel weight within each section is 200 t, while the travel weight of different sections in the actual and optimized group are different. Random delays occur at Stations B, D and F. The planned, actual, and optimized energy consumption of Train 1 are 123.73 MJ, 189.14 MJ and 173.29 MJ, respectively. The optimization provides 8.38% *ESP*.

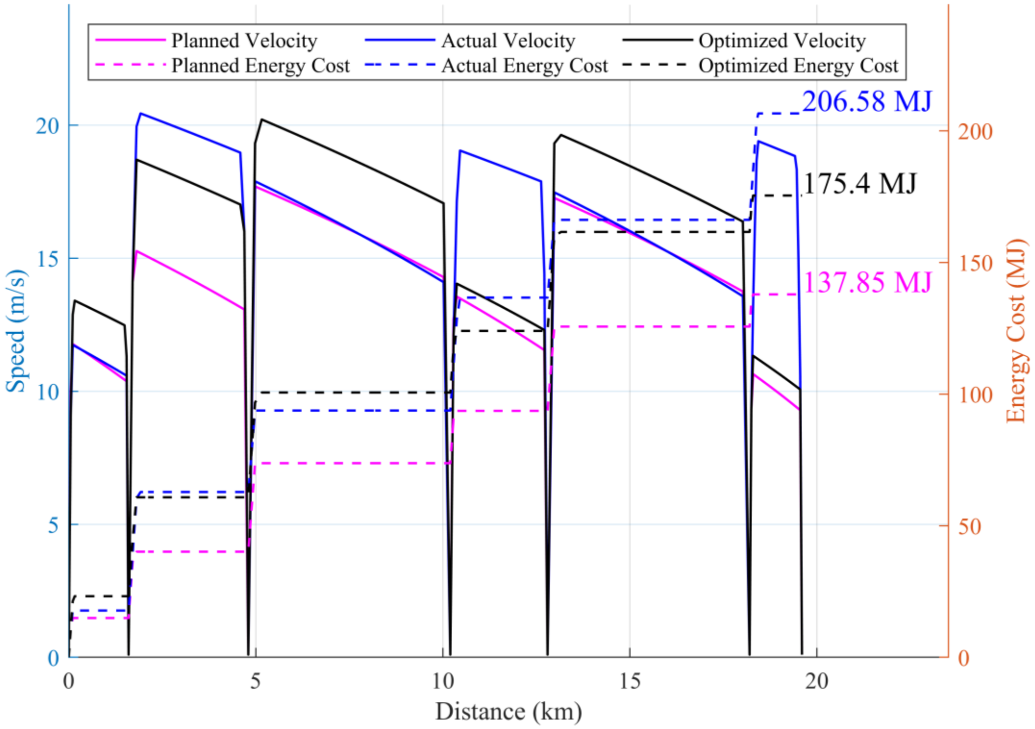


Figure 40: Distance-speed Trajectory and Corresponding Energy Consumption of Train1 in Case Study 4

Table 25: Travel Conditions of Train 1 in Case Study 4

	Station A-B	Station B-C	Station C-D	Station D-E	Station E-F	Station F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	1200

Planned Running Time (s)	201.04	148.96	293.01	216.99	157.92	232.08	1250
Planned Energy Consumption (MJ)	27.87	20.97	20.59	15.29	14.75	24.26	123.73
Actual Travel Weight (t)	240	200	240	220	180	220	1300
Actual Running Time (s)	201.04	108.96	293.01	156.99	157.92	192.08	1110
Actual Energy Consumption (MJ)	33.36	45.88	23.96	32.21	13.4	40.33	189.14
Optimized Travel Weight (t)	240	200	240	220	180	220	1300
Optimized Running Time (s)	176.87	133.13	259.46	190.54	131.79	218.21	1110
Optimized Energy Consumption (MJ)	46.12	27.26	29.73	21.11	19.28	29.8	173.29

Train 2 in Case Study 4

The total given running time of Train 2 is 1370 seconds, as shown in Table 26, and the speed trajectory is shown in Figure 41. Similar to Train1, random delay and travel weight change disturbance occurs during the process. The energy consumption of Train 2 is 106.03 MJ in the planned group, 142.05 MJ in the actual group and 128.78 MJ in the optimized group. The proposed system provides 13.27 MJ energy savings, and the *ESP* for train 2 is 9.34%.

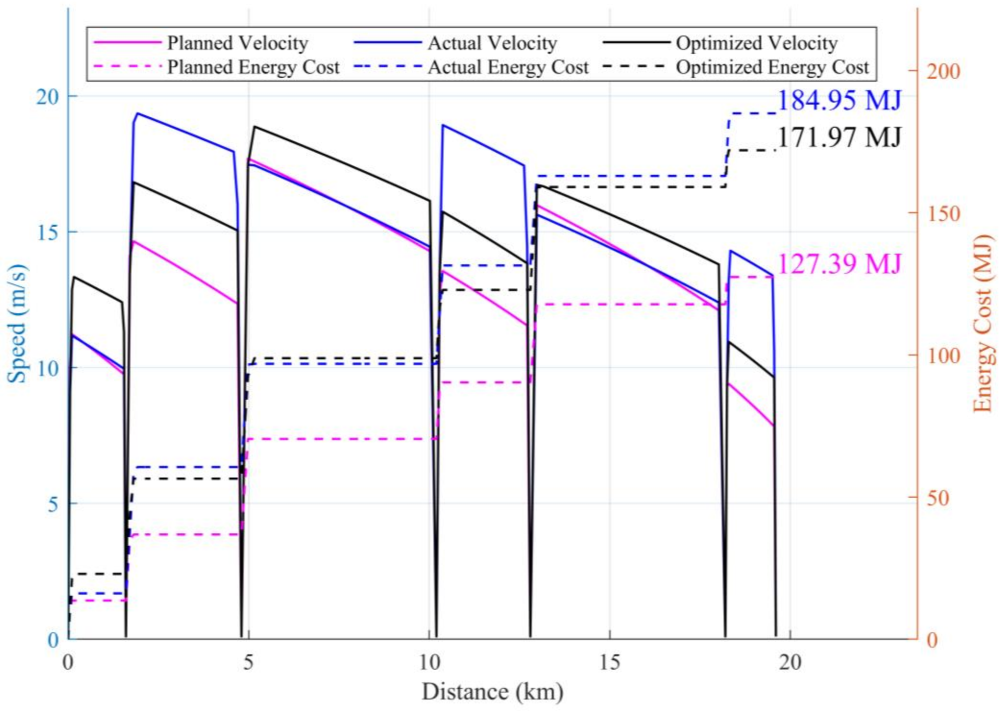


Figure 41: Distance-speed Trajectory and Corresponding Energy Consumption of Train2 in Case Study 4

