Scheduling local and express trains in suburban rail transit lines: Mixed-integer nonlinear programming and adaptive genetic algorithm

Article *in* Computers & Operations Research · June 2021

DOI: 10.1016/j.cor.2021.105436

Andrea D'Ariano
Roma Tre University

Yantong Li
Dalian Maritime University

Marcella Samà
Roma Tre University

Scheduling local and express trains in suburban rail transit lines: mixed-integer nonlinear programming and adaptive genetic algorithm

Lianhua Tang[1], Andrea D'Ariano[2], Xingfang Xu[1*], Yantong Li[3], Xiaobing Ding[4], Marcella Samà[2]

1.The Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, 201804 Shanghai, China

2.Dipartimento di Ingegneria, Sezione di Informatica e Automazione, Università degli Studi Roma Tre, via della Vasca Navale, 79, 00146 Rome, Italy

3.School of Maritime Economics and Management, Dalian Maritime University, 116026 Dalian, China

4.School of Urban Rail Transportation, Shanghai University of Engineering Science, 201620 Shanghai, China

We investigate the train timetabling problem in suburban rail transit lines by considering 1) the traditional stopping mode (TSM), in which all trains stop at each station, and 2) the express/local stopping mode (ELM), in which express trains can skip certain low-demand stations. We first propose two mixed-integer linear programming models for the train timetabling problem under the TSM with and without capacity constraints. Next, we develop two mixed-integer nonlinear programming models under the ELM with and without "overtaking"; thus, a total of four optimization models are proposed. The objective is to minimize the passenger travel time (PTT). Owing to the NP-hardness of the studied problem, we propose an adaptive genetic algorithm (A-GA) that can efficiently solve the four proposed models. The A-GA is customized to solve the train timetabling problem with train capacity, overtaking, and other operational constraints, reducing the PTT. To evaluate the performance of the proposed algorithm, we conduct numerical experiments on 60 randomly generated realistic instances and a real-world case study based on Shanghai Metro Line 16. The computational results for the realistic instances indicate that our A-GA can obtain near-optimal solutions with significantly less computation time than an established commercial solver. The computational results from the real-world case study quantify the benefits of considering the combination of the ELM and overtaking strategies in train timetabling. Furthermore, we perform a sensitivity analysis on key parameters of our mathematical formulations. The results provide insights to railway managers on how to set key parameters when applying the proposed formulations and solution methodology in practice.

Key words: Suburban rail transit (SRT); Train timetabling; Traditional (all) stopping mode (TSM); Express/local stopping mode (ELM); Passenger travel time (PTT); Train overtaking; Adaptive genetic algorithm (A-GA); Mixed-integer nonlinear programming (MINLP).

1. Introduction

The urban rail transit system is considered as an efficient transportation mode (with large infrastructure and vehicle capacity, good punctuality rates, limited energy consumption and pollution, and high passenger safety) and plays an important role in the economic and social development of large and densely populated cities (e.g., Tokyo, Shanghai, Beijing, London, and New York). The objective of optimizing train timetables is to determine the best departure and arrival times of each train at each station according to key performance indicators; this is one of the fundamental problems in rail transit operations. Optimized train timetables should not only make full use of the rail infrastructure and train capacity but also improve the passenger service quality, e.g., by reducing the passenger travel time (PTT).

The advantage of suburban railway transport is becoming increasingly evident as urban rail and road traffic become more crowded; this has led to the rapid growth of suburban passenger flows and the subsequent development of suburban rail transit (SRT) systems. Because the origin–destination (OD) passenger demand varies significantly for each station among a SRT line, the number of passengers entering and leaving some key stations in residential areas or the city center is comparatively large. Consequently, for many urban rail systems, special stopping modes have been developed, e.g., skip–stop, zonal, or express/local, as reported by Vuchic (2005). In the present study, the express/local stopping mode (ELM) is examined, which can provide both direct trips (i.e., local trains that stop at each station) and higher-speed services (i.e., express trains that skip certain low-demand stations, which can reduce the in-vehicle times for medium and long-distance passengers). Applying the ELM to train timetabling can limit the operational costs, reduce the total PTT, and enhance the passenger satisfaction (Gao et al., 2016; Li et al., 2019; Yang et al., 2020).

Despite the numerous benefits, when the ELM is applied to train timetabling, the practitioners face a more complex train timetabling problem, in which express trains may overtake local trains to fully exploit the ELM potential. Specifically, it is difficult to accurately calculate the passenger waiting time and in-vehicle time under the ELM, because 1) determining which train a passenger can board under the constraint of the limited train capacity is a complex problem and 2) the train schedule, train sequence, number of express/local trains, and locations of overtaking stations are variables to be solved for. Researchers have started quantifying the benefits of ELM strategies in the timetabling process. However, timetabling that considers both the ELM and train overtaking has rarely been addressed, despite its considerable potential for improving rail operations. The lack of such studies is mainly due to the complexity involved in considering both the train sequencing and the passenger flow in timetabling. As reported by Meng et al. (2018), owing to the high complexity of urban rail transit systems, most research thus far has focused on the optimization of a single control strategy.

The train timetabling problem has attracted significant attention from railway operations researchers in recent years. [See, for example, the literature reviews of Cacchiani et al. (2015) and

Hansen and Pachl (2014).] A growing stream of train timetabling research for urban rail has considered 1) how to embed passenger demand in this problem (Niu and Zhou, 2013; Barrena et al., 2014; Wang et al., 2015; Huang et al., 2016; Wang, Liao, et al., 2017; Hassannayebi and Zegordi, 2017; Wang et al., 2018; Robenek et al., 2018; Qi et al., 2018; Meng and Zhou, 2019; Xie et al., 2020; Liu et al., 2020; Cacchiani et al., 2020; Yin et al., 2021); 2) how to formulate a joint optimization model for train scheduling and maintenance planning problems (Luan et al., 2017; Huang et al., 2017; Zhang et al., 2020), train circulation planning (Wang, Tang, et al., 2017), or energy-efficient traffic optimization (Yin et al., 2017; Mo et al.,2019); and 3) how to generate recovery strategies (Huang et al., 2020), schedule additional trains (Gao et al., 2018), or reschedule trains with passenger reassignment under large disruptions (Hong et al., 2021). From the perspective of the passengers, adapting the timetables to their demand helps minimize the in-vehicle and waiting times. From the viewpoint of the train operation company (TOC), the number of required trains can be reduced during the off-peak hours, which can reduce the operating costs. Canca et al. (2014) investigated a train scheduling approach to deal with dynamic demand and proposed “modeling tricks” to measure the passenger boarding and alighting times, on-vehicle passengers, and vehicle occupancy. Thus, passenger comfort and safety can be improved during peak hours because the number of required trains can be adjusted dynamically according to the passenger demand, which may improve the service quality.

With the construction of new SRT lines in developed countries, researchers have addressed train timetabling problems involving skip–stop patterns and ELMs. Regarding the management of skip–stop patterns in SRT, Wang et al. (2014) considered a train scheduling problem for urban rail transit systems with skip–stop strategies, aiming to minimize the total travel times of the passengers and the energy consumption of the trains, and proposed an efficient bi-level optimization approach. Niu et al. (2015) focused on minimizing the total passenger waiting times at stations when applying timetables with skip–stop strategies. Zhang et al. (2016) proposed a skip–stop scheme for urban rail lines aiming to minimize the total PTT and maximize the number of passengers served. Zhang et al. (2018) solved the train timetabling problem with an approach that adapted itself to dynamic passenger demands by optimizing the train running times, dwelling times, and number of required trains. Shang et al. (2018) used train skip–stopping patterns to improve the system-wide equity performance in an oversaturated urban rail transit network. However, the foregoing studies did not consider train overtaking. Yang et al. (2016) proposed a collaborative methodology to optimize both train stops and scheduling for minimizing the total travel time of all trains and solved the model to near optimality with a commercial solver. However, because their study focused on optimizing the train schedule on a tactical level instead of an operational level, they did not track the precise number of passengers boarding or alighting from each train at each station; rather, they simply estimated the loading capacity for each train and the number of passengers waiting at each station. Yue et al. (2016) and Jiang et al. (2017) considered similar objective functions from the perspective of the TOC; that is, maximizing the total profit reduced by penalties based on the stopping times and number of stops of each train (Yue et al., 2016) and maximizing the number of scheduled trains while avoiding stop cancelations (Jiang et al., 2017). However, neither of them explicitly considered the passenger demand. Dong et al. (2020) considered the integrated optimization of train stop planning and timetabling; however, they did not consider overtaking.

Another research stream on train timetabling for SRT is related to the ELM. This strategy has been adopted in large cities around the world, including Paris, New York, Tokyo, Beijing, and Shanghai. Gao et al. (2016) formulated a bi-objective train timetable optimization problem based on an ELM strategy and computed compromise solutions between energy consumption reduction and PTT minimization. Li et al. (2019) developed a mixed-integer nonlinear programming (MINLP) formulation to collaboratively optimize the train stopping patterns and scheduling, with the aim of minimizing the total PTT. However, the formulations proposed in the latter two studies assume that the proportions of express and local trains (ELTs) are balanced and that the headways between subsequent express/local trains are identical. Furthermore, oversaturated traffic conditions are not considered.

Other recent studies have focused on optimizing the stopping plan rather than the timetable. Gu et al. (2016) developed mathematical models to optimize a set of strategic variables, such as the stop density and the vehicle headway. Yang et al. (2020) developed a bi-objective model to optimize the stopping pattern, frequencies, and turn-back stations on the premise of "no overtaking facilities". They assumed that the arrival time of passengers at each station was evenly distributed, and express/local trains departed alternately to ensure a balanced service for the waiting passengers at each station. Their methodology focused on off-peak hours, when the passenger demand was low and the service frequency was moderate.

Table 1 presents a systematic review of studies on special stopping modes. Related studies are characterized by the stopping mode, variables, objective, accuracy of the passenger waiting time, train capacity, overtaking, and sequence.

The above review reveals that optimization of train timetables has been well studied. However, recent literature on train timetabling under the ELM has rarely addressed the train capacity; in these studies, it was often assumed that the train capacity was sufficient for all the passengers, which is not always realistic. In numerous studies on skip–stop and ELMs, the methodology of Clarke (1995) has been used to calculate the passenger waiting time, which was assumed to be approximately half of the inter-departure time interval. In the most recent studies on ELMs, the train sequence was assumed to comprise an express train followed by a slow train; that is, the

Table 1 Literature on special stopping modes

Paper	Stopping mode	Variables	Objective	PWT	Train capacity	Train overtaking	Train sequence
Wang et al. (2014)	Skip-stop	Timetable &Stop plan	PTT &EC	Average	√	×	Variable
Niu et al. (2015)	Skip-stop	Timetable	PWT	Accurate	×	×	Variable
Zhang et. (2016)	Skip-stop	Timetable	PTT &MNSP	Average	√	×	Variable
Zhang et al. (2018)	Skip-stop	Timetable	PTT	Average	√	×	Variable
Shang et al. (2018)	Skip-stop	Stop plan	MNPSM	——	√	×	Variable
Yang et al. (2016)	Skip-stop	Timetable &Stop plan	TTT	——	√	√	Variable
Dong et al. (2020)	Skip-stop	Timetable &Stop plan	PTE &TTT	Average	√	×	Variable
Gao et al. (2016)	Express/local	Timetable	PTT &EC	Average	×	√	Fixed
Li et al. (2019)	Express/local	Timetable &Stop plan	PTT	Average	×	√	Fixed
Gu et al. (2016)	Express/local	Stop plan	PTT &AOC	Average	√	√	Variable
Yang et al. (2020)	Express/local	Stop plan	PTT &AOC	Average	×	×	Fixed
This paper	Express/local	Timetable	PTT	Accurate	√	√	Variable

Notation in Column 4: EC = energy consumption; PTT = passenger travel time; PWT = passenger total waiting time; MNSP = maximum number of passengers served; MNPSM=minimum number of passengers who suffer the maximum number of missed trains; TTT = train total running time; PTE = passenger travel efficiency; AOC = agency operations cost.

proportions of ELTs were balanced. However, this assumption is not easily adaptable to different traffic flows in practice. Moreover, overtaking among the trains, which is essential in timetabling with ELM, is rarely studied. In the present study, we simultaneously optimize the train sequence, number of express/local trains, locations of overtaking stations, and train arrival and departure times to minimize the PTT. The contributions of this study can be summarized as follows:

We first propose mixed-integer programming (MIP) formulations for the SRT tactical timetabling process to address 1) the traditional stopping mode (TSM), both with and without train capacity constraints, and 2) the ELM, both with and without train-overtaking variables. The former two models are designed to demonstrate the complexity of this problem and to highlight the importance of developing ad-hoc solution techniques. The latter two models are designed to investigate the potential benefits of the ELM strategies over the TSM strategies.

The proposed models have three novel features compared with the models presented in Table 1. First, our objective function addresses the total PTT instead of the average waiting time or train travel time. Second, passengers being left behind are rigorously considered by means of seat-capacity constraints in trains under congested traffic conditions. To this end, we develop a methodology to track the number of passengers waiting at each platform and boarding/alighting from each train. Third, we precisely calculate passenger waiting times by jointly considering express/local stop patterns and passengers left behind.

Our preliminary results indicate that commercial solvers can only optimally solve small instances and cannot obtain feasible solutions for practical-size instances. Therefore, in this study, we developed an adaptive genetic algorithm (A-GA) to efficiently solve the proposed MIP formulations. Furthermore, we conducted numerous numerical experiments to evaluate the performance of the proposed algorithm. The first set of experiments was based on realistic instances and demonstrated that the A-GA could obtain near-optimal solutions within an acceptable computation time for the railway planners, while the IBM ILOG CPLEX solver—a state-of-the-art commercial solver—took significantly more time to obtain a solution. In the second set of experiments, which was based on the real-world case of Shanghai Metro Line 16, the ELM strategy reduced the PTT to a greater degree than the TSM strategy. A sensitivity analysis was performed to investigate the extent to which the PTT could be influenced by the passenger perception of the waiting times.

The remainder of this paper is organized as follows. Section 2 presents a formal description of the studied problem, and Section 3 describes four mathematical formulations whose numerous variants we propose to solve. Section 4 introduces the proposed metaheuristic algorithm and its customized versions for efficiently solving the formulations described in Section 3. Section 5 presents the studied test case and the instances investigated in this study. Computational experiments were performed on these instances to evaluate the applicability of the methodology. Section 6 presents the main findings related to the computational results as well as key contributions of this research. Possible directions for extending the proposed methodology are also presented.

2. Problem description

We describe the train timetabling problem and outline our assumptions based on the operational characteristics of SRT lines.

2.1. Dwelling process of trains at stations and passenger boarding and alighting process

Different types of timetables are described in Appendix A.1. Let $[0, T_1]$ represent the studied time horizon for the first station, which can be discretized into a set of time nodes: $[0, 1\delta, \cdots, (a-1)\delta, a\delta]$. Thus, the time instant $t \in \{0, 1, \cdots, (a-1), a\}$ corresponds to $t\delta$ time units, from the start of the studied time horizon. The discretization constant δ represents the length of the minimum time interval. P_{kl}^t is defined as the number of passengers traveling from station k to l during the time interval $[t-1, t]$. Furthermore, we assume that the information on passenger arrivals is always available for each time interval. The passenger demand data are commonly available in modern urban transit systems and are collected, for example, through automatic fare collection systems (AFCs), or generated via other demand-forecasting methods.

Because we assume that no passenger is left behind when the last train departs and passengers who arrive at station k later than the departure time of the last train from k cannot be served, the studied time horizon for station k ends according to the departure time of the last train from that station. Similarly, we assume that there is a slow train that departs from the first station at the start time of the studied time horizon. Therefore, passengers who arrive at station k earlier than the departure time of this slow train from station k will be served by other trains that are scheduled prior to the studied time horizon. Consequently, in our assumptions, all the stations have an equivalent time horizon $[\gamma_k^0, T_k]$, where T_k-γ_k^0= T_1 (γ_k^0 and T_k represent the start and end times of the studied time horizon for station k, respectively).

Let i represent the train index. In Figure 1, there are four consecutive trains: i, i_2, i_3, and i_4. Local trains i and i_3 stop at station k, the express train i_2 skips the station, and the express train i_4 overtakes i_3. Two nodes represent the arrival and departure of each train, and one node represents the skip/overtake process. Let t_{ik}^a and t_{ik}^d represent the arrival and departure times of train i at station k. Thus, t_{ik}^a and t_{ik}^d are equal if train i skips station k.

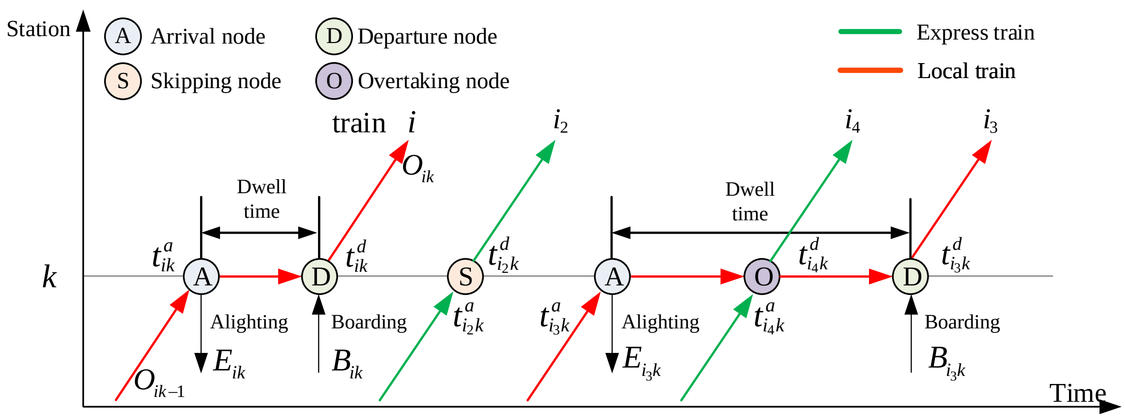


Figure 1 **Illustration of train operation processes at a station**

During a train's dwelling at a station, the passengers perform boarding and alighting activities. The number of passengers alighting from train i at station k is E_{ik}, and the number of passengers on train i when it has left station $k-1$ is Z_{ik-1}. When the current train i arrives at station k, the passengers whose destination is k alight from this train, and the waiting passengers can board the train until the train reaches its maximum capacity. The number of passengers boarding train i at station k is B_{ik}, and the number of passengers on train i as it leaves station k is $Z_{ik-1} - E_{ik} + B_{ik}$. Owing to overcrowding, some passengers may be stranded on the platform.