Table 26: Travel Conditions of Train 2 in Case Study 4

	Station A-B	Station B-C	Station C-D	Station D-E	Station E-F	Station F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	1200
Planned Running Time (s)	214.01	155.99	308.94	241.06	184.38	265.62	1370
Planned Energy Consumption (MJ)	24.69	19.05	18.96	12.91	11.2	19.22	106.03
Actual Travel	200	180	180	220	180	200	1160

Weight (t)							
Actual Running Time (s)	214.01	115.99	308.94	181.06	184.38	205.62	1210
Actual Energy Consumption (MJ)	24.69	35.3	17.51	23.2	10.27	31.09	142.05
Optimized Travel Weight (t)	200	180	180	220	180	200	1160
Optimized Running Time (s)	194.95	135.05	269.99	220.01	155.72	234.28	1210
Optimized Energy Consumption (MJ)	29.9	23.46	21.58	16.22	13.75	23.86	128.78

Train 3 in Case Study 4

The speed trajectory, energy consumption and time distribution of Train 3 are shown in Figure 42 and Table 27. Train 3 has 1370 seconds of running time for the planned group. Both delay and weight change disturbances occur during the process of running. The planned, actual, and optimized energy consumption for Train 3 is 103.85 MJ, 154.08 MJ and 138.37 MJ. Thus the *ESP* for Train 3 is 10.19%, and the energy savings is 15.71MJ.

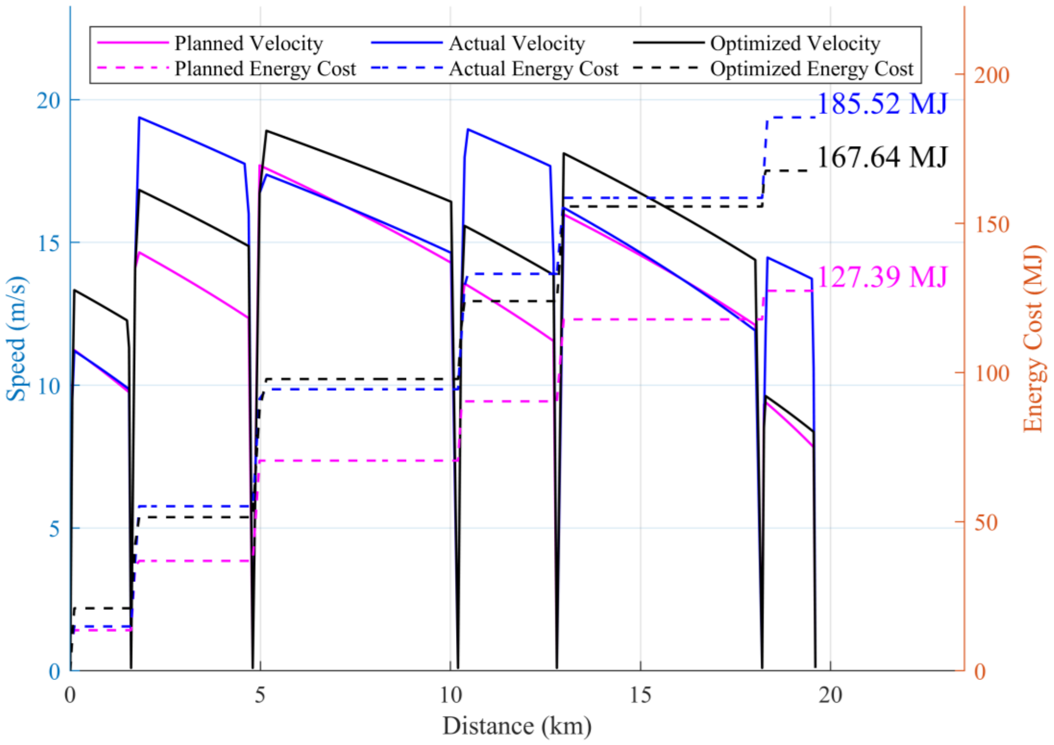


Figure 42: Distance-speed Trajectory and Corresponding Energy Consumption of Train3 in Case Study 4

Table 27: Travel Conditions of Train 3 in Case Study 4

	Station A-B	Station B-C	Station C-D	Station D-E	Station E-F	Station F-G	Total
Planned Travel Weight (t)	200	200	200	200	200	200	1200
Planned Running Time (s)	223.79	166.21	300.28	229.72	184.38	265.62	1370
Planned Energy Consumption (MJ)	22.77	16.92	19.8	13.94	11.2	19.22	103.85
Actual Travel Weight (t)	200	220	220	240	200	220	1300

Actual Running Time (s)	223.79	126.21	300.28	169.72	184.38	205.62	1210
Actual Energy Consumption (MJ)	22.77	35.71	21.35	28.82	11.2	34.22	154.08
Optimized Travel Weight (t)	200	220	220	240	200	220	1300
Optimized Running Time (s)	199.12	150.88	262.88	207.12	155.46	234.54	1210
Optimized Energy Consumption (MJ)	28.46	22.38	26.78	19.56	15.18	26.01	138.37

Numerical Analysis of Case Study 4

A total of 11755 scenarios were tested to verify the speed and efficiency of the proposed system under delay and weight change disturbance. A train runs among three stations in each scenario. Delays and weight change disturbances occur during the process. The range of each parameter is the same as the previous two case studies, while the travel weight and time in the planning, actual and optimized group are different. The delay in the tested situation is greater than 10% and less than 20% of the planned total travel time.