2.2. Phenomenon of passengers left behind

In train operations with the TSM, the waiting passengers may not always manage to board the first train that arrives at the station if there is overcrowding. In a train operation with the ELM, in addition to the limited train capacity, express trains may skip stations, causing some passengers to wait. Figure 2 shows an example of the two cases where passengers must to wait for the next train at stations where express trains stop. Four consecutive trains—i, i_2, i_3, and i_4—stop at station 3 and depart at times 12, 17, 22, and 26, respectively. We suppose that trains i and i_4 are local and that express trains i_2 and i_3 stop at stations 1, 3, 4, 6, 7, and 9, skipping stations 2, 5, and 8.

Considering that all the passengers take only one direct train to arrive at their destination, the first type of situation where passengers are left behind is caused by the inconsistency in the destinations of the passengers with the stopping scheme of the express trains. p_{kl}^{t} is defined as a passenger traveling from station k to l during the time interval $[t-1,t]$. Passengers p_{35}^{7} and p_{38}^{10} cannot board trains i_2 and i_3, and p_{38}^{18} cannot board train i_3. The second type of situation where passengers are left behind is caused by a limited train capacity; here, passengers p_{35}^{7} and p_{38}^{10} cannot board train i; p_{39}^{16} cannot board train i_2; and p_{35}^{23}, p_{37}^{24}, and p_{38}^{25} cannot board train i_4.

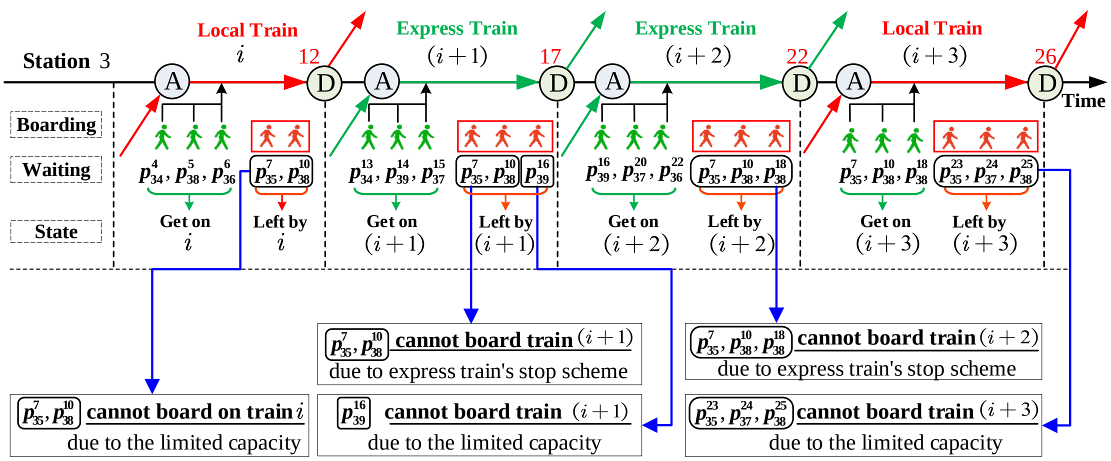


Figure 2 ~~Phenomenon of p~~Passengers~~'~~ left behind when an express train stops

Detailed descriptions of the waiting and in-vehicle times of passengers are presented in Appendix A.2, along with the calculation process for passengers' waiting and in-vehicle times.

2.3. Effective loading times of passengers

To accurately calculate the number of onboard passengers and their waiting times, we introduce the variable A_{ikl}^{t} as an effective loading indicator, which is equal to 1 if passengers who arrive at station k at time t with destination station l have boarded train i. When an express train skips a station, the effective loading time is a null value, because the train leaves all waiting passengers behind. For a stopping station, where express trains necessarily stop, we take trains i and i_2 in Figure 2 as an example to illustrate the effective loading times.

As discussed previously, in our example, passengers p_{34}^{4}, p_{38}^{5}, and p_{36}^{6} can board train i, whereas p_{35}^{7} and p_{38}^{10} cannot. The latest arrival time to board train i is Time 6 (coinciding with the arrival time of passenger p_{36}^{6}).Thus, $\sum_{t>6} A_{ikl}^{t} = 0, \forall k < l \leq s$; that is, the passengers who arrive after Time 6 cannot board train i. As an express train, according to assumption A.2, train i_2 is boarded only by passengers bound for stations, where it stops (i.e., 4, 6, 7, and 9), while other passengers (with destination stations 5 and 8) are stranded. Then, the effective loading indicator for the passengers with destination stations 5 and 8 is always 0; i.e., $A^t_{i_2kl} = 0, \forall l \in [5,8]$. The effective loading time for passengers bound for stations 4, 6, 7, and 9 to board train i_2, which is Time 15 in this case, depends on the corresponding train capacity, that is, $A^t_{i_2kl} = 1, \forall 6 < t \leq 15, l \in [4,6,7,9]$ and $A^t_{i_2kl} = 0, \forall t{>}15, l \in [4,6,7,9]$. Illustrations of effective loading times for trains i and i_2 are shown in Figure 3.

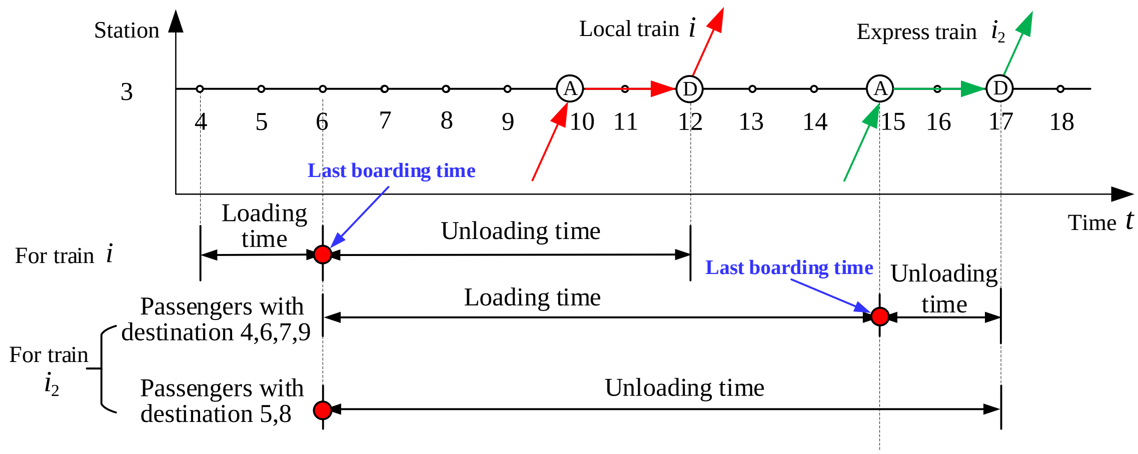


Figure 3 Illustration of an effective loading period

Therefore, the key to solving this timetabling problem under the ELM is to accurately calculate A^t_{ikl} for each arriving passenger at each station.

2.4. Assumptions

We make the following assumptions throughout this study.

A.1) The passenger OD demand has a temporal time unit granularity of δ, and the stop plan of express trains is predetermined.

A.2) Passengers take only one train to their destination; in other words, transfers are not allowed, and passengers must board the first train to their destination if the arriving train capacity is sufficient.

A.3) All the passengers boarding a train obey the principle of first-in-first-out (FIFO), and they must wait for the next train when the current train reaches its capacity.

A.4) An express train can only overtake a local train at certain stations that have overtaking tracks, and express trains are not allowed to stop at the overtaking stations.

A.5) The acceleration and deceleration phases of train operations are considered to be predetermined constants.

A.6) The total train capacity is always larger than or equal to the total passenger demand, and the last train is a local train that departs from the first station at the exact end time T_1 of the studied time horizon.

A.7) Once passengers enter the SRT, they do not change their mode of travel.

In practice, an increasing number of lines apply the ELM. Hence, assumption A.1 is reasonable. In train operations with the ELM, passenger behavior is more complex than that with the TSM. For example, a passenger may let the current local train pass and take the next express train, or a passenger at a station who has been skipped by the express trains may take a local train to the next station, where express trains stop, and board an express train. To facilitate the problem formulation, we assume in A.2 that all the passengers wait at their origin station for a train traveling directly to their destination. In general, passengers arriving earlier have a better chance to board an arriving train. Without the FIFO principle, it is difficult to determine the exact boarding order of the passengers. Therefore, assumption A.3 is reasonable in practice. Inside a station, an express train may first stop at an overtaking station and then overtake a slow train. However, this situation is often not practical in Chinese suburban transit lines, because an additional track with a platform is required, which may significantly increase the demand for space and the cost of the rail infrastructure. For this reason, we adopt assumption A.4, which stipulates that an express train is not allowed to stop at overtaking stations. This reflects the current infrastructure layout at most Chinese transit lines. The main difference between the ELM and the TSM is the number of stops of ELTs, which affects their running times. The acceleration/deceleration time associated with each train stop is viewed as a part of the running time between the stations. Assumption A.6 ensures that passengers arrive at their destination stations during the predefined time horizon (Niu et al., 2015). The situation where passengers switch to other transportation modes (if they cannot board a train on time) may increase the complexity of the problem. Hence, we assume in A.7 that passengers are not allowed to abandon the railway system, even though the proposed solution can give rise to long headways. Additionally, this study focuses on unidirectional railway lines. Train timetabling that considers bidirectional corridor operation is more practical, albeit more complex—particularly when combined with special stopping modes. Moreover, the existing research on timetabling combined with special stopping modes is often based on unidirectional operation of railway lines (Wang et al., 2014; Niu et al., 2015; Zhang et., 2016; Zhang et al., 2016; Li et al., 2019).

3. Mathematical formulations

We now focus on the timetabling problem under the TSM and ELM. We present four mathematical programming models.

- Model A: trains operate under the TSM with unlimited capacities.
- Model B: trains operate under the TSM with limited capacities.
- Model C: trains operate under the ELM with limited capacities, but overtaking is not allowed.
- Model D: trains operate under the ELM with limited capacities, and overtaking is allowed.

Model A considers the ideal situation [inspired by Barrena et al. (2014)], in which each train has an unlimited capacity and can take on all waiting passengers at each platform. Model B is more realistic because each train has a limited capacity, and passengers cannot always board the first arriving train. Examples of the train timetables established by Models C and D are shown in Figures 19(b) and(c), where Model C can be directly applied to lines without additional conditions, whereas Model D can only be applied with the provision of overtaking tracks. Models C and D can accommodate train capacity constraints.

3.1. Model A

Model A serves as the basis for the other three models. Models B, C, and D are generalizations of Model A and are derived by considering more practical constraints. We present the formulations of these four models step-by-step to provide a comprehensive understanding of the complexity introduced by the new constraints in Models B, C, and D. Moreover, Models A and B are used to perform a direct comparison with an optimum or a good lower bound to the optimum, for assessing the performance of our solution methods.

3.1.1. Parameters

Table 2 presents all the relevant subscripts and parameters used in Model A. This notation is also used for Models B, C, and D, and we introduce new parameters as needed in the subsequent sections.

Table 2 Subscripts and parameters used in Model A

Notations	Definition
S	Set of stations, indexed by $S = \{1, 2, \cdots, s\}$
Φ	Set of trains, indexed by $\Phi = \{1, 2, \cdots, N\}$, where N represents the number of trains within the studied time horizon
k, l	Station indices; $k, l \in S$
i, j	Train indices; $i, j \in \Phi$
δ	Length of the time interval, e.g., 30 s or 1 min
γ_k^0	Start of the studied time horizon for station k; passengers arriving earlier than γ_k^0 will not be served
T_k	End of the studied time horizon for station k; passengers arriving later than T_k will not be served
Γ_k	Set of time intervals within the studied time horizon at station k; $\Gamma_k = \{\varsigma, \varsigma+1, ..., \xi\}$, $\varsigma = \gamma_k^0/\delta$, $\xi = T_k/\delta$
t	Time index; $t \in \{0, 1, ...T_s/\delta\}$
P_{kl}^t	Number of passengers traveling from station k to l during the time interval $[(t-1), t]$
w_{ik}	Dwell time of train i at station k
ρ_{ik}	Free-flow running time of train i between stations k and $k+1$
t_{ik}^r	Running time of train i between stations k and $k+1$
ϖ_a	Additional acceleration time associated with a train stop
ϖ_d	Additional deceleration time associated with a train stop
I_{dd}	Minimum time interval between the departures of consecutive trains at each station
I_{aa}	Minimum time interval between the arrivals of consecutive trains at each station
I_{da}	Minimum time interval between the departure and arrival of consecutive trains at each station

3.1.2. Decision variables

g_{ik}^t = 1 if train i departs from station k at time t, and it is equal to 0 otherwise

f_k^t Number of passengers waiting at station k at the end of time period $t-1$ that must wait for the full length of a time interval (i.e.,δ) until time t

U_k^t Number of boarding passengers departing from station k at time t

3.1.3. Formulation

For the TSM, regardless of which train passengers choose to take, the in-vehicle times are the same. Therefore, the objective function is to compute the total passenger waiting time. We consider that each passenger arriving during the time interval $[(t-1), t]$ waits half of this time interval on average, that is, $\delta/2$, plus the full length of each time interval, that is, δ, during the remaining time intervals until boarding a suitable train for his/her destination. Therefore, the total passenger waiting time is $\frac{\delta}{2}\sum_{k=1}^{s-1}\sum_{l=k+1}^{s}\sum_{t\in\Gamma_k} P_{kl}^t + \delta\sum_{k=1}^{s-1}\sum_{t\in\Gamma_k} f_k^t$. Because the first term is a constant, it can be ignored in calculating the objective function.

Thus, the problem can be formulated as follows:

$$min \quad \delta\sum_{k=1}^{s-1}\sum_{t\in\Gamma_k} f_k^t \tag{1}$$

Subject to:

$$f_k^t = \sum_{\tau=1}^{t-1}\left(\sum_{l=k+1}^{s} P_{kl}^\tau - U_k^\tau\right),\ \forall k\in S/\{s\}, t\in\Gamma_k/\{1\}, f_k^1 = 0 \tag{2}$$

$$U_k^t \le \sum_{i=1}^{N} g_{ik}^t \sum_{\tau=1}^{t}\sum_{l=k+1}^{s} P_{kl}^\tau,\ \forall k\in S/\{s\}, t\in\Gamma_k/\{1\} \tag{3}$$

$$\sum_{t\in\Gamma_k} U_k^t = \sum_{t\in\Gamma_k}\sum_{l=k+1}^{s} P_{kl}^t,\ \forall k\in S/\{s\} \tag{4}$$

$$\sum_{t\in\Gamma_k} g_{ik}^t = 1,\ \forall i\in\Phi, k\in S/\{s\} \tag{5}$$

$$g_{ik}^t \le \sum_{\tau=1}^{t} g_{i-1k}^\tau,\ \forall i\in\Phi/\{1\}, k\in S/\{s\}, t\in\Gamma_k \tag{6}$$

$$t_{ik}^r = \rho_{ik} + \varpi_a + \varpi_d,\ \forall i\in\Phi, k\in S/\{s\} \tag{7}$$

$$\sum_{t\in\Gamma_{k+1}} tg_{ik+1}^t = \sum_{t\in\Gamma_k} tg_{ik}^t + t_{ik}^r + w_{ik+1},\ \forall i\in\Phi, k\in S/\{s-1, s\} \tag{8}$$

$$\sum_{t\in\Gamma_k} tg_{i+1k}^t - \sum_{t\in\Gamma_k} tg_{ik}^t \ge I_{dd},\ \forall i\in\Phi/\{N\}, k\in S/\{s\} \tag{9}$$

$$[\sum_{t\in\Gamma_k} tg_{i+1k}^t - w_{i+1k}] - [\sum_{t\in\Gamma_k} tg_{ik}^t - w_{ik}] \ge I_{aa},\ \forall i\in\Phi/\{N\}, k\in S/\{1\} \tag{10}$$

$$[\sum_{t\in\Gamma_k} tg_{i+1k}^t - w_{i+1k}] - \sum_{t\in\Gamma_k} tg_{ik}^t \geq I_{da},\ \forall i \in \Phi/\{N\}, k \in S/\{1,s\} \tag{11}$$

Objective function (1) minimizes the total passenger waiting time. Constraints (2) define the number of passengers waiting at station k at the end of time period $t-1$ who must wait for a full-time interval until time t. The right-side of the formula defines the number of passengers who arrive at station k before time $t-1$ minus the number who board the train leaving station k until time $t-1$. Constraints (3) define the upper bound of the number of boarding passengers for each train departing from station k at time t. If no train departs at time t, the number of leaving passengers is zero. Constraints (4) ensure that at each station, the total number of passengers departing is equal to the number of passengers arriving, i.e., all the arriving passengers during the studied time horizon must be served. Constraints (5) ensure that all trains at each station depart during the studied time horizon and depart no more than once. Constraints (6) order the trains by their indices, i.e., train i cannot depart before train $i-1$. Constraints (7) model the running time of each train in each section, which is defined as the free-flow running time, in addition to the additional acceleration and deceleration times associated with the train stops. Constraints (8) refer to the relationship between the departure times of each train at adjacent stations, i.e., that the departure time from station $k+1$ is determined by the departure time from the previous station k plus the running time at the previous section k and the dwelling time at station $k+1$. Constraints (9)–(11) are related to the headway between consecutive trains at a station, as shown in Figure 21(a).

3.2. Model B

3.2.1. Additional parameter

C	Train capacity

3.2.2. Additional decision variables

u_{kl}^t	Number of passengers boarding the train leaving station k toward station l at time t
K_{ikl}	Total number of boarding passengers at station k whose destination is station l as train i departs; that is, all the passengers who have been transported from station k with station l as the destination at the departure time of train i
X_{ik}	Total number of alighting passengers at station k as train i departs
V_{ik}	Total number of boarding passengers at station k as train i departs
E_{ik}	Number of alighting passengers at station k from train i
B_{ik}	Number of passengers at station k boarding train i
Z_{ik}	Number of passengers aboard train i departing from station k

3.2.3. Formulation

The difference between Models A and B is that in Model B, the exact number of passengers on each train (and thus the available train capacity) is known. In addition to the variables of Model A, Model B introduces an integer variable u_{kl}^t, which represents the number of passengers boarding the train leaving station k with station l as the destination at time t. We introduce variables B_{ik} and E_{ik} to represent the numbers of passengers boarding and alighting from train i at station k, respectively, which are used to calculate the number of passengers on train i. Similar to Model A, the objective of Model B is to minimize the total passenger waiting time, as indicated by Equation (1).

Thus, the constraints include (2) and (5)–(11), in addition to the following:

$$U_k^t = \sum_{l=k+1}^{s} u_{kl}^t,\ \forall k \in S/\{s\}, t \in \Gamma_k \tag{12}$$

$$u_{kl}^t \leq \sum_{i=1}^{N} g_{ik}^t \sum_{\tau=1}^{t} P_{kl}^t,\ \forall 1 \leq k < l \leq s, t \in \Gamma_k \tag{13}$$

$$\sum_{t \in \Gamma_k} \sum_{l=k+1}^{s} u_{kl}^t = \sum_{t \in \Gamma_k} \sum_{l=k+1}^{s} P_{kl}^t,\ \forall k \in S/\{s\} \tag{14}$$

$$\sum_{t \in \Gamma_k} \sum_{k=1}^{l-1} u_{kl}^t = \sum_{t \in \Gamma_k} \sum_{k=1}^{l-1} P_{kl}^t,\ \forall l \in S/\{1\} \tag{15}$$

$$K_{ikl} = \sum_{t \in \Gamma_k} [g_{ik}^t \sum_{\tau=1}^{t} u_{kl}^\tau],\ \forall i \in \Phi, 1 \leq k < l \leq s \tag{16}$$

$$X_{ik} = \sum_{k'=1}^{k-1} K_{ik'k},\ \forall i \in \Phi, k \in S/\{1\} \tag{17}$$

$$V_{ik} = \sum_{l=k+1}^{s} K_{ikl},\ \forall i \in \Phi, k \in S/\{s\} \tag{18}$$

$$E_{ik} = \begin{cases} X_{ik} & i = 1 \\ X_{ik} - X_{i-1k} & i \in \Phi/\{1\} \end{cases},\ \forall k \in S/\{1\} \tag{19}$$

$$B_{ik} = \begin{cases} V_{ik} & i = 1 \\ V_{ik} - V_{i-1k} & i \in \Phi/\{1\} \end{cases},\ \forall k \in S/\{s\} \tag{20}$$

$$Z_{ik} = \begin{cases} B_{ik} & k = 1 \\ \sum_{k'=1}^{k} B_{ik'} - \sum_{k'=2}^{k} E_{ik'} & k \in S/\{1, s\} \end{cases},\ \forall i \in \Phi \tag{21}$$

$$Z_{ik} \leq C,\ \forall i \in \Phi, k \in S/\{s\} \tag{22}$$

Constraints (12) ensure that the number of boarding passengers at time t at station k equals the total number of passengers boarding at time t at station k with station l as the destination. Constraints (13) define an upper bound on the number of boarding passengers at time t leaving station k with station l as the destination. Constraints (14) and (15) ensure that the total number of boarding passengers is equal to the number of arriving passengers, i.e., all passengers arriving during the studied time horizon are served. Constraints (16) define the total number of boarding passengers leaving station k with station l as the destination when train i departs. Constraints (17) and (18) define the total numbers of alighting and boarding passengers, respectively, at station k when train i departs. Constraints (19) model the number of passengers alighting from train i at station k, and constraints (20) model the number of passengers boarding train i at station k. Constraints (21) identify the number of passengers aboard train i when it departs from station k. Constraints (22) ensure that the number of onboard passengers does not exceed the corresponding train capacity.

3.3. Model C

3.3.1. Parameters

Table 3 presents the relevant parameters used in Model C.

Table 3 Parameters used in Model C

Notations	Definition
e_k	A value of 0 or 1 to indicate whether an express train skips station k; 1 indicates that it does not (that is, the express train stops at station k when e_k is equal to 1)
h_{ik}	A value of 0 or 1 to indicate whether train i stops at station k; 1 indicates that it does
α_w	Perception coefficient of waiting time for passengers
α_v	Perception coefficient of in-vehicle time for passengers
I_{ds}	Minimum time interval between the departure–skipping times of consecutive trains at non-overtaking stations
I_{sa}	Minimum time interval between the skipping–arrival times of consecutive trains at non-overtaking stations
I_{ss}	Minimum time interval between the skipping–skipping times of consecutive trains at non-overtaking stations
b_{ikl}	Number of passengers boarding train i from station k with station l as the destination
M	A very large positive integer
W_{ikl}	Waiting time for train i at station k for passengers with station l as the destination
V_{ikl}	In-vehicle time for passengers in train i traveling from station k to l

3.3.2. Decision variables

In the ELM, Model C focuses on generating an optimal train schedule for ELTs simultaneously, where it is necessary to specify the following operational characteristics for each train at every station.

x_i: A value of 0 or 1 indicating whether train i is an express train; 1 indicates an express train

t^d_{ik}: Departure time of train i from station k

t^a_{ik}: Arrival time of train i at station k

A^t_{ikl}: Loading indicator to suggest whether passengers who arrive at station k with station l as the destination at time t choose to board train i; 1 indicates that they do, and 0 indicates that they do not

R_{kl}^{t}: Actual boarding time for a passenger who arrives at station k with station l as the destination at time t

3.3.3. Formulation

The same train stopping plan and running speed in the TSM lead to the same in-vehicle times for the passengers. Hence, the objective of Models A and B is to minimize the passenger waiting times at the stations. However, the ELM may introduce different in-vehicle times when passengers take different types of trains to reach the same destination station, as shown in Figure 23. Hence, Model C aims to minimize the total passenger waiting times at the stations and the total in-vehicle times on the trains.

To establish Model C, the following two groups of constraints are generated.

(1) Train running constraints

$$h_{ik} = (1 - x_i) + x_i e_k,\ \forall i \in \Phi, k \in S \tag{23}$$

$$t_{ik}^{r} = \rho_{ik} + h_{ik}\varpi_a + h_{ik+1}\varpi_d,\ \forall i \in \Phi, k \in S/\{s\} \tag{24}$$

$$t_{ik}^{d} = t_{ik}^{a} + h_{ik}w_{ik},\ \forall i \in \Phi, k \in S/\{s\} \tag{25}$$

$$t_{ik}^{a} = t_{ik-1}^{d} + t_{ik-1}^{r},\ \forall i \in \Phi, k \in S/\{1\} \tag{26}$$

$$t_{ik}^{d} - t_{i-1k}^{d} \geq h_{i-1k}h_{ik}I_{dd},\ \forall i \in \Phi/\{1\}, k \in S/\{s\} \tag{27}$$

$$t_{ik}^{a} - t_{i-1k}^{a} \geq h_{i-1k}h_{ik}I_{aa},\ \forall i \in \Phi/\{1\}, k \in S/\{1\} \tag{28}$$

$$\begin{aligned} t_{ik}^{a} - t_{i-1k}^{d} \geq\ & h_{i-1k}h_{ik}I_{da} + h_{i-1k}(1-h_{ik})h_{ds} + (1-h_{i-1k})h_{ik}I_{sa} \\ & + (1-h_{i-1k})(1-h_{ik})I_{ss}, \forall i \in \Phi\{1\}, k \in S/\{1, s\} \end{aligned} \tag{29}$$

$$x_N = 0 \tag{30}$$

$$t_{N1}^{d} = T_1 \tag{31}$$

Constraints (23) model the stop scheme of train i at station k, which is related to the train type. Constraints (24) ensure that the train running time is computed according to the free-flow running time, as well as the additional acceleration and deceleration times associated with train stops. Constraints (25) and (26) allow the calculation of a train's arrival and departure times. Constraints (27)–(29) define different types of headway constraints between adjacent trains, as illustrated in Figures 21(b) and 22(a)–(c). Constraints (30) and (31) correspond to Assumption A.6. Constraints (30) ensure that the last train in the studied time horizon is a local train. Constraints (31) indicate that the last train departs from the first station at the end of the studied time horizon.

(2) Passenger loading constraints

As shown in Figure 3, the constraints related to the passenger loading time periods are as follows:

$$\sum_{i=1}^{N} A_{ikl}^{t} = 1,\ \forall 1 \leq k < l \leq s, t \in \Gamma_k \tag{32}$$

$$2A_{ikl}^{t} \leq x_{ik} + x_{il},\ \forall i \in \Phi, 1 \leq k < l \leq s, t \in \Gamma_k \tag{33}$$

$$t_{ik}^{d} \geq t - M(1 - A_{ikl}^{t}),\ \forall i \in \Phi, 1 \leq k < l \leq s, t \in \Gamma_k \tag{34}$$

$$R_{kl}^{t} \geq t,\ \forall 1 \leq k < l \leq s, t \in \Gamma_k \tag{35}$$

$$R_{kl}^{t+1} \geq R_{kl}^{t},\ \forall 1 \leq k < l \leq s, t \in \Gamma_k \tag{36}$$

$$t_{ik}^{a} \leq R_{kl}^{t} + M(1 - A_{ikl}^{t}),\ \forall i \in \Phi, 1 \leq k < l \leq s, t \in \Gamma_k \tag{37}$$

$$t_{ik}^{d} \geq R_{kl}^{t} - M(1 - A_{ikl}^{t}),\ \forall i \in \Phi, 1 \leq k < l \leq s, t \in \Gamma_k \tag{38}$$

Constraints (32) ensure that each passenger chooses one train to board and that all passengers are transported to their destination during the studied time horizon; that is, no passenger is stranded when the last train departs. Constraints (33) ensure that A_{ikl}^{t} is equal to 0 if train i does not stop at station k or station l, which corresponds to Assumption A.2. Constraints (34) indicate that if the passengers arriving at station k at time t can take train i to destination station l, their arrival time t must be earlier than the departure time of train i at station k. Constraints (35) ensure that a passenger's boarding time on each train is later than his/her arrival time at the station. Constraints (36) ensure that the passengers satisfy the FIFO service rule. Constraints (37) and (38) ensure that if the passengers arriving at station k at time t choose to take train i to destination station l, their boarding time is earlier than the departure time of train i and later than the arrival time of train i.

According to A_{ikl}^{t}, we arrive at the following constraints:

$$b_{ikl} = \sum_{t \in \Gamma_k} (A_{ikl}^{t} P_{kl}^{t}),\ \forall i \in \Phi, 1 \leq k < l \leq s \tag{39}$$

$$E_{ik} = \sum_{k'=1}^{k-1} b_{ik'k},\ \forall i \in \Phi, k \in S/\{1\} \tag{40}$$

$$B_{ik} = \sum_{l=k+1}^{s} b_{ikl},\ \forall i \in \Phi, k \in S/\{s\} \tag{41}$$

Constraints (39) determine the exact number of passengers boarding train i at station k with station l as the destination. Constraints (40) and (41) determine the number of alighting and boarding passengers, respectively, at station k for train i. The onboard passengers when train i leaves station k and train capacity constraints are modeled by Constraints (21) and (22).

According to the foregoing considerations, the calculation of the PTT (denoted as T_P) is defined by the following constraints. The formulation is subject to Constraints (21)–(41). The resulting objective function is nonlinear; hence, Model C is an MINLP formulation.

$$min \quad T_P = \delta \sum_{i=1}^{N} \sum_{k=1}^{s-1} \sum_{l=k+1}^{s} (\alpha_w W_{ikl} + \alpha_v V_{ikl}) \tag{42}$$

$$W_{ikl} = \sum_{t \in \Gamma_k} [A_{ikl}^t P_{kl}^t (t_{ik}^d - t)], \ \forall i \in \Phi, 1 \le k \le l \le s \tag{43}$$

$$V_{ikl} = b_{ikl}(t_{il}^a - t_{ik}^d), \ \forall i \in \Phi, 1 \le k \le l \le s \tag{44}$$

3.4. Model D

3.4.1. Additional parameters

Sets:

Δ Set of stations with overtaking tracks; $\Delta \in S$

Parameters:

I_{ao} Minimum time interval for arrival–overtaking of consecutive trains at an overtaking station

I_{od} Minimum time interval for overtaking–departure of consecutive trains at an overtaking station

M_d Maximum dwell time for each train

3.4.2. Additional decision variables

y_{ijk} A value of 0 or 1 to indicate the departure order for trains i and j from station k; 1 if train i departs from station k before train j and 0 otherwise

O_{ijk} A value of 0 or 1 to indicate whether train i is overtaken by train j at station k; 1 if train i is overtaken and 0 otherwise

Q_{ik} Dwell time of train i at station k

3.4.3. Formulation

The main difference between Models C and D is whether overtaking between trains is allowed in the model, which means that the train that has departed just before train i is not necessarily train $i-1$. For example, if train i overtakes train $i-1$ at station k, train i is the previous departing train for train $i-1$ from station k. Therefore, the constraints in Model D are Constraints (21)–(24), (26), and (30)–(44), as well as the following additional constraints:

$$y_{ijk} + y_{jik} = 1, \ \forall i, j \in \Phi, i \ne j, k \in S/\{s\} \tag{45}$$

$$O_{ijk} - O_{jik} = y_{ijk-1} - y_{ijk}, \ \forall i, j \in \Phi, i \ne j, k \in S/\{1, s\} \tag{46}$$

$$O_{ijk} + O_{jik} \le 1, \ \forall i, j \in \Phi, i \ne j, k \in S \tag{47}$$

$$\sum_{k=1}^{s} \sum_{i} O_{ijk} \le M x_j, \ \forall i, j \in \Phi, i \ne j \tag{48}$$

$$\sum_{k=1}^{s}\sum_{j} O_{ijk} \leq M(1-x_i),\ \forall i,j \in \Phi, i \neq j \tag{49}$$

$$t_{ik}^{d} = t_{ik}^{a} + Q_{ik},\ \forall i \in \Phi, k \in S/\{s\} \tag{50}$$

$$Q_{ik} = (1-\sum_{j} O_{ijk})h_{ik}w_{ik} + (\sum_{j} O_{ijk})h_{ik}Q_{ik},\ \forall i \in \Phi, k \in S/\{s\} \tag{51}$$

$$Q_{ik} \leq M_d,\ \forall i \in \Phi, s \in S/\{s\} \tag{52}$$

$$t_{ik}^{d} - t_{jk}^{d} \geq O_{ijk}I_{od} + (1-O_{ijk})h_{ik}h_{jk}I_{dd} - My_{ijk}, \forall i,j \in \Phi, i \neq j, k \in S/\{s\} \tag{53}$$

$$t_{jk}^{a} - t_{ik}^{a} \geq O_{ijk}I_{ao} + (1-O_{ijk})h_{ik}h_{jk}I_{aa} - M(1-y_{ijk-1}), \forall i,j \in \Phi, i \neq j, k \in S/\{1\} \tag{54}$$

$$\begin{aligned} t_{jk}^{a} - t_{ik}^{d} \geq{}& (1-O_{ijk})\left\{h_{ik}h_{jk}I_{da} + h_{ik}(1-h_{jk})I_{ds} + (1-h_{ik})h_{jk}I_{sa} + (1-h_{ik})(1-h_{jk})I_{ss}\right\} \\ & -M[1-(1-O_{ijk})y_{ijk-1}], \forall i,j \in \Phi, i \neq j, k \in S/\{1,s\} \end{aligned} \tag{55}$$

Constraints (45) define the departure order of trains i and j at station k. If train i departs before train j, y_{ijk} is equal to 1 and 0; otherwise, it is equal to 0. Constraints (46) define whether train i is overtaken by train j at station k; that is, $O_{ijk} = 1$ when $y_{ijk-1} - y_{ijk} = 1$ (i.e., train i departs from station $k-1$ before train j, while train i departs after train j at station k), and $O_{jik} = 1$ when $y_{ijk-1} - y_{ijk} = -1$ (i.e., train j is overtaken by train i). Constraints (47) forbid each pair of trains to be mutually overtaken by the other in the pair. Constraints (48) and (49) are defined according to Assumption A.4. The former constraint indicates that no overtaking occurs when train j is a local train; that is, a local train cannot overtake other trains. The latter constraint indicates that train i cannot be overtaken by other trains if it is an express train. Constraints (50)–(52) define the train's dwell time at the station, which is not longer than a given maximum dwell time, to prevent it from becoming too long, when a local train is overtaken by an express train. Constraints (53)–(55) define different types of headway constraints between consecutive trains. When O_{ijk}=1, we can obtain $y_{ijk-1} = 1$ and $y_{ijk} = 0$; then, Constraints (53) and (54) indicate the trains' overtaking–departure and arrival–overtaking intervals at the overtaking stations (as shown in Figure 22(d)). When O_{ijk}=0, Constraints (53) and (54) indicate the trains' departure–departure and arrival–arrival intervals at non-overtaking stations (as shown in Figure 21(b)). Constraints (55) indicate four different train tracking intervals at non-overtaking stations that are related to whether the front and rear trains stop (i.e., the trains' departure–arrival, departure–skipping, skipping–arrival, and skipping–skipping intervals, as shown in Figures 21(b) and 22(a)–(c)).