The disturbed trains in the actual group always try to arrive at the next station on time after a disturbance occurs so as to reduce the differences between the actual and planned timetable. Furthermore, the travel weight of each section is also not considered in the actual group. On the contrary, each train optimizes the timetable among the clusters in the optimized group when a disturbance occurs. The ESP of each group and the corresponding $|\Delta \mathrm{T_r}|$, and$|\Delta \mathrm{W_r}|$ are shown in Figure 43. The three axes of Figure 43 represent $|\Delta T_r|$, $|\Delta W_r|$, and ESP. According to the figure, the energy savings significantly increases with the increase of $|\Delta T_r|$, while ΔW_r does not show the same obvious influence. Note that the highest absolute value of the tested ΔT_r is higher than ΔW_r, for the weight of the train is very heavy, and the change in passenger flow will not cause great changes in ΔW_r. The average, maximum, and minimum value of *ESP* for the tested scenarios in this case are 10.02%, 56.72%, and 0.00025%, respectively.

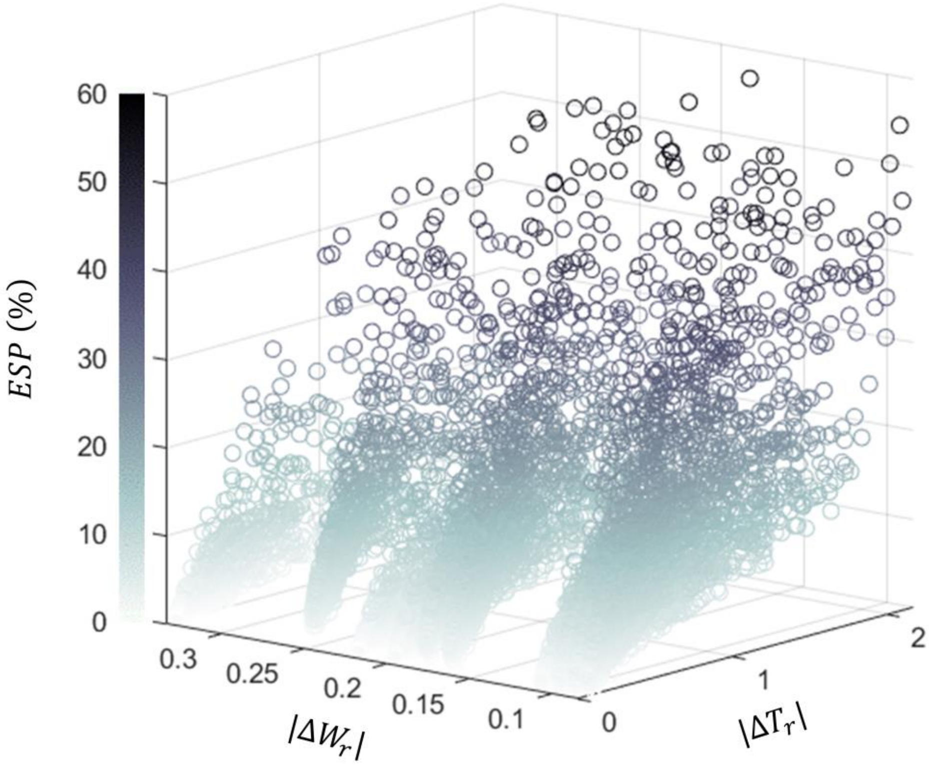


Figure 43: *ESP* with different $|\Delta W_r|$ and $|\Delta T_r|$

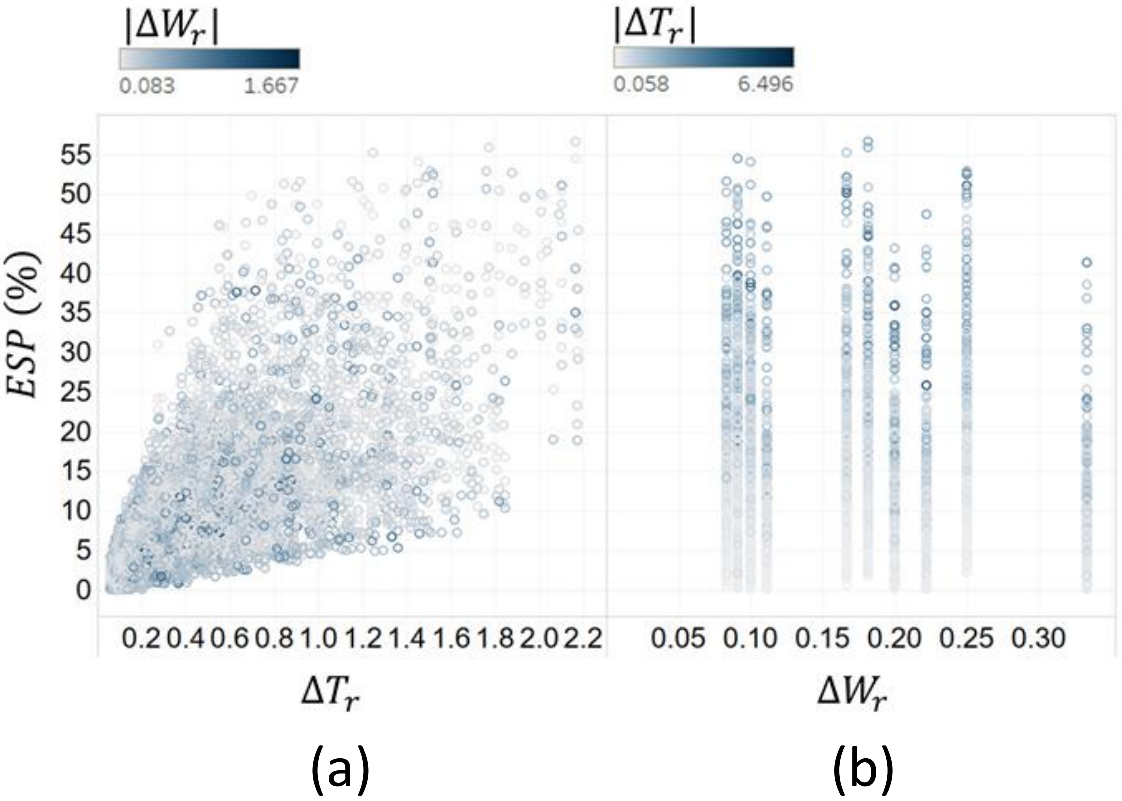


Figure 44: Energy-saving percentage with different $|\Delta W_r|$ and $|\Delta T_r|$

It is expected that in the practical application, the proposed system realizes higher energy-saving performance when dealing with delay disturbance than the weight change disturbance according to the numerical test results of this case study, which is consistent with the conclusion drawn by comparing case study 2 and case study 3.