4. Optimization algorithm

The studied train timetabling problem was proven to be NP-hard (Cai and Goh, 1998; Caprara et al., 2002); hence, it is impractical to use commercial solvers, such as CPLEX, with large-scale instances. The formulation of the train timetabling problem under the ELM becomes nonlinear, which is beyond the ability of CPLEX when dealing with real-world instances. To this end, we developed an A-GA to efficiently handle all the four variants presented earlier.

Genetic algorithms (GAs) simulate the natural evolution of a population and its genetics to generate better solutions in successive generations (Holland, 1975). Initially, a set of potential solutions is encoded as chromosomes that comprise the genes. A fitness function evaluates the performance of the solutions. The *selection* process selects individuals in proportion to their fitness values (Goldberg, 1989). An "Elitist" replacement method is used, where the best chromosome replaces the worst chromosome (Souai and Teghem, 2009). *Crossover* is implemented by combining two chromosomes (parents) to produce a new chromosome (offspring), and *mutation* is achieved by changing one or more gene values of a chromosome. New solutions with better fitness values will replace their parents. Selection, crossover, and mutation are repeated until the number of new chromosomes equals the population size. This completes an iteration (generation). The iterations continue until they reach a given maximum number.

The traditional GA suffers from the difficulty of properly setting the genetic operators (e.g., crossover and mutation parameters), which if not done well can significantly impact the performance. Some researchers have attempted to solve this problem. Others have combined the GA with local search algorithms (Nachtigall and Voget, 1996) or focused on parameter tuning and running the GA using fixed values (De Jong, 1975; Grefenstette, 1986). However, these efforts usually involve trial-and-error methods based on observations, and the use of unchanging parameter values renders the algorithm inefficient (Yun and Gen 2003). Another direction is parameter control, i.e., adaptively adjusting the parameters according to the problem specifications. (Srinivas and Patnaik, 1994; Yang et al., 2015).

Our A-GA employs an adaptive procedure to set the crossover and mutation operators. A lower crossover/mutation probability is set for a chromosome whose fitness value is higher than the average, to prevent individuals from being destroyed, by giving them a higher probability to pass through crossover and mutation. A higher crossover/mutation probability is set for individuals whose fitness value is lower than the average, to possibly produce better individuals through crossover and mutation. Therefore, the adaptive crossover/mutation operator can calculate the most suitable probability according to the fitness function of each individual at each generation.

Furthermore, in our A-GA, an initial solution may be infeasible when different types of trains operate in the ELM. For example, an express train may overtake a local train outside the station, which is not practical in reality. Additionally, the headway time constraint may be violated. Therefore, a "gene feasibility repair strategy" is necessary to obtain a feasible solution. Below, we describe different elements of the A-GA and elaborate on the process of solving the timetabling problem under the ELM with overtaking (Model D, which is a comprehensive model that considers all the factors, including the train capacity and train overtaking). We introduce the chromosome representation in Section 4.1, and the gene feasibility repair strategy is presented in Section 4.2. In section 4.3, we define the fitness function. We explain the A-GA operation in Section 4.4 and the stop criterion in Section 4.5.

4.1. Chromosome representation

Each solution is represented by a chromosome, which represents the train departure and arrival times at each station and its type/sequence. Each chromosome is built based on two steps. Given $1, 2, \cdots, N$ trains, the first step consists of randomly generating the departure time of each train from its departure station and its train type/running sequence (as shown in Figure 4). The second step is to obtain the arrival and departure times for each train at each station via Equations (26), (50) and (51), as shown in Figure 5. It should be noted that once the train departure time at its departure station changes, the arrival and departure times at each of the following stations change accordingly. Therefore, we call the chromosome generated in the first step the *"basic chromosome"*, on which subsequent crossover and mutation operations are carried out.

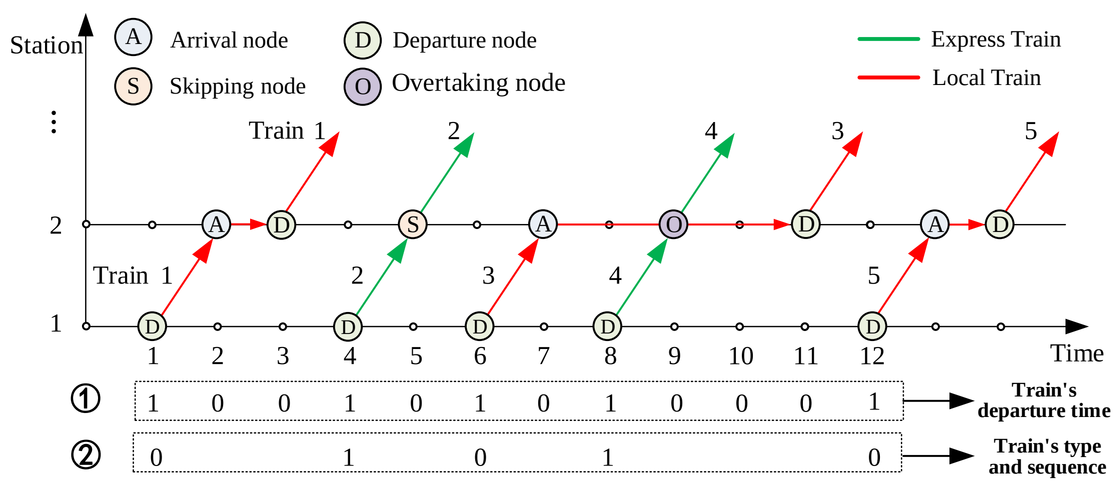


Figure 4 Illustration of solution encoding with the ELM

Each *basic chromosome* consists of $(T_1/\delta + N)$ genes, and a binary encoding method is used in this study. The first (T_1/δ) genes denote the departure time of each train, where 1 indicates a train departure, and the next N genes represent the running sequence of the trains; here, 1 and 0 indicate an express train and a local train, respectively. Counters with frequencies of 1 and 0 correspond to the numbers of ELTs, respectively. The departure time should satisfy the headway constraints between successive trains, where the first train is randomly generated from $[0, T_1 - (N-2)I_{dd}]$, and the other trains are randomly generated $(i = 2, \cdots, N-1)$ from $[t^d_{i-1,1} + I_{dd}, T_1 - [N-(i+1)]I_{dd}]$. Figure 4 illustrates a chromosome 100101010001—01010 with $(T_1/\delta = 12)$ and $(N = 5)$, where the first 12 genes represent trains departing from the departure station at times 1, 4, 6, 8, and 12, while the last five genes indicate that the running sequence is local → express → local → express → local. Furthermore, we obtain the values of each train type/sequence and the departure and arrival times at each station, which are stored as a vector and two cellulars, respectively.

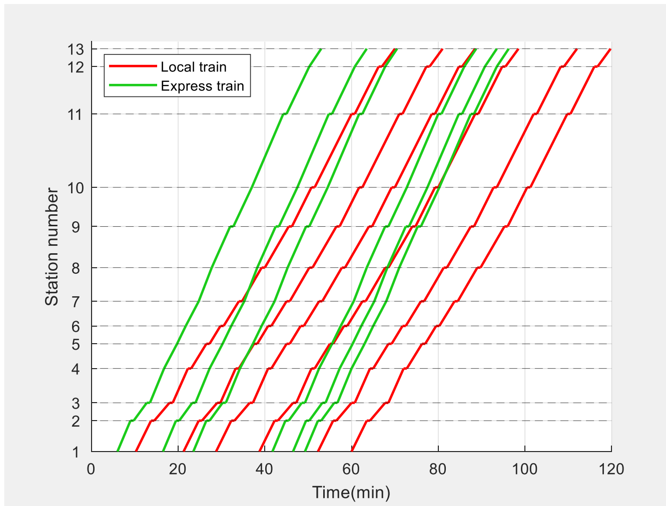


Figure 5 Generation of initial population

4.2. Gene feasibility repair strategy

The initial individual may be infeasible for the timetable under the ELM because 1) an express train may overtake a local train in an interval between the stations, which is not practical in reality, and 2) the headway time constraints may be violated. To fix these infeasibilities, we designed a gene feasibility repair strategy consisting of the following three steps.

4.2.1. First step: determine the overtaking station

Assumption A.4 states that an express train can only overtake a local train at a station. If an express train overtakes a local train, we must identify a station where this can occur.

To this end, if the overtaking occurs in an interval between stations k and $k+1$, the local train should wait at station k', i.e., the closest station before station $k+1$, where overtaking can occur (with additional tracks), until the express train departs from station k'. As shown in Figure 6(a), overtaking should occur at station $k-1$, because station k, where the express train stops, does not allow it. In this case, the dwell time of the local train being overtaken may exceed the maximum allowed [as shown in Figure 6(b)]. Thus, we adjust the departure time of the express train from the departure station [as shown in Figure 6(c)] such that the dwell time of the local train does not exceed the maximum and the arrival–overtaking headway requirements are satisfied [as shown in Figure 6(d)].

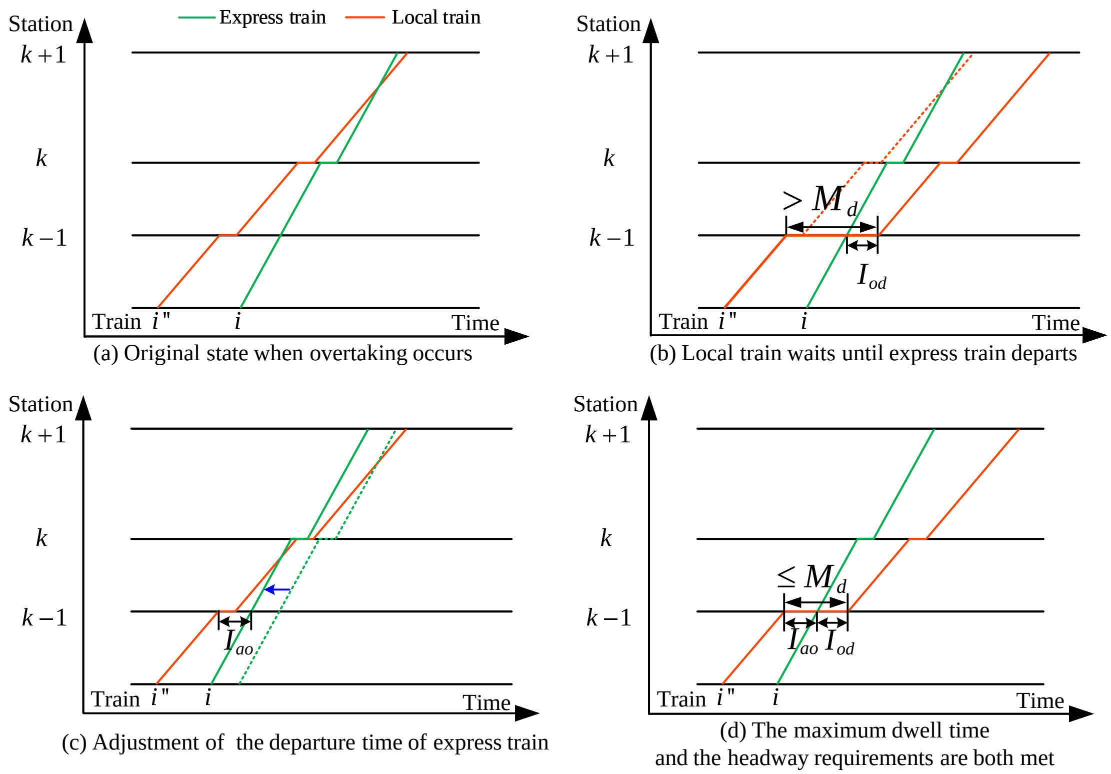


Figure 6 Adjusting trains when overtaking occurs

At the end of this procedure, we obtain an adjusted timetable for which all overtaking stations are found, and the maximum dwell time, arrival-overtaking, and overtaking-departure headway requirements are satisfied, as shown in Figure 7.

4.2.2. Second step: adjust the headway time between consecutive trains

Let us consider an infeasibility, in which the headway constraints between two adjacent trains are violated. We take Figure 7 as an example, in which the third local train and fourth express train cannot satisfy the arrival–arrival headway constraints at the terminal station (marked by a blue circle). We must adjust the trains to satisfy the headway constraints. In the case of an infeasibility, if the headway constraints between adjacent trains i and j are violated at station k, we first adjust the departure time of train j. There are two possible cases. 1) If train j is an express train, we recognize the set of local trains ξ_1 that have been overtaken by train j (i.e., the trains departing before train j at the departure station and arriving after train j at the last station). Then, we find the set of express trains ϑ_1 that have overtaken the local train belonging to ξ_1. 2) If train j is a local train, we recognize the set of express trains ϑ_2 that overtake train j. Then, we find the set of local trains ξ_2 that have been overtaken by express trains belonging to ϑ_2. Next, we adjust all the corresponding trains belonging to the recognized sets ξ_1, ξ_2, ϑ_1, and ϑ_2.

Because express trains generally do not increase the dwell time in practice, we adjust the departure times of the corresponding express trains from the departure station, while those of the local trains are usually adjusted from the closest overtaking station before station k, to reduce the required adjustment times. This procedure yields the adjusted timetable, as shown in Figure 8.

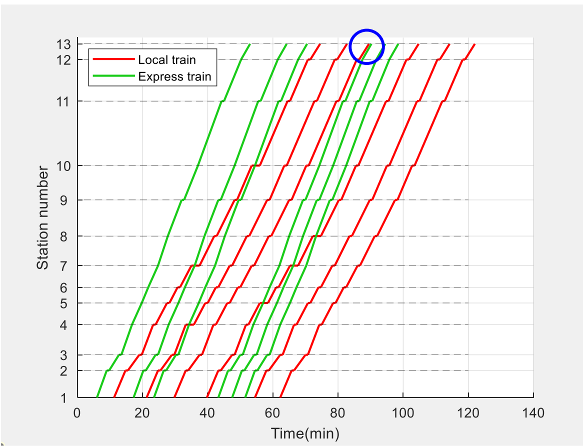


Figure 7 Determininge the overtaking station

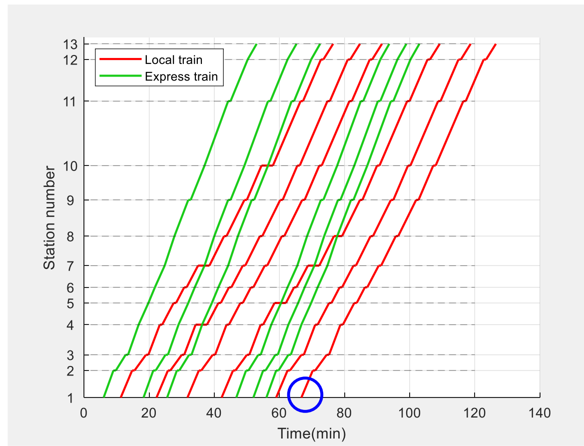


Figure 8 Adjust the headway time between consecutive trains

4.2.3. Constructing initial feasible timetable

Assumption A.6 states that the last train departs from the first station at the exact end time T_1. However, according to the above two steps, the departure time of the last train from the departure station may exceed T_1 (as shown in Figure 8, marked by a blue circle).

We can easily enforce this departure time in some cases by moving all the trains forward (as shown in Figure 9), whereas in other cases, suitable overtaking stations may need to be added to ensure that all the trains depart from the departure station during the studied time horizon (as shown in Figure 10, an overtaking station is added at station 8, which is marked by a blue circle). In this case, the above two steps must be performed repeatedly, until there are no headway time violations and Assumption A.6 is satisfied. At the end of this procedure, we always obtain a feasible solution that satisfies all the running constraints.

The initialization procedure is summarized in Algorithm 1.

4.3. Fitness function

An evaluation function is adopted to assign the probability of reproduction to each chromosome such that its likelihood of being selected is proportional to its fitness (Yang et al., 2015). Therefore, each solution is evaluated by specifying a fitness function and on the basis of these fitness values, genetic operators generate the next individual population.

Algorithm 1 Initialization procedure

1. Set $g = 1$.
2. Randomly generate a chromosome C_g and check its feasibility. If it is not feasible, use the gene feasibility repair strategy.
3. If $g \leq P_s$, set $g = g + 1$ and return to step 2.
4. Return the initialized population $C_g(g = 1, \cdots, P_s)$.

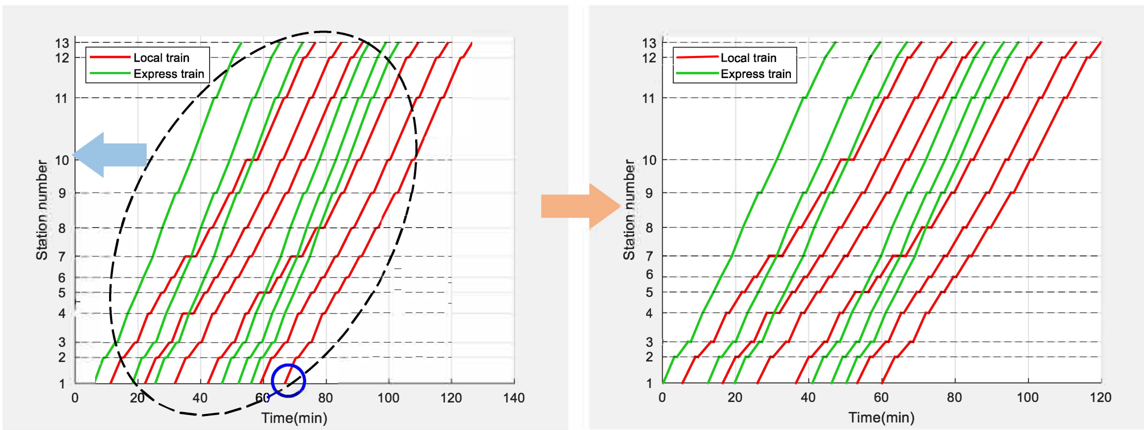


Figure 9 **Constructing the initial feasible timetable**

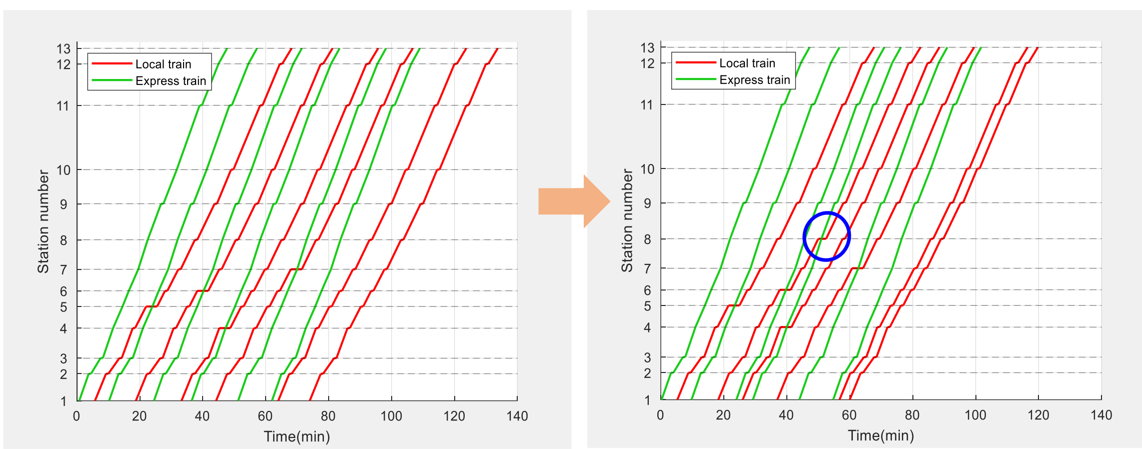


Figure 10 **Adjustments required to compute an initial feasible solution**

The train timetabling problem considered in this study aims to minimize the PTT, and we adopt the objective function of a chromosome to indicate its fitness. We define $fit(\theta) = 1/obj(\theta)$ as the fitness function of the chromosome, where $obj(\theta)$ is the corresponding objective function of the train timetabling models, i.e., Equations (1) and (42).

4.4. GA operations

4.4.1. Selection operation

Solutions are selected according to their fitness function using a “roulette wheel” approach, i.e., a higher fitness value corresponds to a higer likelihood to reproduce. An elitist replacement strategy ensures that the best parent survives to the next iteration (Souai and Teghem, 2009).

The selection procedure is shown in Algorithm 2, which comprises the following eight steps. Steps 1 and 2 involve the computation of the fitness value and the cumulative probability of each individual X_q. In Step 3, the selection probabilities of the individuals are generated. In Steps 4-6, the "roulette selection" approach is employed for each individual. Finally, in Steps 7 and 8, the "elitist replacement" strategy is executed.

Algorithm 2: Selection procedure steps

1. Calculate the fitness value $fit(X_q)$ of each individual X_q and identify the best individual X_b according to the fitness value.
2. Calculate the probability $P(X_q) = fit(X_q)/\sum_{q=1}^{P_s} fit(X_q)$ that an individual can be selected, as well as the cumulative probability of each individual $C(X_q) = \sum_{j=1}^{q} P(X_j)$.
3. Randomly generate P_s numbers between 0 and 1 and put them into a vector R; then, $R(q)$ represents the random selection probability of individual X_q.
4. Set $q = 1$.
5. Roulette selection:
   5.1 If $R(q) < C(X_q)$, individual X_q is selected; proceed to step 6;
   5.2 If $C(X_q) < R(q) < C(X_{q+1})$, individual X_{q+1} is selected; proceed to step 6.
6. If $q \leq P_s$, let $q = q + 1$; return to step 5.
7. Calculate the fitness value of each individual that is going to be selected and identify the worst individual X_w.
8. Use the elitist replacement strategy: replace X_w with X_b.

4.4.2. Crossover Operation

Let P_c represent the initial crossover probability. To increase the diversity of the chromosomes, the crossover operation is conducted separately for the two parts of a basic chromosome: the train departure time and its type/sequence. For each part, we select two parent chromosomes and randomly generate a number $c \in [0, 1]$. If $c \leq P_c$, we use parents p and q to produce offspring chromosomes x and y. If chromosomes x and y are infeasible, we execute the gene feasibility repair strategy.

A two-point crossover approach is adopted in this study, as shown in Figure 4.4.2. For any two chromosomes undergoing a crossover, two crossover points a and b are randomly generated within the length of the selected chromosome. The gene segments between the two swapping points in the two chromosomes are swapped, resulting in two new chromosomes.

The crossover procedure is presented in Algorithm 3, which comprises the following eight steps. Steps 1-3 involve the generation of a random number c indicating whether the crossover operation is conducted. In Step 4, the crossover operation is performed. Step 5 ensures the feasibility of the

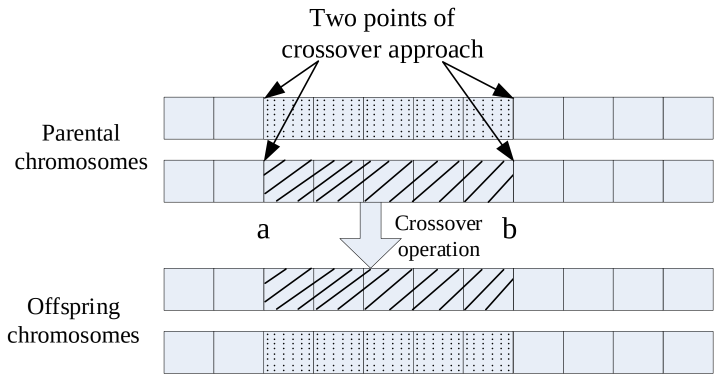


Figure 11 Two-point crossover approach

offspring chromosomes. In Step 6, the fitness value of the offspring chromosomes is calculated, and it is determined whether the parent chromosomes are replaced. Steps 7 and 8 ensure that a new population set with size P_s is obtained.

Algorithm 3: Crossover procedure

1. Initialize a crossover probability P_c and set $g = 1$.
2. Calculate the crossover probability of parent chromosomes g and $g+1$ via Equation (56), and generate a random number c in the interval $[0, 1]$.
3. If $c \leq P_c(g, g+1)$, select the two parent chromosomes g and $g+1$ to produce offspring chromosomes x and y; otherwise, retain g and $g+1$ and proceed to step 7.
4. Crossover for train departure time at the departure station and for train type
   4.1 For each part of each basic chromosome, generate two random crossover points a and b (via the two-point crossover approach, which is explained in Figure 4.4.2) within the intervals $[1, T_1/\delta]$ and $[1, N]$, respectively.
   4.2. Perform crossover for these two parts of the chromosome.
5. Use the gene feasibility repair strategy if x or y is infeasible.
6. Calculate the fitness function of chromosomes x and y. If the fitness of $x(y)$ is better than that of $g(g+1)$, use $x(y)$ to replace the parent; otherwise, retain the parent.
7. If $g \leq P_s$, set $g = g+1$ and return to step 2.
8. Return a new population $C_g(g = 1, \cdots, P_s)$.

4.4.3. Mutation operation

Mutation is a process in which some gene values in a chromosome are replaced by other gene values with a certain mutation probability, generating a new individual. Let P_m represent the initial mutation probability. The mutation operation is performed separately for the two parts of each basic chromosome. We select a parent chromosome q and generate a random number u between 0 and 1. If $u \leq P_m$, we use parent q to produce an offspring chromosome z. If chromosome z is infeasible, we perform a gene feasibility repair strategy.

The mutation procedure is shown in Algorithm 4, which also consists of eight steps. Its main difference from the crossover operation is that each parent chromosome produces one offspring chromosome via a mutation operation, whereas in the crossover operation, each pair of parent chromosomes produces two offspring chromosomes.

Algorithm 4: Mutation procedure

1. Initialize a mutation probability P_m and set $g = 1$.
2. Calculate the mutation probability of parent chromosome g via Equation (57), and generate a random number $u \in [0, 1]$.
3. If $u \leq P_m(g)$, select the parent chromosome g to produce an offspring chromosome z; otherwise, proceed to step 7.
4. Mutation for train departure time at departure station and for train type
   - 4.1. For each part of each basic chromosome, randomly generate two unrepeated gene locations in the intervals $[1, T_1/\delta]$ and $[1, N]$.
   - 4.2. Perform mutation for these two parts of the chromosome (exchange the values of the selected genes).
5. Use the gene feasibility repair strategy if chromosome z is infeasible.
6. Calculate the fitness function of chromosome z. If the fitness of z is better than that of g, use z to replace g; otherwise, retain g.
7. If $g \leq P_s$, set $g = g + 1$ and return to step 2.
8. Return a new population $M_g(g = 1, \cdots, P_s)$.

4.4.4. Adaptive crossover and mutation operators

During the genetic process, we construct adaptive crossover and mutation operators. If the GA consistently produces a better offspring, we reduce the incidence of the crossover and mutation operators; otherwise, we increase their incidence. The probabilities of the crossover and mutation operators are formulated as follows:

$$P_c(q_1, q_2) = \begin{cases} P_c \frac{F_M - f}{F_M - F_A} & f \geq F_A \\ P_c & otherwise \end{cases} \tag{56}$$

$$P_m(q) = \begin{cases} P_m \frac{F_M - f_q}{F_M - F_A} & f_q \geq F_A \\ P_m & otherwise \end{cases} \tag{57}$$

where f represents the higher fitness value between chromosomes q_1 and q_2; f_q is the fitness function of chromosome q; and F_M and F_A are the maximum and average fitness functions, respectively, for all the chromosomes.

4.4.5. Stopping criterion

After the A-GA process, we update the number of generations as $G = G + 1$. If G equals the maximum number of generations M_G, we terminate the A-GA. Otherwise, we produce the next generation.

4.5. General procedure

The aforementioned process is summarized in Algorithm 5, which makes use of Algorithms 1–4 (i.e., initialization, selection, crossover, and mutation procedures). The general procedure of our A-GA is based on the following seven steps. Step 1 involves the initialization of the parameterss. In Step 2, P_s feasible chromosomes are added to the initial population. In Step 3, the fitness-function values for all the chromosomes are calculated. In Steps 4–5, the next generation is produced through GA operations (selection, crossover, and mutation). In Step 6, the stopping criterion is checked, and the best solution found by the algorithm is output when the stopping condition is satisfied.

Algorithm 5: A-GA procedure

1. Set the parameters P_s, M_G, P_c, and P_m. Set the generation index $G = 1$.
2. Initialize P_s feasible chromosomes as the initial population using Algorithm 1.
3. Calculate the fitness-function values for all the chromosomes.
4. Select the chromosomes using Algorithm 2.
5. Produce the next generation through crossover and mutation operations using Algorithms 3 and 4.
6. If $G = M_G$, stop and return the best found solution; otherwise, set $G = G + 1$ and return to step 3.

5. Numerical experiments

We performed computational experiments to evaluate our models and the A-GA. To compare the computational performance of the A-GA with that of the CPLEX commercial solver, a realistic case was adopted, which is described in Section 5.1. CPLEX could directly solve this case for Models A and B (TSM) to obtain a near-optimal solution. To evaluate the efficiency of the A-GA and its potential to solve the studied problems, we focused on the computation time and optimality gap and compared our solution approach with CPLEX. As described in Section 5.2, a real-world case based on Line 16 of the Shanghai subway network was used to discuss the optimized solutions computed using Models C and D (ELM) with the real-world AFC data. The ELM and TSM solutions were compared to quantify the potential benefits of applying the corresponding strategies.

5.1. Realistic test case

We initially tested a set of 60 realistic instances, which varied in terms of the studied time horizon, granularity of traffic flow calculation (time unit), and number of trains in the network. These instances were generated according to the following key parameters:

- Studied time horizon (T_1: 60, 90, 120, and 150 min);
- Time unit (δ: 0.25, 0.5, 1, and 2 min);
- Number of trains (N: 4, 8, and 12).

The studied instances were identified using a four-field code: $X - T_1 - \delta - N$. Here, X indicates the model type, i.e., A, B, C, or D. For example, A-60-0.25-4 corresponds to a train timetable for Model A, where the time horizon of traffic optimization is 60 min, the time unit is 0.25 min, and the number of trains is 4. These 60 instances simulated the traffic flows at 13 stations, and the passenger demand between different ODs was randomly generated between 0 and 4 for each time unit. The network used for these 60 instances was based on Line 16 of the Shanghai subway network.

The aforementioned realistic instances were used for Models A and B, which are mixed-integer linear programming (MILP) formulations that can be directly solved by CPLEX. We compared the performance of the A-GA with that of CPLEX in terms of the computation time and solution quality. To obtain high-quality solutions (the best-known lower and upper bounds), all instances were solved by CPLEX 12.9 not only on an ordinary personal computer (PC) (i5-8250U CPU, 1.60 GHZ, 8 GB RAM) but also on a powerful PC (a server with two Intel Xeon CPUs E5-2699 v4, 2.20 GHz and 1.5 TB RAM, i.e., Window Server 2012 R2). On both the PCs, all the instances were initially solved with a maximum computation time of 3 h. For some instances, because CPLEX could not obtain an optimal solution in this timeframe, a maximum computation time of 3 d was imposed on the powerful PC. Thus, the best lower bound was obtained by the powerful PC in 3 d. The A-GA was implemented in MATLAB R2019b, and the corresponding experiments were conducted on the ordinary PC described above.

5.1.1. Computational results for Models A and B

The computational results for the aforementioned performance indicators obtained by CPLEX using the powerful PC are discussed in Appendix B. As indicated by the results in Table B1, 1) nearly half of the instances (27 of 60) were solved by CPLEX and the powerful PC to optimality within 3 h, and 2) for the most complex instances, we could not obtain an optimal solution within 3 h; however, the optimality gap could be significantly reduced when the maximum computation time was increased to 3 d (the average optimality gap for the 60 instances decreased from 35.63% to 7.86%).

CPLEX took a long time to obtain the near-optimal solutions; thus, a customized algorithm is needed to solve Models A and B in a reasonable timeframe.

5.1.2. Comparison between CPLEX and A-GA

We used the A-GA to quickly compute the near-optimal solutions with Models A and B. The tuning of the A-GA parameters is explained in Appendix C. To assess the performance of the A-GA, a detailed comparison of the A-GA and CPLEX solutions was conducted (summarized in Tables D1 and D2 in Appendix D), where three different combinations of P_s and M_G were investigated.

The A-GA is a heuristic algorithm with randomization factors, which implies that different runs do not necessarily yield the same results. Hence, to demonstrate the stability of the A-GA, each instance was solved by the A-GA multiple times. We performed 10 runs of the A-GA for each instance and identified the worst and best solutions. We then computed the average results for all the instances in terms of the best, average, and worst solutions. Therefore, the comparison between CPLEX and the A-GA was based on the following three A-GA cases: worst, average, and best solutions.

Figure 12 shows a cumulative comparison of the 60 realistic instances. The numbers 1–12 on the x-axis represent the following configurations: 1 and 2, CPLEX with the powerful PC with maximum computation times of 3 h and 3 d, respectively; 3, CPLEX with the normal PC within a maximum computation time of 3 h; and 4–12, worst, average, and best solutions (among 10 runs of the A-GA) computed by the A-GA with the normal PC with $P_s = 60, M_G = 60$, $P_s = 125, M_G = 125$ and $P_s = 250, M_G = 250$, respectively. The left-side y-axis indicates the average optimality gap (%), and the right-side y-axis indicates the average computation time (min).

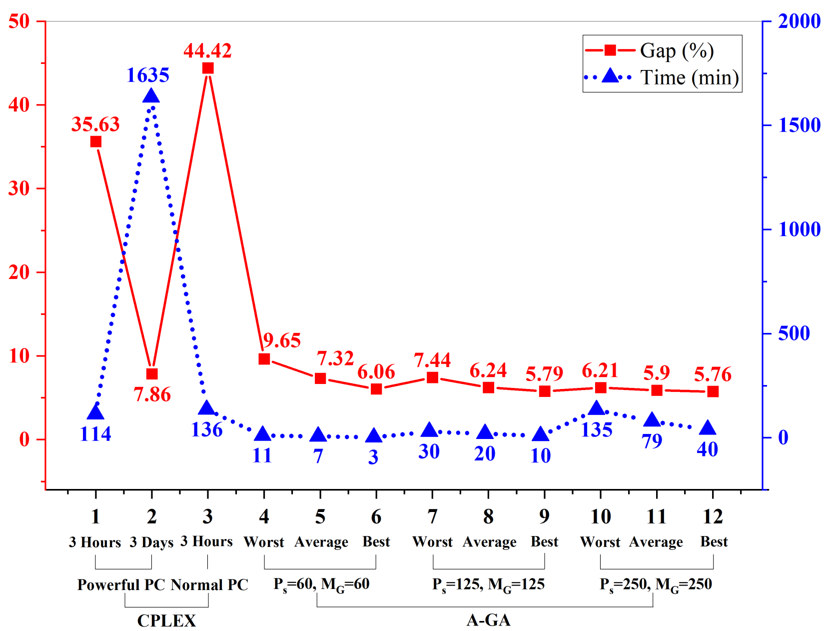


Figure 12 **CPLEX versus the A-GA on 60 instances~~: in terms of~~ optimality gap and computation time**

From the results shown in Figure 12, we observe the following.