Optimization Duration of Case Study 4

The box diagram in Figure 45 shows the time cost for optimizing the test scenarios in Case Study 4. The maximum optimization duration is 8.84 seconds, as shown in the diagram, while the average optimization is 0.77 seconds. Compare with case study 2 and case study 3. It is concluded that the disturbance type would not influence the optimization duration of the proposed system. Thus the system satisfies the near real-time optimization requirements.

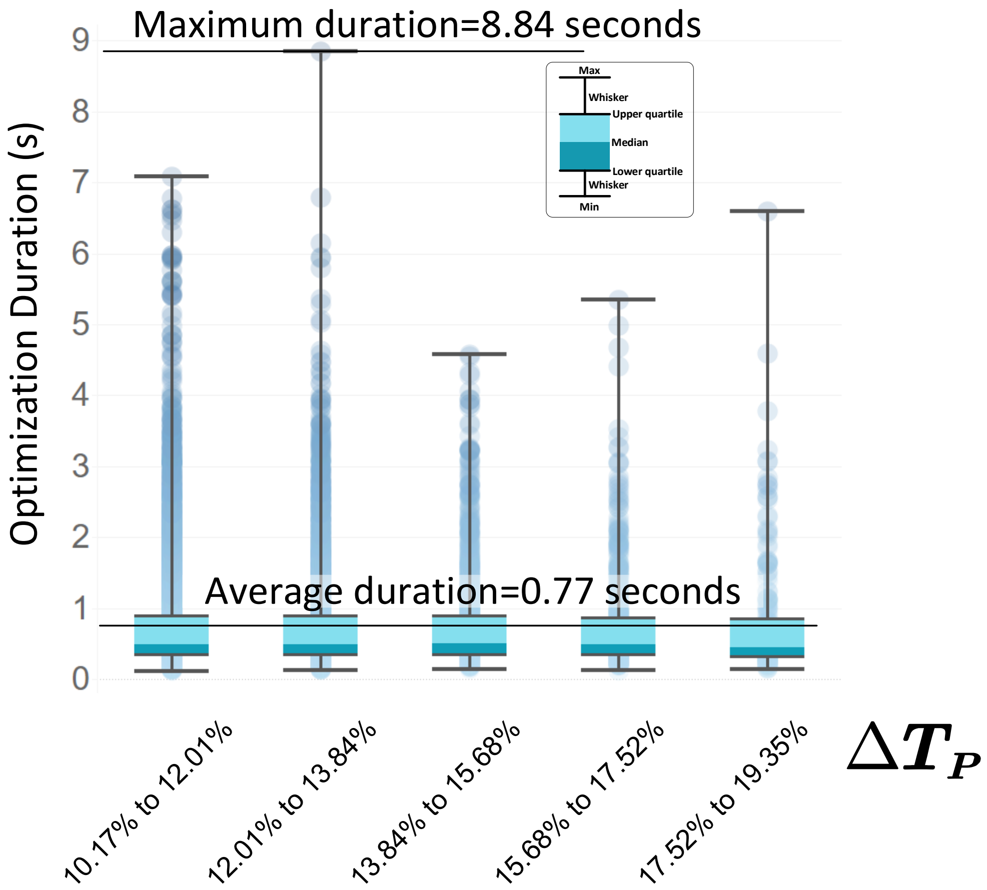


Figure 45: Optimization Duration of Tested Scenarios Under Combined Disturbances

Table 28: Optimization Duration of Tested Scenarios under Mass Change Disturbance

Optimization Duration (s)	≤ 0.34	≤ 0.49	≤ 0.88	≤ 2.38	≤ 8.84
Satisfied Scenarios	2939	5878	8817	11168	11755
Satisfied Percentage	25%	50%	75%	95%	100%

4.2 Summary

The statistical results of the tested scenarios in the above-mentioned four types of cases are summarized in Table 28. As shown in the table, the average optimization time are 2.36, 0.83, 0.86, 0.77 seconds for the four types of scenarios, which proves that the proposed system can meet the requirements of near real-time decision-making for rail transit control in four types of disturbance scenarios. In addition, the longest optimization time is 16.6s, which is within the acceptable range. In addition to the optimization speed, the proposed system has encouraging performances in energy savings.

In the test scenarios of the first Case (accidental temporary stop of a train), the average, maximum and minimum ESP are 25.98%, 82.91% and 0.00%, respectively. It shows that a train can't follow the predefined speed trajectory to ensure safety under this interference, and the proposed system can save a lot of energy. If the interference does not affect the safe driving of the rear car, the control group and the optimization group adopt the same speed trajectory, so the energy consumption is the same (so the minimum optimized energy consumption is 0.00%). Therefore, the proposed system saves a large amount of energy if a train is not able to follow the predefined speed trajectory to ensure safety. If the interference does not affect the safe driving of the following trains, however, the system would not change the planned speed trajectory that gives 0.00% ESP.

In Case 2 (delay due to passenger flow), the average, highest, and the lowest ESP of the tested scenarios are 10.13%, 51.97% and 0.32%, respectively. The average and maximum ESP are lower than Case 1, while the disturbance of Case 2 happens more frequently in the daily operation.

The average, maximum, and minimum ESP of the tested scenarios in Case 3 are 0.21%, 5.5% and -0.009%, respectively, which prove that the difference between the actual travel weight and the planned travel weight will not have a significant impact on the optimal time distribution of the optimized timetable. Therefore, more attention should be paid to the disturbance of delay than weight change.

Case 4 is the disturbance of combined delay and travel weight change, which shows 10.02%, 56.72%, and 0.00025% average, maximum, and minimum *ESP*, respectively. It is reasonable to believe that the main disturbance affecting the optimal timetable distribution in this case are delays based on the result of Case2 and Case3.