1) The optimality gap obtained by CPLEX with the powerful PC and with 3 d of computation was significantly better than that obtained with 3-h of computation, as the computation time increased from 114 to 1635 min. Therefore, CPLEX is not suitable for practical applications, because a long computation time may be needed to compute near-optimal solutions. However, the practitioners must re-compute the timetables multiple times during the tactical planning; hence, a single optimization run should take no more than a few hours.

2) The average solutions obtained by the A-GA with $P_s = 60, M_G = 60$ (i.e., 7.32% and 7 min) were better than those obtained by CPLEX with the powerful PC within 3 d (i.e., 7.86% and 1635 min). When the P_s and M_t values were increased, the A-GA was better (on average) than CPLEX, even when the worst solutions computed by the A-GA were considered. The latter result confirms that the solution quality of the A-GA was superior to that of CPLEX. With regard to the computation time, the A-GA was significantly faster than CPLEX—particularly when P_s and M_t were assigned values up to 60. Thus, the A-GA stably computed near-optimal solutions for the tested instances and reduced the computation time significantly compared with the CPLEX solver.

3) When the values of P_s and M_G were increased, the difference (gap) between the best and worst solutions decreased, indicating that the A-GA was even more stable for larger values of P_s and M_G. This was expected, as a larger population and more iterations may lead to the exploration of a larger number of solutions, increasing the probability of obtaining a near-optimal solution. The drawback is that the computation time may increase proportionally.

5.2. Case study on Shanghai subway line 16

The analysis in Section 5.1 demonstrates that the A-GA can obtain high-quality solutions for realistic instances. We now discuss the computational performance of the A-GA on real-world instances of train timetabling with the ELM (Models C and D). In these practical-size instances, no feasible solution could be obtained by CPLEX, even after a long computation time. Thus, we used the A-GA to solve the more sophisticated models, i.e., Models C and D.

5.2.1. Case study description

Line 16 is one of the busiest SRT lines in the Shanghai subway network and connects PuDong District with the city's core areas. As shown in Figure 13, this line comprises 13 stations and extends from Long Yang Road Station to Di Shui Hu Station. Long Yang Road Station is a transfer station that connects Lines 2 and 7. Travel from Long Yang Road Station to Di Shui Hu Station is the focus of our case study.

The studied time horizon comprised the morning peak hours from 7:00 a.m. to 9:00 a.m., on a typical Monday with oversaturated traffic conditions. In the studied time horizon of Line 16, the departure interval was 5 min (however, in practice, this is not the minimum technical headway time); consequently, 24 trains were used in this period. The passenger demand data were obtained from the smart card AFC system. The express trains were set to stop at stations 2, 3, 9, and 11, according to the passenger flow distribution. The maximum capacity of each train was 842, and stations LSR, HTD, YSDWY, and HND had an additional overtaking track, making train overtaking possible.

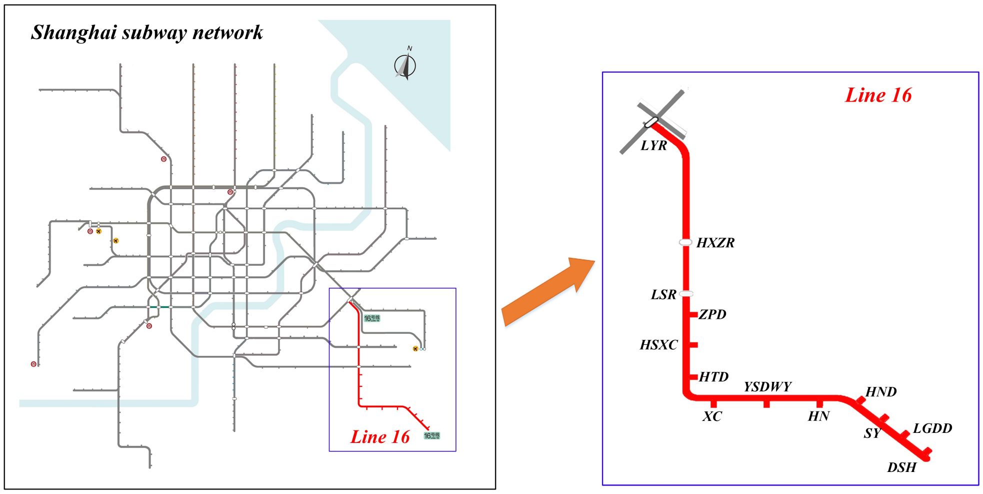


Figure 13 Line 16 of Shanghai subway network

The technical parameter values of the Shanghai metro trains are presented in Table 4. These values were based on the operational practices of the SRT lines. According to the analysis in Appendix C, the parameters P_s, M_G, P_c, and P_m of the A-GA were set to 60, 60, 0.9, and 0.4, respectively.

Table 4 Values of the parameters

Symbol	Value	Symbol	Value	Symbol	Value	Symbol	Value	Symbol	Value
T_1	120 min	M_d	3.5 min	α_w/α_v	1.0	I_{dd}	120 sec	I_{sa}	120 sec
δ	15 sec	ϖ_a	15 sec	I_{ao}	60 sec	I_{aa}	120 sec	I_{ss}	120 sec
w_{ik}	45 sec	ϖ_d	15 sec	I_{od}	90 sec	I_{da}	90 sec	I_{ds}	150 sec

5.2.2. Optimization results

The optimized timetable for the TSM is presented in Figure 14. As shown, there were only 24 local trains. The optimized timetable for the ELM without overtaking is presented in Figure 15. In this case, there were three express trains and 21 local trains. Because overtaking between trains was not allowed, this lack of operational flexibility may weaken the benefits of the ELM. Thus, it is essential to integrate overtaking into the timetabling with the ELM. As shown in Figure 16, the optimized timetable with the ELM and overtaking possibility utilized 13 express trains and 11 local trains. The express trains frequently overtook local trains to reduce the total PTT.

The results for the PTT and computation time under the TSM and ELM are presented in Figure 17. As shown, the PTTs were 9208 and 9751 h for the ELMs with and without overtaking, respectively. This indicates that the PTT was reduced by approximately 6% by applying the overtaking strategy. For the timetable under the TSM, the PTT reached 9822 h, which was even longer than that obtained by the timetable under the ELM without overtaking. Compared with

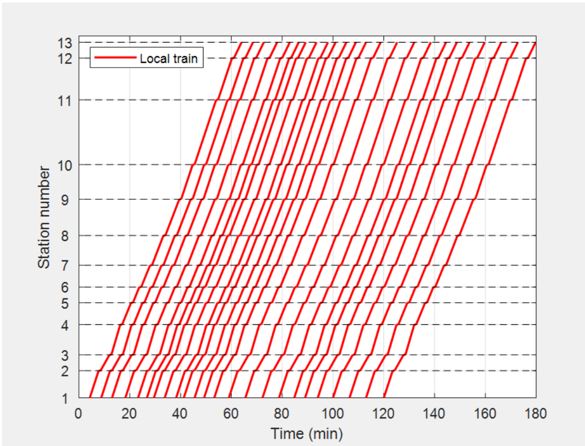


Figure 14 Optimized train timetable under the TSM

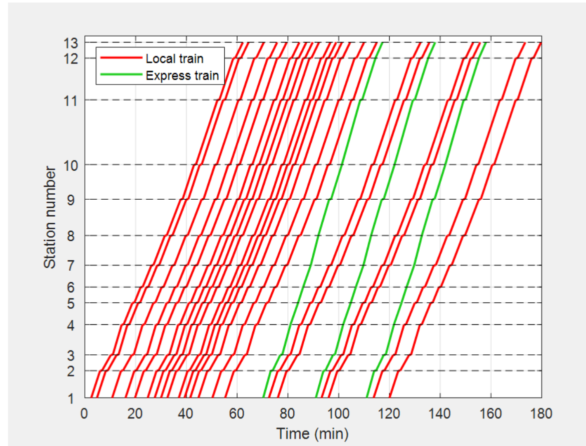


Figure 15 Optimized train timetable under the ELM without overtaking

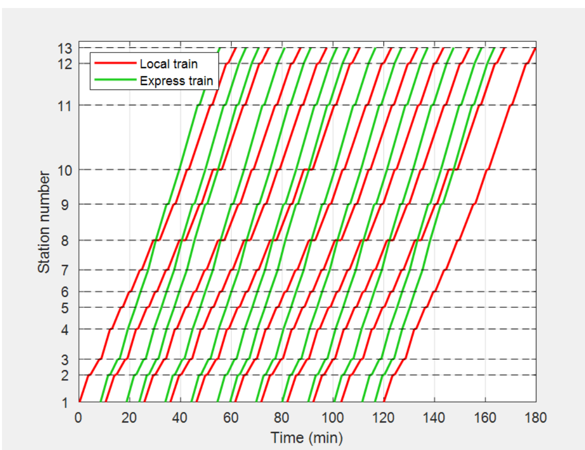


Figure 16 Optimized train timetable under the ELM with overtaking

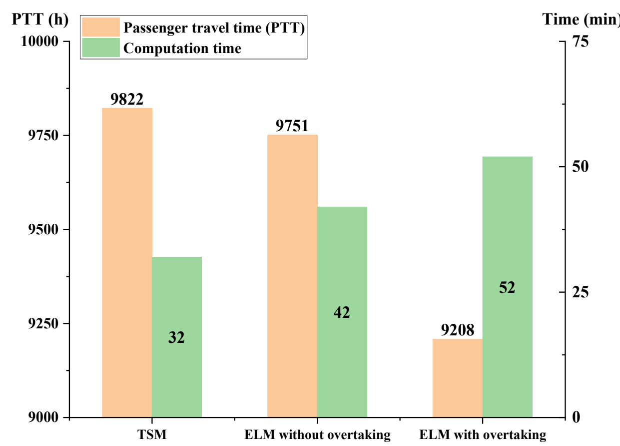


Figure 17 PTT and computation time for various train timetabling strategies

the TSM, the PTT was reduced by approximately 7% by applying the ELM with overtaking. In summary, it is beneficial to apply the ELM strategy in SRT timetabling, as it reduces the PTT; this benefit can be enhanced when overtaking between trains is allowed.

The green bars in Figure 17 indicate the computation times needed to obtain the optimized timetables using the A-GA. As shown, the A-GA took only 32 min to obtain an optimized timetable under the TSM. The computation time increased to 42 min when the ELM was considered and increased further to 52 min when overtaking was also allowed. These results imply that the timetabling problem under the ELM with the overtaking possibility is the most complex, perhaps because of the number of ELTs accommodated, and that the overtaking decisions must be optimized over a wide range of values.

5.2.3. Sensitivity analysis

To show the impact of passengers' endurance of waiting time on the train timetable under the ELM with overtaking, we conducted a sensitivity analysis on the perception coefficient of the waiting times, using the practical OD passenger demand from 7:00 a.m. to 8:00 a.m. on Shanghai Metro Line 16.

We recall that passengers perceive that time passes slowly when waiting on a station platform. Therefore, a parameter reflecting the passengers' perception of waiting times α_w is introduced. The waiting time perceived by passengers is α_w multiplied by the actual waiting time. However, further investigations are needed to identify a suitable value of α_w, which may depend on the types of people who are being analyzed. This motivated us to study the changing trend of the PTT when varying α_w; thus, we varied α_w from 1.0 to 3.0 in increments of 0.5. The corresponding results for the PTT and optimal number of ELTs are shown in Figure 18. Here, the x-axis indicates α_w, the left-side y-axis indicates the total PTT, and the right-side y-axis indicates the number of ELTs.

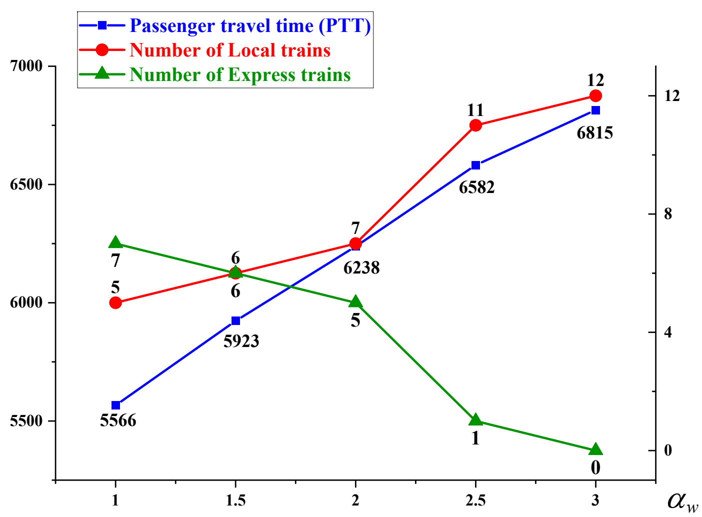


Figure 18 PTT and optimal number of ELTs for different values of α_w

As shown in Figure 18, as α_w increased, the optimal number of local trains increased, while the number of express trains decreased. This establishes the trend that when α_w increased and the passenger-perceived waiting times increased, the effect of saving passenger in-vehicle time was reduced. A possible solution to this trend is to reduce the number of express trains when α_w increases. The optimal number of ELTs must be determined according to the total PTT, which is affected by α_w.

6. Conclusions and further research

In this study, we investigated the train timetabling problems in suburban transit lines. First, we studied the train timetabling problem under the TSM, where all the trains are of the same type and stop at each station. Second, we generalized the problem to incorporate train capacity constraints. Third, according to the characteristics of the passenger flows in suburban transit lines, we studied the train timetabling problem under the ELM, which is frequently used in practice. In the ELM, express trains can skip certain stations, while local trains stop at each station. Fourth, we addressed the train timetabling problem under the ELM with the possibility of train overtaking, i.e., express trains can overtake local trains at some stations. The objective of this study was to minimize the total PTT within a given planning horizon.

For the train timetabling problem under the TSM, we presented two MILP models: one with and one without train capacity constraints. For the train timetabling problem under the ELM, we presented two MINLP models, one with and one without the possibility of train overtaking. Owing to the NP-hardness of the studied problems, an A-GA was developed to provide near-optimal solutions within an acceptable computation time. The A-GA improves the classic GA by considering the fitness values of the parents and offspring during each solution generation phase. It employs adaptive crossover and mutation operators. The A-GA was customized to solve the train timetabling problem under the ELM and with the train-overtaking possibility.

To evaluate the performance of the proposed algorithm, we performed numerical experiments on 60 realistic instances. The computational results indicate that the A-GA provided near-optimal solutions with an average optimality gap of 7.32% within 7 min of computation, on average. The generalized IBM ILOG CPLEX MIP solver yielded an optimality gap of 7.86% on average, with a significantly longer computation time.

We used the proposed algorithm to solve a real case study involving a Shanghai subway line with up to 24 trains, 13 stops, and 120 min of planning horizon. This practical case study could not be directly solved by CPLEX in a reasonable time, whereas the computational results for the A-GA indicated that it provided feasible timetables in a short time. A sensitivity analysis was conducted, providing the following insights into the studied problems. The analysis quantified the interesting trend that for train operations with the ELM, the PTT was significantly influenced by the passengers' perception coefficients of waiting times. The ELM best satisfied the passengers' indicator, i.e., the total PTT, when the decreasing effect of the in-vehicle times was greater than the increasing effect of the waiting times.

We next discuss interesting possibilities for further research on topics closely related to the problems examined in this study. There remain several critical problems to be solved in the operations management of SRT systems. First, the dwell time process is frequently subject to uncertainty, for example, owing to crowded passenger flows. Stochastic models are an interesting option for dealing with such uncertainties. However, the distribution of stochastic variables is difficult to determine in advance. A possible research direction is the application of robust optimization techniques with consideration of different scenarios. As another research direction, in the train timetabling problem with the ELM, passengers may have individual preferences to take express or local trains. For example, some passengers may prefer to take a local train directly with no transfer, while others may prefer to transfer once or twice at overtaking stations to limit their waiting time at the stations (Corman, 2020). Second, in this study, we assumed that the passengers could not change their traveling modes after entering the SRT, whereas in reality, they may prefer to switch to other transportation modes and thus leave the railway system when the waiting time is too long. Therefore, as an interesting research direction, we propose the integration of passenger route choice strategies and passenger behavior into the studied train timetabling problem. Third, as reported by Corman et al. (2014) and Samà et al. (2017), the negative effects of disruptions of railway traffic may last for hours after the end of each disruption, and effective train scheduling and routing algorithms are needed during railway traffic disturbances. Therefore, train rescheduling under severe disturbances requires further study—particularly for metro lines with ELTs, in which several rescheduling measures are allowed, including train-overtaking, short-turning of trains, skip–stop patterns, and cancelation of train services. Finally, timetabling problems based on a bidirectional corridor (which is more complex than the one examined in the present study), along with the extension of our methodology to deal with more complex networks, deserve further investigation.

Acknowledgements: This research was supported by the International Exchange Program for Graduate Students, Tongji University (No. 2019 6310).

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

References

Angelova, M., Pencheva, T., 2011. Tuning genetic algorithm parameters to improve convergence time. Int. J. Chem. Eng. 2011. https://doi.org/10.1155/2011/646917

Barrena, E., Canca, D., Coelho, L.C., Laporte, G., 2014. Exact formulations and algorithm for the train timetabling problem with dynamic demand. Comput. Oper. Res. 44, 66–74. https://doi.org/10.1016/j.cor.2013.11.003

Cacchiani, V., Galli, L., Toth, P., 2015. A tutorial on non-periodic train timetabling and platforming problems. EURO J. Transp. Logist. 4, 285–320. https://doi.org/10.1007/s13676-014-0046-4

Cacchiani, V., Qi, J., Yang, L., 2020. Robust optimization models for integrated train stop planning and timetabling with passenger demand uncertainty. Transp. Res. Part B Methodol. 136, 1–29.

https://doi.org/10.1016/j.trb.2020.03.009

Cai X, Goh CJ, M.A., 1998. Greedy heuristics for rapid scheduling of trains on a single track. IIE Trans 30, 481–493. https://doi.org/10.1023/A

Canca, D., Barrena, E., Algaba, E., Zarzo, A., 2014. Design and analysis of demand-adapted railway timetables. J. Adv. Transp. 48, 119–137. https://doi.org/10.1002/atr.1261

Caprara, A., Fischetti, M., Toth, P., 2002. Modeling and solving the train timetabling problem. Oper. Res. 50, 851-861+916. https://doi.org/10.1287/opre.50.5.851.362

Clarke, D.B., 1995. An Examination of Railroad Capacity and its Implications for Rail-Highway Intermodal Transportation 188.

Colin R. Reeves and Jonathan E. Rowe. 2002. Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory. Kluwer Academic Publishers, USA.

Corman, F., 2020. Interactions and Equilibrium Between Rescheduling Train Traffic and Routing Passengers in Microscopic Delay Management: A Game Theoretical Study. Transp. Sci. 1–38. https://doi.org/10.1287/trsc.2020.0979

Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M., 2014. Dispatching and coordination in multi-area railway traffic management. Comput. Oper. Res. 44, 146–160. https://doi.org/10.1016/j.cor.2013.11.011

D.E. Goldberg (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, AddisonWesley, Reading, Massachusetts.

D'Ariano, A., Meng, L., Centulio, G., Corman, F., 2019. Integrated stochastic optimization approaches for tactical scheduling of trains and railway infrastructure maintenance. Comput. Ind. Eng. 127, 1315–1335. https://doi.org/10.1016/j.cie.2017.12.010

D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wiley Publishing Company, Massachusetts, Mass, USA, 1989.

Dong, X., Li, D., Yin, Y., Ding, S., Cao, Z., 2020. Integrated optimization of train stop planning and timetabling for commuter railways with an extended adaptive large neighborhood search metaheuristic approach. Transp. Res. Part C Emerg. Technol. 117, 102681. https://doi.org/10.1016/j.trc.2020.102681

Gao, Y., Kroon, L., Yang, L., Gao, Z., 2018. Three-stage optimization method for the problem of scheduling additional trains on a high-speed rail corridor. Omega (United Kingdom) 80, 175–191. https://doi.org/10.1016/j.omega.2017.08.018

Gao, Y., Yang, L., Gao, Z., 2016. Energy consumption and travel time analysis for metro lines with express/local mode. Transp. Res. Part D Transp. Environ. 93, 425–449. https://doi.org/10.1016/j.trd.2016.10.009

Grefenstette, J. J. (1986). Optimization of Control Parameters for Genetic Algorithms. IEEE Transactions on Systems, Man, Cybernetics. 16 (1), 122–128. https://doi.org/10.1016/B978-044452701-1.00039-9

Gu, W., Amini, Z., Cassidy, M.J., 2016. Exploring alternative service schemes for busy transit corridors. Transp. Res. Part B Methodol. 93, 126–145. https://doi.org/10.1016/j.trb.2016.07.010

Hassannayebi, E., Zegordi, S.H., 2017. Variable and adaptive neighbourhood search algorithms for rail rapid transit timetabling problem. Comput. Oper. Res. 78, 439–453.

https://doi.org/10.1016/j.cor.2015.12.011

Hansen and Pachl (2014). I.A. Hansen, J. Pachl (Eds.), Railway Timetabling &Operations, Eurailpress, Hamburg, Germany

Hong, X., Meng, L., D'Ariano, A., Veelenturf, L.P., Long, S., Corman, F., 2021. Integrated optimization of capacitated train rescheduling and passenger reassignment under disruptions. Transp. Res. Part C Emerg. Technol. 125, 103025. https://doi.org/10.1016/j.trc.2021.103025

Huang, Y., Mannino, C., Yang, L., Tang, T., 2020. Coupling time-indexed and big-M formulations for real-time train scheduling during metro service disruptions. Transp. Res. Part B Methodol. 133, 38–61. https://doi.org/10.1016/j.trb.2019.12.005

Huang, Y., Yang, L., Tang, T., Cao, F., Gao, Z., 2016. Saving energy and improving service quality: Bicriteria train scheduling in urban rail transit systems. IEEE Trans. Intell. Transp. Syst. 17, 3364–3379. https://doi.org/10.1109/TITS.2016.2549282

Huang, Y., Yang, L., Tang, T., Gao, Z., Cao, F., 2017. Joint train scheduling optimization with service quality and energy efficiency in urban rail transit networks. Energy 138, 1124–1147. https://doi.org/10.1016/j.energy.2017.07.117

Jiang, F., Cacchiani, V., Toth, P., 2017. Train timetabling by skip-stop planning in highly congested lines. Transp. Res. Part B Methodol. 104, 149–174. https://doi.org/10.1016/j.trb.2017.06.018

J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan

K. A. DeJong, Analysis of the behavior of a class of genetic adaptive systems, Ph.D. thesis, Dept. Computer and Communication Sciences, Univ. of Michigan, 1975.

Li, Z., Mao, B., Bai, Y., Chen, Y., 2019. Integrated Optimization of Train Stop Planning and Scheduling on Metro Lines with Express/Local Mode. IEEE Access 7, 88534–88546. https://doi.org/10.1109/ACCESS.2019.2921758

Liu, R., Li, S., Yang, L., 2020. Collaborative optimization for metro train scheduling and train connections combined with passenger flow control strategy. Omega (United Kingdom) 90, 101990. https://doi.org/10.1016/j.omega.2018.10.020

Luan, X., Miao, J., Meng, L., Corman, F., Lodewijks, G., 2017. Integrated optimization on train scheduling and preventive maintenance time slots planning. Transp. Res. Part C Emerg. Technol. 80, 329–359. https://doi.org/10.1016/j.trc.2017.04.010

Meng, L., 2018. Special issue on Integrated optimization models and algorithms in rail planning and control 88, 87–90. https://doi.org/10.1016/j.trc.2018.01.003

Meng, L., Zhou, X., 2019. An integrated train service plan optimization model with variable demand: A team-based scheduling approach with dual cost information in a layered network. Transp. Res. Part B Methodol. 125, 1–28. https://doi.org/10.1016/j.trb.2019.02.017

Mo, P., Yang, L., Wang, Y., Qi, J., 2019. A flexible metro train scheduling approach to minimize energy cost and passenger waiting time. Comput. Ind. Eng. 132, 412–432. https://doi.org/10.1016/j.cie.2019.04.031

Nachtigall, K., Voget, S., 1996. A genetic algorithm approach to periodic railway synchronization.

Comput. Oper. Res. 23, 453–463. https://doi.org/10.1016/0305-0548(95)00032-1

Niu, H., Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. Transp. Res. Part C Emerg. Technol. 36, 212–230. https://doi.org/10.1016/j.trc.2013.08.016

Niu, H., Zhou, X., Gao, R., 2015. Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints. Transp. Res. Part B Methodol. 76, 117–135. https://doi.org/10.1016/j.trb.2015.03.004

O. Roeva, Improvement of Genetic Algorithm Performance for Identification of Cultivation Process Models, Advanced Topics on Evolutionary Computing, Artificial Intelligence Series-WSEAS, 2008.

Pettit, E., Swigger, K.M., 1983. An Analysis of Genetic-Based Pattern Tracking and Cognitive-Based Component Tracking Models of Adaptation. pp. 327–332.

Qi, J., Li, S., Gao, Y., Yang, K., Liu, P., 2018. Joint optimization model for train scheduling and train stop planning with passengers distribution on railway corridors. J. Oper. Res. Soc. 69, 556–570. https://doi.org/10.1057/s41274-017-0248-x

Robenek, T., Azadeh, S.S., Maknoon, Y., de Lapparent, M., Bierlaire, M., 2018. Train timetable design under elastic passenger demand. Transp. Res. Part B Methodol. 111, 19–38. https://doi.org/10.1016/j.trb.2018.03.002

Samà, M., D'Ariano, A., Corman, F., Pacciarelli, D., 2017. A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. Comput. Oper. Res. 78, 480–499. https://doi.org/10.1016/j.cor.2016.02.008

Shang, P., Li, R., Liu, Z., Yang, L., Wang, Y., 2018. Equity-oriented skip-stopping schedule optimization in an oversaturated urban rail transit network. Transp. Res. Part C Emerg. Technol. 89, 321–343. https://doi.org/10.1016/j.trc.2018.02.016

Sinha, S.K., Salsingikar, S., SenGupta, S., 2016. An iterative bi-level hierarchical approach for train scheduling. J. Rail Transp. Plan. Manag. 6, 183–199. https://doi.org/10.1016/j.jrtpm.2016.06.004

Souai, N., Teghem, J., 2009. Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. Eur. J. Oper. Res. 199, 674–683. https://doi.org/10.1016/j.ejor.2007.10.065

Srinivas, M., Patnaik, L.M., 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. IEEE Trans. Syst. Man Cybern. 24, 656–667. https://doi.org/10.1109/21.286385

Sun, L., Jin, J.G., Lee, D.H., Axhausen, K.W., Erath, A., 2014. Demand-driven timetable design for metro services. Transp. Res. Part C Emerg. Technol. 46, 284–299. https://doi.org/10.1016/j.trc.2014.06.003

Vansteenwegen, P., Oudheusden, D. Van, 2007. Decreasing the passenger waiting time for an intercity rail network 41, 478–492. https://doi.org/10.1016/j.trb.2006.06.006

Vuchic, V.R., 2005. Urban Transit: Operation, Planning and Economics. John Wiley &Sons Inc, New Jersey, USA

Wang, Y., De Schutter, B., Van Den Boom, T.J.J., Ning, B., Tang, T., 2014. Efficient Bilevel approach for urban rail transit operation with stop-skipping. IEEE Trans. Intell. Transp. Syst. 15, 2658–2670. https://doi.org/10.1109/TITS.2014.2323116

Wang, Y., D'Ariano, A., Yin, J., Meng, L., Tang, T., &Ning, B. (2018). Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line. Transportation Research Part B: Methodological, 118, 193-227.

Wang, Y., Liao, Z., Tang, T., Ning, B., 2017a. Train scheduling and circulation planning in urban rail transit lines. Control Eng. Pract. 61, 112–123. https://doi.org/10.1016/j.conengprac.2017.02.006

Wang, Y., Tang, T., Ning, B., Meng, L., 2017b. Integrated optimization of regular train schedule and train circulation plan for urban rail transit lines. Transp. Res. Part E Logist. Transp. Rev. https://doi.org/10.1016/j.tre.2017.06.001

Wang, Y., Tang, T., Ning, B., van den Boom, T.J.J., De Schutter, B., 2015. Passenger-demands-oriented train scheduling for an urban rail transit network. Transp. Res. Part C Emerg. Technol. 60, 1–23. https://doi.org/10.1016/j.trc.2015.07.012

Wardman, M., 2004. Public transport values of time. Transp. Policy 11, 363–377. https://doi.org/10.1016/j.tranpol.2004.05.001

Xie, J., Wong, S.C., Zhan, S., Lo, S.M., Chen, A., 2020. Train schedule optimization based on schedule-based stochastic passenger assignment. Transp. Res. Part E Logist. Transp. Rev. 136. https://doi.org/10.1016/j.tre.2020.101882

Yang, A., Wang, B., Huang, J., Li, C., 2020. Service replanning in urban rail transit networks: Cross-line express trains for reducing the number of passenger transfers and travel time. Transp. Res. Part C 115, 102629. https://doi.org/10.1016/j.trc.2020.102629

Yang, L., Qi, J., Li, S., Gao, Y., 2016. Collaborative optimization for train scheduling and train stop planning on high-speed railways. Omega (United Kingdom) 64, 57–76. https://doi.org/10.1016/j.omega.2015.11.003

Yang, X., Li, X., Ning, B., Tang, T., 2015. An optimisation method for train scheduling with minimum energy consumption and travel time in metro rail systems. Transp. B 3, 79–98. https://doi.org/10.1080/21680566.2015.1007577

Yin, J., Tang, T., Yang, L., Gao, Z., Ran, B., 2016. Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. Transp. Res. Part B Methodol. 91, 178–210. https://doi.org/10.1016/j.trb.2016.05.009

Yin, J., Yang, L., Tang, T., Gao, Z., Ran, B., 2017. Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. Transp. Res. Part B Methodol. 97, 182–213. https://doi.org/10.1016/j.trb.2017.01.001

Yin, J., D'Ariano, A., Wang, Y., Yang, L., Tang, T., 2021. Timetable coordination in a rail transit network with time-dependent passenger demand. Eur. J. Oper. Res. 1. https://doi.org/10.1016/j.ejor.2021.02.059

Yue, Y., Wang, S., Zhou, L., Tong, L., Saat, M.R., 2016. Optimizing train stopping patterns and schedules for high-speed passenger rail corridors. Transp. Res. Part C Emerg. Technol. 63, 126–146. https://doi.org/10.1016/j.trc.2015.12.007

Yun, Y., Gen, M., 2003. Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics. Fuzzy Optim. Decis. Mak. 2, 161–175. https://doi.org/10.1023/A:1023499201829

Zhang, C., Gao, Y., Yang, L., Gao, Z., Qi, J., 2020. Joint optimization of train scheduling and maintenance planning in a railway network: A heuristic algorithm using Lagrangian relaxation. Transp. Res. Part B Methodol. 134, 64–92. https://doi.org/10.1016/j.trb.2020.02.008

Zhang, P., Liu, X., Chen, M., 2016. Optimal train scheduling under a flexible skip-stop scheme for urban rail transit based on smartcard data, in: 2016 IEEE International Conference on Intelligent Rail Transportation, ICIRT 2016. https://doi.org/10.1109/ICIRT.2016.7588545

Zhang, T., Li, D., Qiao, Y., 2018. Comprehensive optimization of urban rail transit timetable by minimizing total travel times under time-dependent passenger demand and congested conditions. Appl. Math. Model. 58, 421–446. https://doi.org/10.1016/j.apm.2018.02.013

Zhang, Y., D'Ariano, A., He, B., Peng, Q., 2019. Microscopic optimization model and algorithm for integrating train timetabling and track maintenance task scheduling. Transp. Res. Part B Methodol. 127, 237–278. https://doi.org/10.1016/j.trb.2019.07.010

Appendix A: Problem description

A.1 Different forms of timetables

Train timetables using the TSM and ELM are shown in Figure 19. The main difference between the two modes is that in the TSM, all trains are local and stop at all stations, whereas in the ELM, express trains only stop at a subset of stations.

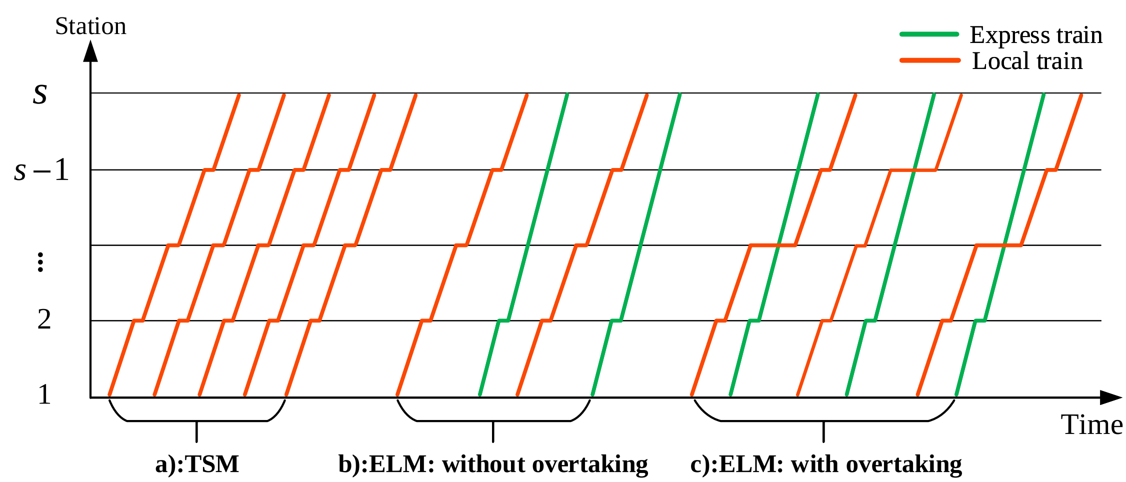


Figure 19 Train timetables under the TSM and ELM

In the ELM, an express train can catch up to a local train because of the difference in their numbers of stoppages. However, overtaking can only occur under certain conditions (e.g., where an additional overtaking track is provided). Figure 19 shows two types of ELMs: with (b) and without (c) overtaking.

The subway sets an overtaking track in stations where overtaking occurs; hence, an express train overtaking a local train is not required to stop at the overtaking station. As shown in Figure 20, case (a) allows overtaking, whereas case (b) does not.