Table 29: Summary of the Tested Scenarios

	Case 1	Case 2	Case 3	Case 4
Tested Scenarios	1212	8619	26537	11755
Average *ESP*	25.98%	10.13%	0.21%	10.02%
Maximum *ESP*	82.91%	51.97%	5.50%	56.72%
Minimum *ESP*	0.00%	0.32%	-0.009%	0.00025%
Average Optimization Duration (s)	2.36	0.83	0.86	0.77
Maximum Optimization Duration (s)	16.6	6.81	7.2	8.84

Chapter 5: Conclusion

This research proposes a MAS system to reduce the extra energy consumption caused by a disturbance in the rail transit system. The decent extensibility and flexibility of MAS are suitable for re-optimizing the influenced train when rail transit encounters interference. A complex rail transit optimization problem could be transformed into several simple problems and solved by coordinating multiple agents.

Furthermore, the proposed system has good affordability. The training, optimization, and simulation procedure are processed on a personal laptop, and the test shows that the agents are able to give feedback within seconds. The test results reflect that the hardware cost of applying the system is acceptable. In practical applications, the programming part can be realized by Python instead of Matlab to reduce the software cost.

The rest of this chapter summarizes the main contributions of the research as well as the advantages and limitations of the proposed system, based on which several directions for further research are derived.

5.1 Main Contribution

This research proposed a multi-agent-based system to realize the near real-time energy-aimed speed trajectory re-optimization and timetable re-scheduling for disturbed trains. The system improves the capability of anti-disturbance for rail transit by coordinating the interaction among multiple agents, and four types of disturbances that commonly exist in rail transit are explored. The first type is that a train needs to stop temporarily in an emergency situation, which may affect the normal travel of the following trains. The second type is that the actual dwell time of a train at a station exceeds the planned one due to the heavy passenger flow or other reasons. The third one is that the travel weight change of a train among different sections due to the fluctuation of the number of boarding and alighting passengers. The rest type is the combination of the second and the third one, that is, the train dwell time and travel weight change simultaneously. A large amount of scenarios are simulated in the case study section for each type of disturbance to test the performance and reliability performance of the proposed system. The results prove that the system has advantages in the following aspects.

The first advantage is that the proposed system shows decent energy-saving performance in different types of disturbances. There were 1212, 8619, 26537 and 11755 scenarios that tested for the four disturbance categories. The first category achieves a 25.98% average ESP compared with the found nearest MAS research in rail transit (proposed by Hassanabadi et al. In 2015). The second category achieves a 10.13% average ESP compared with the method of catching up directly when the delay occurs. The third category achieves a 0.21% average ESP compared with the original travel plan. Finally, 10.02% average ESP is achieved in the fourth category compared with the method of catching up directly when the delay occurs. The proposed system has decent non-negative optimization performance for all scenarios in the first, second, and last categories of disturbances, and most scenarios of the third one. Furthermore, the maximum additional energy consumption caused by the system is -0.009% in the third category, which is neglectable. It is expected that negative optimization can be avoided by using larger data sets to train the neural network in practical applications.

In addition, the proposed system shows good optimization speed performance, which is significant for the fast response requirement in applications. The average optimization time of the proposed system is 2.36 seconds, 0.83 seconds, 0.86 seconds, and 0.77 seconds respectively, and the longest optimization time is 16.6 seconds, 6.81 seconds, 7.2 seconds, and 8.84 seconds respectively. The result proves the proposed system is able to support near real-time decision-making when the train system encounters disturbance in practice. In addition, the feedback confirmation mechanism between train agents ensures that the train will decelerate if the optimization time is too long to guarantee safety.

Finally, the established system is based on a hybrid multi-agent structure, which has decent stability and flexibility. Three types of agents (train agent, station agent, and central agent) interact and make decisions through appropriate logic and algorithms in the proposed system. An agent is able to communicate through other information exchange channels when the agent does not receive the expected feedback from the default channel, which provides higher reliability than the system that only relies on a single communication channel. In addition, various functions can be flexibly realized by deploying new sensors, algorithms, and actuators to corresponding agents with the continuous development of hardware devices.

5.2 Limitations and Futureworks

Part of the functions mentioned in this research expects to use the images collected by the monitoring system and the objective recognition algorithm to identify the number of passengers. However, the effectiveness of adding the image recognition algorithm has not been verified in this research due to the lack of data. Besides, using the image data may cause some ethical problems (such as whether rail transit companies have the authority to use passenger images). Although the case study section has proved that the proposed method can still achieve obvious energy savings without image recognition, it is still worth verifying whether the system performance can be further improved with the participation of monitoring image. Furthermore, the algorithms based on supervised learning need tremendous data. Although more than a hundred thousand data are generated in training the DNN, a few scenarios have minor negative optimization resulting in the disturbance of train weight change. The dependence on data requires the model to be fully trained before it is applied to practical engineering. Finally, the research focus is on the small disturbance that will not lead to train cancellation, the mechanisms that enable the system to deal with the large disturbance that leads to train cancellation are insufficiently discussed. Finally, the proposed research focuses on small-scale disturbances, and the mechanism to deal with disruptions that lead to cancelling of trains has not been designed at this stage.

Future research can cooperate with rail transit companies and collect monitoring data of train stations from them. Then adopt the actual data into the training process to verify the efficiency of combining the image recognition algorithm with the train weight prediction. Furthermore, suitable mathematical optimization models and optimization methods can be established to replace neural networks so as to improve the flexibility and applicability of the systems. Another research direction is to discover the potential of MAS to deal with large-scale disruptions that cause cancelling of trains.

Reference

[1] Albrecht, A., Howlett, P., Pudney, P., Vu, X., & Zhou, P. (2016a). The key principles of optimal train control—Part 1: Formulation of the model, strategies of optimal type, evolutionary lines, location of optimal switching points. *Transportation Research Part B: Methodological*. https://doi.org/10.1016/j.trb.2015.07.023

[2] Albrecht, A., Howlett, P., Pudney, P., Vu, X., & Zhou, P. (2016b). The key principles of optimal train control—Part 2: Existence of an optimal strategy, the local energy minimization principle, uniqueness, computational techniques. *Transportation Research Part B: Methodological*. https://doi.org/10.1016/j.trb.2015.07.024

[3] Albrecht, T. (2004). Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control. *Advances in Transport*, *15*, 885–894.

[4] Albrecht, T., & Oettich, S. (2002). A new integrated approach to dynamic-schedule synchronization and energy-saving train control. *Computers in Railways Viii*, *13*, 847–856.

[5] Apter, M. J. (2018). *The Computer Simulation of Behaviour*. Routledge.

[6] Asnis, I. A., Dmitruk, A. V., & Osmolovskii, N. P. (1985). Solution of the problem of the energetically optimal control of the motion of a train by the maximum principle. *USSR Computational Mathematics and Mathematical Physics*. https://doi.org/10.1016/0041-5553(85)90006-0

[7] Balaji, P. G., & Srinivasan, D. (2010). An Introduction to Multi-Agent Systems. In D. Srinivasan & L. C. Jain (Eds.), *Innovations in Multi-Agent Systems and Applications—1* (pp. 1–27). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-14435-6_1

[8] Benjamin, B. R., Milroy, I. P., & Pudney, P. J. (1989). Energy-efficient operation of long-haul trains. *Fourth International Heavy Haul Railway Conference 1989: Railways in Action; Preprints of Papers, The*, 369.

[9] Binder, A., & Albrecht, T. (2013). Timetable evaluation and optimization under consideration of the stochastic influence of the dwell times. *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transportation Systems 2013*, *3*, 471–481.

[10] Bocharnikov, Y. V., Tobias, A. M., Roberts, C., Hillmansen, S., & Goodman, C. J. (2007a). Optimal driving strategy for traction energy saving on DC suburban railways. *IET Electric Power Applications*, *1*(5), 675–682. https://doi.org/10.1049/iet-epa:20070005

[11] Bu, B., Yang, J., Wen, S., & Zhu, L. (2013). Predictive Function Control for Communication-Based Train Control (CBTC) Systems. *International Journal of Advanced Robotic Systems*, *10*(1), 79. https://doi.org/10.5772/53514

[12] Burckert, H., Fischer, K., & Vierke, G. (1998). Transportation Scheduling with holonic MAS: The Teletruck Approach. *Intelligent Agents*.

[13] Cucala, A. P., Fernandez, A., Sicre, C., & Dominguez, M. (2012). Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response. *Engineering Applications of Artificial Intelligence*, *25*(8), 1548–1557. https://doi.org/10.1016/j.engappai.2012.02.006

[14] Dalapati, P., Agarwal, P., Dutta, A., & Bhattacharya, S. (2016). *Real-time Rescheduling in Distributed Railway Network: An Agent-Based Approach*. 1–26.

[15] De Persis, C., Sailer, R., & Wirth, F. (2013). Parsimonious event-triggered distributed control: A Zeno free approach. *Automatica*, *49*(7), 2116–2124. https://doi.org/10.1016/j.automatica.2013.03.003

[16] Ding, L., Han, Q.-L., & Guo, G. (2013). Network-based leader-following consensus for distributed multi-agent systems. *Automatica*, *49*(7), 2281–2286. https://doi.org/10.1016/j.automatica.2013.04.021

[17] Ding, Y., Liu, H., Bai, Y., & Zhou, F. (2011). A Two-level Optimization Model and Algorithm for Energy-Efficient Urban Train Operation. *Journal of Transportation Systems Engineering and Information Technology*, *11*(1), 96–101. https://doi.org/10.1016/S1570-6672(10)60106-7

[18] Feitelson, & Eran. (1994). The potential of rail as an environmental solution: Setting the agenda. *Transportation Research Part A Policy & Practice*, *28*(3), 209–221.

[19] Ge, X., Yang, F., & Han, Q. (2017). Distribute d networke d control systems: A brief overview. *Information Sciences*, *380*, 117–131. https://doi.org/10.1016/j.ins.2015.07.047

[20] Ghoseiri, K., Szidarovszky, F., & Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation Research Part B-Methodological*, *38*(10), 927–952. https://doi.org/10.1016/j.trb.2004.02.004

[21] Guinaldo, M., Lehmann, D., Sánchez, J., Dormido, S., & Johansson, K. H. (2014). Distributed event-triggered control for non-reliable networks. *Journal of the Franklin Institute*, *351*(12), 5250–5273. https://doi.org/10.1016/j.jfranklin.2014.09.004

[22] Guo, G., Ding, L., & Han, Q.-L. (2014). A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems. *Automatica*, *50*(5), 1489–1496. https://doi.org/10.1016/j.automatica.2014.03.017

[23] Guo, Y., Zhang, C., Wu, C., & Lu, S. (2021). Multiagent System–Based Near-Real-Time Trajectory and Microscopic Timetable Optimization for Rail Transit Network. *Journal of Transportation Engineering, Part A: Systems*, *147*(2), 04020153. https://doi.org/10.1061/jtepbs.0000473

[24] Haahr, J. T., Pisinger, D., & Sabbaghian, M. (2017). A dynamic programming approach for optimizing train speed profiles with speed restrictions and passage points. *Transportation Research Part B: Methodological*, *99*, 167–182. https://doi.org/10.1016/j.trb.2016.12.016

[25] Hassanabadi, H., Moaveni, B., & Karimi, M. (2015). A comprehensive distributed architecture for railway traffic control using multi-agent systems. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *229*(2), 109–124. https://doi.org/10.1177/0954409713503458

[26] Howlett, P. G. (2000). The optimal control of a train. *Annals of Operations Research*, *98*(98), 65–87. https://doi.org/10.1023/a:1019235819716

[27] Howlett, P. G., Milroy, I. P., & Pudney, P. J. (1994). Energy-efficient train control. *Control Engineering Practice*. https://doi.org/10.1016/0967-0661(94)90198-8

[28] Huang, M. (2014). *Study on Functional Safety Analysis of CBTC on-board ATP System*. Southwest Jiaotong University.

[29] Hui, G., Zhang, H., Wu, Z., & Wang, Y. (2014). Control synthesis problem for networked linear sampled-data control systems with band-limited channels. *Information Sciences*, *275*, 385–399. https://doi.org/10.1016/j.ins.2014.01.042

[30] Ichikawa, K. (1968). Application of Optimization Theory for Bounded State Variable Problems to the Operation of Train. *Bulletin of JSME*. https://doi.org/10.1299/jsme1958.11.857

[31] Jia, X.-C., Chi, X.-B., Han, Q.-L., & Zheng, N.-N. (2014). Event-triggered fuzzy HO control for a class of nonlinear networked control systems using the deviation bounds of asynchronous normalized membership functions. *Information Sciences*, *259*, 100–117. https://doi.org/10.1016/j.ins.2013.08.055

[32] Khmelnitsky, E. (2000). On an optimal control problem of train operation. *IEEE Transactions on Automatic Control*, *45*(7), 1257–1266. https://doi.org/10.1109/9.867018

[33] Kokotovic, P., & Singh, G. (1972). Minimum-energy control of a traction motor. *IEEE Transactions on Automatic Control*, *17*(1), 92–95.

[34] Li, X., & Lo, H. K. (2014). An energy-efficient scheduling and speed control approach for metro rail operations. *Transportation Research Part B: Methodological*, *64*, 73–89. https://doi.org/10.1016/j.trb.2014.03.006

[35] Lind, J., & Fischer, K. (1999). Transportation scheduling and simulation in a railroad scenario: A multi-agent approach. *Logistik Management*, 171–183.

[36] Litman, T. (2020). Rail Transit In America: A Comprehensive Evaluation of Benefits. In *Transportation Research Record*.

[37] Liu, F., Xun, J., & Bin, N. (2017). An optimization method for train driving trajectory in urban rail systems. *Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016*, 413–418. https://doi.org/10.1109/YAC.2016.7804929

[38] Liu, F., Xun, J., & Bin, N. (2016). An optimization method for train driving trajectory in urban rail systems. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 413–418. https://doi.org/10.1109/YAC.2016.7804929

[39] Liu, H., Tian, H., & Li, Y. (2015). An EMD-recursive ARIMA method to predict wind speed for railway strong wind warning system. *Journal of Wind Engineering and Industrial Aerodynamics*, *141*, 27–38. https://doi.org/10.1016/j.jweia.2015.02.004

[40] Liu, Q., Wang, Z., He, X., & Zhou, D. H. (2015). Event-Based H<inf>∞</inf> Consensus Control of Multi-Agent Systems with Relative Output Feedback: The Finite-Horizon Case. *IEEE Transactions on Automatic Control*, *60*(9), 2553–2558. https://doi.org/10.1109/TAC.2015.2394872

[41] Liu, R., & Golovitcher, I. M. (2003). Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, *37*(10), 917–932. https://doi.org/10.1016/j.tra.2003.07.001

[42] Lu, S., Hillmansen, S., Ho, T. K., & Roberts, C. (2013a). Single-train trajectory optimization. *IEEE Transactions on Intelligent Transportation Systems*, *14*(2), 743–750. https://doi.org/10.1109/TITS.2012.2234118

[43] Lu, S., Wang, M. Q., Weston, P., Chen, S., & Yang, J. (2016). Partial Train Speed Trajectory Optimization Using Mixed-Integer Linear Programming. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2911–2920. https://doi.org/10.1109/TITS.2016.2535399

[44] Lu, S., Weston, P., Hillmansen, S., Gooi, H. B., & Roberts, C. (2014). Increasing the regenerative braking energy for railway vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *15*(6), 2506–2515. https://doi.org/10.1109/TITS.2014.2319233

[45] Mills, R., Perkins, S., & Pudney, P. (1991). Dynamic rescheduling of long-haul trains for improved timekeeping and energy-conservation. *Asia-Pacific Journal of Operational Research*, *8*(2), 146–165.

[46] Miyatake, M., & Matsuda, K. (2009). Energy saving speed and charge/discharge control of a railway vehicle with on-board energy storage by means of an optimization model. *IEEJ Transactions on Electrical and Electronic Engineering*, *4*(6), 771–778. https://doi.org/10.1002/tee.20479

[47] Ortega, F. A., Pozo, M. A., & Puerto, J. (2018). On-line timetable rescheduling in a transit line. *Transportation Science*, *52*(5), 1106–1121. https://doi.org/10.1287/trsc.2017.0807

[48] Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc.

[49] Pena-Alcaraz, M., Fernandez, A., Paloma Cucala, A., Ramos, A., & Pecharroman, R. R. (2012). Optimal underground timetable design based on power flow for maximizing the use of regenerative-braking energy. *Proceedings of the Institution of Mechanical Engineers Part F-Journal of Rail and Rapid Transit*, *226*(F4), 397–408. https://doi.org/10.1177/0954409711429411

[50] Peng, C., & Han, Q.-L. (2013). A novel event-triggered transmission scheme and l2 control co-design for sampled-data control systems. *IEEE Transactions on Automatic Control*, *58*(10), 2620–2626. https://doi.org/10.1109/TAC.2013.2256015

[51] Peng, C., Han, Q.-L., & Yue, D. (2013). To transmit or not to transmit: A discrete event-triggered communication scheme for networked takagi-sugeno fuzzy systems. *IEEE Transactions on Fuzzy Systems*, *21*(1), 164–170. https://doi.org/10.1109/TFUZZ.2012.2199994

[52] Proenca, H., & Oliveira, E. (2004). MARCS Multi-agent Railway Control System. *Ibero-American Conference on Artificial Intelligence*, 12–21. https://doi.org/10.1007/978-3-540-30498-2_2

[53] Rousset, A., Herrmann, B., Lang, C., & Philippe, L. (2016). A survey on parallel and distributed multi-agent systems for high performance computing simulations. *Computer Science Review*, *22*, 27–46. https://doi.org/10.1016/j.cosrev.2016.08.001

[54] Russell, S., & Norvig, P. (1996). Artificial Intelligence A Modern Approach Second Edition. In *Artificial Intelligence*.

[55] Scheepmaker, G. M., Goverde, R. M. P., & Kroon, L. G. (2017). Review of energy-efficient train control and timetabling. *European Journal of Operational Research*, *257*(2), 355–376. https://doi.org/10.1016/j.ejor.2016.09.044

[56] Tan, Z., Lu, S., Bao, K., Zhang, S., Wu, C., Yang, J., & Xue, F. (2018). Adaptive Partial Train Speed Trajectory Optimization. *Energies*, *11*(12), 3302. https://doi.org/10.3390/en11123302

[57] Tian, Z., Weston, P., Zhao, N., Hillmansen, S., Roberts, C., & Chen, L. (2017). System energy optimisation strategies for metros with regeneration. *Transportation Research Part C: Emerging Technologies*, *75*, 120–135. https://doi.org/10.1016/j.trc.2016.12.004

[58] Verma, A., & Pattanaik, K. K. (2014). Mobile Agent based Train Control System for Mitigating Meet Conflict at Turnout. *Procedia Computer Science*, *32*, 317–324. https://doi.org/10.1016/j.procs.2014.05.430

[59] Wang, P., & Goverde, R. M. P. (2016). Multiple-phase train trajectory optimization with signalling and operational constraints. *Transportation Research Part C: Emerging Technologies*, *69*, 255–275. https://doi.org/10.1016/j.trc.2016.06.008

[60] Wang, P., & Goverde, R. M. P. (2017). Multi-train trajectory optimization for energy efficiency and delay recovery on single-track railway lines. *Transportation Research Part B: Methodological*, *105*, 340–361. https://doi.org/10.1016/j.trb.2017.09.012

[61] Wang, Y., De Schutter, B., van den Boom, T. J. J., & Ning, B. (2013). Optimal trajectory planning for trains—A pseudospectral method and a mixed integer linear programming approach. *Transportation Research Part C: Emerging Technologies*. https://doi.org/10.1016/j.trc.2013.01.007

[62] Wang, Y.-L., & Han, Q.-L. (2015). Quantitative analysis and synthesis for networked control systems with non-uniformly distributed packet dropouts and interval time-varying sampling periods. *International Journal of Robust and Nonlinear Control*, *25*(2), 282–300. https://doi.org/10.1002/rnc.3087

[63] Wang, Z., Ding, D., Dong, H., & Shu, H. (2013). H∞ consensus control for multi-agent systems with missing measurements: The finite-horizon case. *Systems and Control Letters*, *62*(10), 827–836. https://doi.org/10.1016/j.sysconle.2013.06.004

[64] Weiss, G. (2012). Multiagent Systems. In *Foundations* (2nd ed.). The MIT Press.

[65] Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*. https://doi.org/10.1017/S0269888900008122

[66] Xun, J., Ning, B., Li, K. ping, & Zhang, W. bin. (2013). The impact of end-to-end communication delay on railway traffic flow using cellular automata model. *Transportation Research Part C: Emerging Technologies*, *35*, 127–140. https://doi.org/10.1016/j.trc.2013.06.008

[67] Yang, G., Zhang, F., Gong, C., & Zhang, S. (2019). Application of a Deep Deterministic Policy Gradient. *Energies*, *12*, 1–19.

[68] Yang, X., Chen, A., Li, X., Ning, B., & Tang, T. (2015). An energy-efficient scheduling approach to improve the utilization of regenerative energy for metro systems. *Transportation Research Part C: Emerging Technologies*, *57*, 13–29. https://doi.org/10.1016/j.trc.2015.05.002

[69] Yang, X., Chen, A., Ning, B., & Tang, T. (2016). A stochastic model for the integrated optimization on metro timetable and speed profile with uncertain train mass. *Transportation Research Part B: Methodological*, *91*, 424–445. https://doi.org/10.1016/j.trb.2016.06.006

[70] Yang, X., Li, X., Gao, Z., Wang, H., & Tang, T. (2013). A Cooperative Scheduling Model for Timetable Optimization in Subway Systems. *Ieee Transactions on Intelligent Transportation Systems*, *14*(1), 438–447. https://doi.org/10.1109/TITS.2012.2219620

[71] Yang, X., Ning, B., Li, X., & Tang, T. (2014). A Two-Objective Timetable Optimization Model in Subway Systems. *Ieee Transactions on Intelligent Transportation Systems*, *15*(5), 1913–1921. https://doi.org/10.1109/TITS.2014.2303146

[72] You, K., Li, Z., & Xie, L. (2013). Consensus condition for linear multi-agent systems over randomly switching topologies. *Automatica*, *49*(10), 3125–3132. https://doi.org/10.1016/j.automatica.2013.07.024

[73] Yue, D., Tian, E., & Han, Q. L. (2013). A delay system method for designing event-triggered controllers of networked control systems. *IEEE Transactions on Automatic Control*. https://doi.org/10.1109/TAC.2012.2206694

[74] Zhang, C., & Hammad, A. (2012). Multiagent Approach for Real-Time Collision Avoidance and Path Replanning for Cranes. *Journal of Computing in Civil Engineering*. https://doi.org/10.1061/(asce)cp.1943-5487.0000181

[75] Zhang, H., Feng, G., Yan, H., & Chen, Q. (2014). Observer-based output feedback event-triggered control for consensus of multi-agent systems. *IEEE Transactions on Industrial Electronics*, *61*(9), 4885–4894. https://doi.org/10.1109/TIE.2013.2290757

[76] Zhang, X.-M., & Han, Q.-L. (2014). Event-triggered dynamic output feedback control for networked control systems. *IET Control Theory & Applications*. https://doi.org/10.1049/iet-cta.2013.0253

[77] Zhao, N., Roberts, C., Hillmansen, S., & Nicholson, G. (2015). A Multiple Train Trajectory Optimization to Minimize Energy Consumption and Delay. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2363–2372. https://doi.org/10.1109/TITS.2014.2388356

[78] Zhao, N., Roberts, C., Hillmansen, S., Tian, Z., Weston, P., & Chen, L. (2017). An integrated metro operation optimization to minimize energy consumption. *Transportation Research Part C: Emerging Technologies*, *75*, 168–182. https://doi.org/10.1016/j.trc.2016.12.013

[79] Zhu, Y., & Goverde, R. M. P. (2020). Dynamic and robust timetable rescheduling for uncertain railway disruptions. *Journal of Rail Transport Planning and Management*, *15*(July 2019), 100196. https://doi.org/10.1016/j.jrtpm.2020.100196