In train operations with the TSM, a train's state at a station can be categorized as either arrival or departure. There are three types of headways between adjacent trains: I_{aa}, I_{da}, and I_{dd}, which indicate

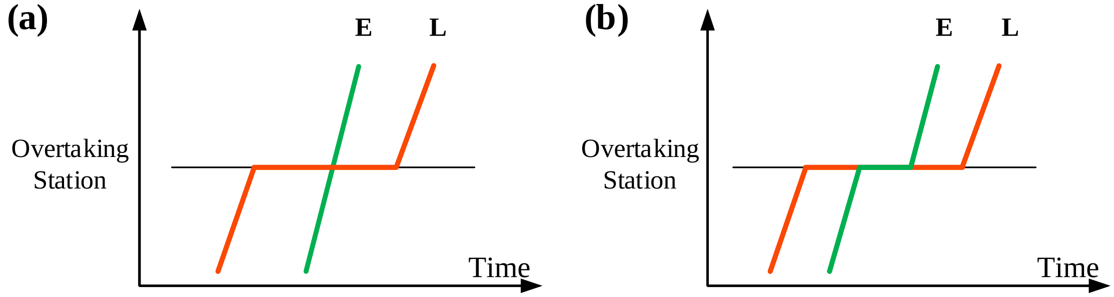


Figure 20 **Different overtaking situations: (a) overtaking is allowed; (b) overtaking is not allowed**

the minimum time interval between two successive trains in states of arrival–arrival, departure–arrival, and departure–departure, respectively, as shown in Figure 21(a). In train operations with the ELM, a train's state at the stopping station, where both express and slow trains stop, can also be categorized as either arrival or departure, and there are three types of headways, as shown in Figure 21(b).

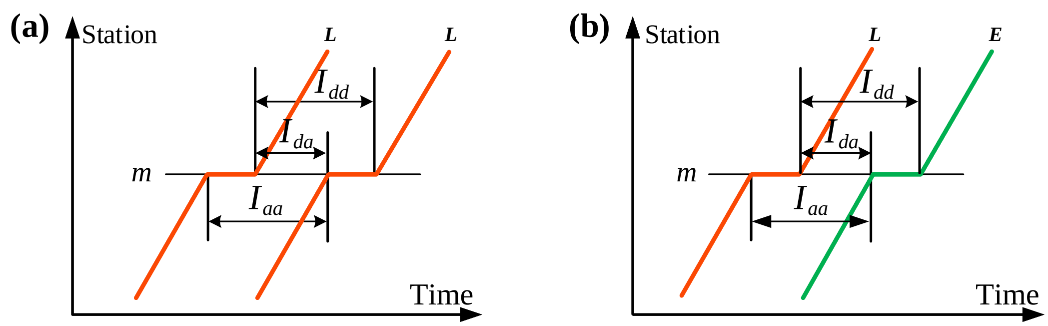


Figure 21 **Headways between trains: (a) TSM; (b) ELM in a station where express/local trains stop**

In train operations with the ELM, an express train's state at stop–skipping stations can be categorized as arrival, departure, or skipping at non-overtaking stations and as arrival, departure, and overtaking at overtaking stations. Figure 22 shows a train's headway in the ELM at a stop–skipping station.

As shown in Figure 22, there are three types of headways at each non-overtaking station between adjacent trains, i.e., I_{ds}, I_{sa}, and I_{ss}, indicating the minimum time interval between departure–skipping, skipping–arrival, and skipping–skipping, respectively, of two successive trains. There are two types of headways at each overtaking station, i.e., I_{ao} and I_{od}, which indicate the minimum time interval between arrival–overtaking and overtaking–departure, respectively, of two successive trains. It is thus evident that the ELM has more complex time headway constraints than the TSM.

A.2 Passenger waiting and in-vehicle times

The passenger waiting time is defined as the departure time of the boarded train minus the arrival time of the passenger at the station. Taking passenger p_{35}^{7} in Figure 2 as an example, his/her waiting time at station k is labeled as "waiting time of p_{35}^{7}" in Figure 23. Let us assume that another passenger p_{58}^{12} is waiting at station 5, which is skipped by the express trains; hence, this passenger must wait for local trains. If the first local train i has an insufficient capacity, the passenger must wait for the next local train i_4, which is

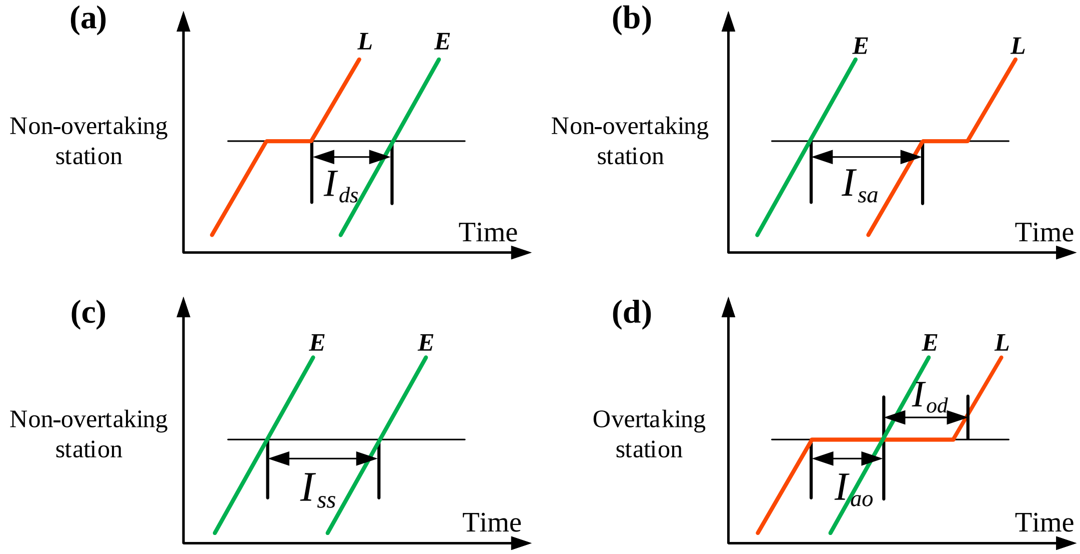


Figure 22 Headways between trains in the ELM for a stop–skipping station

overtaken by express train i_5 at station 5; accordingly, the passenger waiting time at station 5 is labeled in Figure 23 as "waiting time of p_{58}^{12}." Therefore, in train operations with the ELM, the passengers waiting at the stations skipped by express trains may wait longer than those for the TSM, resulting in longer PTTs under busy traffic conditions.

Because passengers usually feel that time goes slowly while waiting on the platform, the weight of the waiting time in the total travel time can be increased appropriately (Vansteenwegen and Oudheusden, 2007). In this study, a perception coefficient of waiting time α_w was introduced to calculate the waiting time perceived by the passengers. A long waiting time to board the train may significantly affect the passenger's satisfaction with rail transit, causing them to select another mode of transportation.

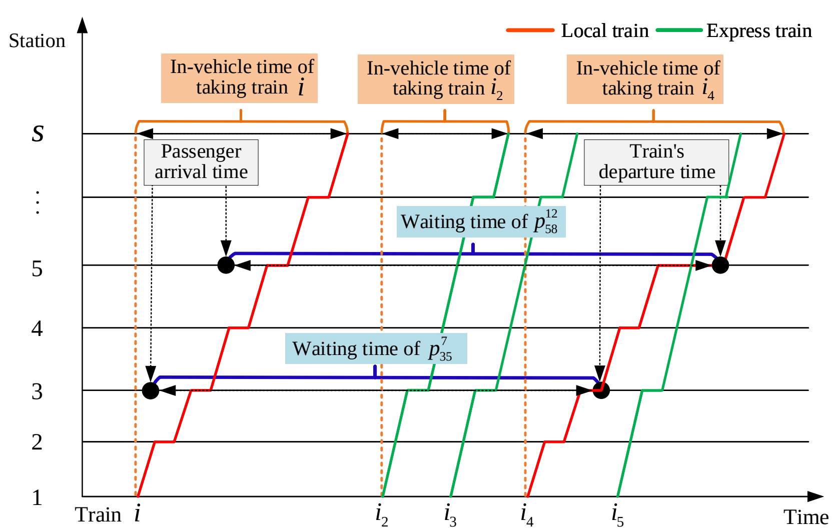


Figure 23 Passenger waiting and in-vehicle times

The passenger in-vehicle time is defined as the arrival time of the boarded train at the passenger's destination station subtracted by the departure time from his/her origin station. Assume that a passenger with the destination of station "s" waits at station "1". He/she can take local train i, express train i_2, or local train i_4, and the corresponding in-vehicle times are presented in Figure 23 as "in-vehicle time of taking train i", "in-vehicle time of taking train i_2", and "in-vehicle time of taking train i_4", respectively. Therefore, passengers with the same OD may have different in-vehicle times if they take different trains to their destination. Specifically, in train operations with the ELM, the in-vehicle times may decrease, and the waiting times of some passengers may increase; therefore, it is necessary to reasonably arrange the train timetables under the ELM. Only when the total travel times is reduced, an implementation of the ELM becomes meaningful.

Appendix B: Computational results for Models A and B obtained using CPLEX and powerful PC

Table B1 presents the computational results of a realistic test case with Models A and B (with CPLEX and the powerful PC), where LB and UB represent the lower and upper bounds, respectively; Gap represents the optimality gap ((UB-LB)/UB), which is expressed as a percentage; and Time represents the total computation time of the solver (min). In some instances, the best possible upper and lower bounds (the optimality gap being 0%) were obtained within 3 h, and we did not have to run the powerful PC for more than 3 h. Numbers marked with * are the best-known LBs, and grids marked with • indicate that the UB could not be obtained.

Table B1: Computational results for models A and B (obtained with CPLEX and the powerful PC)

Instance code	Maximum comp. time of 3 h				Maximum comp. time of 3 d			
	LB	UB	Gap (%)	Time (min)	LB	UB	Gap (%)	Time (min)
A-60-0.25-4	65333	82451	20.76	180	82205*	82205	0.00	201
A-60-0.25-8	9728	43756	77.77	180	33332*	43163	22.78	4320
A-60-0.25-12	21902	27753	21.08	180	25652*	27457	6.57	4320
A-60-0.5-4	82916*	82916	0.00	15				
A-60-0.5-8	36777	41565	11.52	180	41395*	41395	0.00	1516
A-60-0.5-12	27716*	27716	0.00	18				
A-90-0.5-4	166902	208876	20.10	180	208876*	208876	0.00	263
A-90-0.5-8	17346	126777	86.32	180	88249*	105830	16.61	3660
A-90-0.5-12	14438	91711	84.26	180	60271*	69831	13.69	4320
A-120-0.5-4	292268	363658	19.63	180	363658*	363658	0.00	249
A-120-0.5-8	21320	209894	89.84	180	145655*	193049	24.55	4320
A-120-0.5-12	16118	190164	91.52	180	18637*	136899	86.39	4320
A-150-0.5-4	432704	565874	23.53	180	565874*	565874	0.00	2220
A-150-0.5-8	23749	386540	93.86	180	202578*	333023	39.17	4320
A-150-0.5-12	15449	546168	97.17	180	19284*	266189	92.76	4320
A-60-1-4	83496*	83496	0.00	2				
A-60-1-8	41974*	41974	0.00	13				
A-60-1-12	28295*	28295	0.00	2				
A-90-1-4	209781*	209781	0.00	10				
A-90-1-8	94795	106575	11.05	180	104620*	104620	0.00	627
A-90-1-12	64850	70406	7.89	180	70039*	70058	0.03	990
A-120-1-4	364870*	364870	0.00	14				
A-120-1-8	31050	230031	86.50	180	171648*	182797	6.10	4320
A-120-1-12	91170	137217	33.56	180	112966*	122847	8.04	4320
A-150-1-4	567371*	567371	0.00	75				
A-150-1-8	37509	345810	89.15	180	235934*	284799	17.16	2820
A-150-1-12	29826	245953	87.87	180	163808*	190357	13.95	4320
A-60-2-4	83804*	83804	0.00	0				

A-60-2-8	42524*	42524	0.00	0				
A-60-2-12	29142*	29142	0.00	0				
A-90-2-4	210395*	210395	0.00	1				
A-90-2-8	105315*	105315	0.00	4				
A-90-2-12	70357*	70357	0.00	10				
A-120-2-4	365473*	365473	0.00	1				
A-120-2-8	183205*	183205	0.00	11				
A-120-2-12	122257*	122257	0.00	6				
A-150-2-4	567783*	567783	0.00	3				
A-150-2-8	283481*	283481	0.00	49				
A-150-2-12	181392	190125	4.59	180	189501*	189501	0.00	267
Average of Model A	131661	196960	27.13	98	163445	180835	8.92	1442
B-60-1-4	83535*	83535	0.00	112				
B-60-1-8	40765	42212	3.43	180	41986*	41986	0.00	200
B-60-1-12	28295*	28295	0.00	66				
B-90-1-4	69018	699437	90.13	180	209781*	209781	0.00	1020
B-90-1-8	12253	•	100.00	180	100160*	104661	0.04	4320
B-90-1-12	12685	274518	95.38	180	65064*	70897	1.21	4320
B-120-1-4	97601	789957	87.64	180	365200*	365200	0.00	1893
B-120-1-8	15839	•	100.00	180	17832*	711693	75.88	4320
B-120-1-12	14504	•	100.00	180	110267*	129753	12.94	4320
B-60-2-4	84266*	84266	0.00	5				
B-60-2-8	42524*	42524	0.00	13				
B-60-2-12	29142*	29142	0.00	10				
B-90-2-4	210395*	210395	0.00	135				
B-90-2-8	105315*	105315	0.00	178				
B-90-2-12	70357*	70357	0.00	170				
B-120-2-4	312139	391557	20.28	180	365473*	365473	0.00	1515
B-120-2-8	57096	1035651	94.49	180	178816*	183205	0.00	4320
B-120-2-12	27010	648303	95.83	180	119010*	123617	1.10	4320
B-150-2-4	163230	2137609	92.36	180	567783*	567783	0.00	1950
B-150-2-8	33256	•	100.00	180	254421*	392937	27.86	4320
B-150-2-12	32693	•	100.00	180	176850*	199033	4.79	4320
Average of Model B	73425	417067	51.41	144	153642	196183	5.90	1992
Average of Models A & B	111278	260991	35.63	114	160013	186207	7.86	1635

Appendix C: Tuning of A-GA parameters

There are several operators, functions, parameters, and settings in GA that can be ameliorated and implemented differently for various problems (Goldberg, 1989; Roeva, 2008). In this study, four of the main A-GA parameters were investigated: the population size (P_s), maximum number of generations (M_G), crossover probabilities (P_c), and mutation probabilities (P_m). GAs generally perform poorly with very small P_s values, because a small population provides an insufficient sample size for most hyperplanes (Pettit and Swigger, 1983). However, a large value P_s would require more evaluations per generation and possibly result in unacceptably slow convergence. Increasing M_G does not always improve the performance of the A-GA, particularly when the solution efficiency is considered. The crossover and mutation probabilities are the parameters that affect the frequency at which the crossover and mutation operators are applied. With a high P_c value, new chromosomes are quickly introduced into the population, whereas a low P_c value may cause stagnation owing to the lower exploration rate. Mutation increases the variability of the population and is often randomly applied with a low probability (Angelova and Pencheva, 2011). Because the population size and the maximum number of generations affect the solution quality and computation time of the A-GA, we first investigated different values of P_s and M_G and then different values of P_c and P_m. This step-by-step tuning was based on the importance of each tested parameter and on previous experience in tuning GAs (Grefenstette, 1986; Reeves C.R and Rowe J.E., 2002).

To fine-tune the parameters of the A-GA in our study, four instances were analyzed: A-60-0.25-12, A-60-0.5-12, A-60-1-12, and A-60-2-12. In our experiments, we evaluated five different values of P_s and M_G, over a range of $60 - 1000$. We also considered four different crossover (mutation) probabilities, varying from 0.6 to 0.9 (from 0.1 to 0.4) with increments of 0.1. The chosen numerical values are commonly used in GAs (Angelova and Pencheva, 2011; Yang et al., 2015). Additionally, because the A-GA is a heuristic algorithm with randomization factors, different runs do not necessarily yield the same result. Therefore, we performed 10 runs of the A-GA for each instance with the same set of parameters. The computational results for various values of P_s and M_G are presented in Table C1 and Figure 24. For instance, for A-60-0.25-12, because the computation time of $P_s = 250$ exceeded 3 h (marked by a *), the experimental results obtained with $P_s = 500$ and $P_s = 1000$ are not included in Table C1. Similarly, the results for instance A-60-0.5-12 with $P_s = 1000$ are not included. Figure 24 shows violin plots for the runs of A-GA with the different parameter configurations tested, in terms of the optimality gap (percentage) and computation time (seconds). In each violin plot, the box plot for the runs with the corresponding parameter configuration is shown, together with the related probability density. Each violin plot presents the median value (white dot), the first and third quantiles (thick black central lines), the minimum and maximum values (extremes of the thin central lines), and the kernel density estimation of the probability density function (rotated grey areas on each side of the central line).

Table C1: Computational results for different values of P_s and M_G

Par.Value		Average Optimality Gap (%)					Average Computation Time (min)				
P_s	M_G	A-60-0.25-12	A-60-0.5-12	A-60-1-12	A-60-2-12	Average	A-60-0.25-12	A-60-0.5-12	A-60-1-12	A-60-2-12	Average
60	60	7.48	1.00	0.80	0.14	2.36	28	16	7	4	14
	125	7.29	1.08	1.35	0.22	2.48	19	8	4	5	9
	250	7.88	1.90	0.77	0.29	2.71	18	8	7	4	9
	500	7.57	0.79	0.98	0.29	2.40	19	13	6	5	11
	1000	7.18	2.26	1.02	0.11	2.64	34	8	7	5	14
125	125	6.09	0.45	0.32	0.06	1.73	68	22	20	12	30
	250	6.18	0.83	0.43	0.12	1.89	74	53	19	11	39
	500	6.14	0.50	0.28	0.01	1.73	106	38	21	13	44
	1000	6.14	0.43	0.29	0.06	1.73	76	46	22	11	39
250	125	5.88	0.27	0.25	0.05	1.61	243*	117	46	34	110
	250	5.88	0.25	0.22	0.01	1.59	230*	132	54	33	112
	500	5.91	0.37	0.16	0.01	1.61	216*	115	45	30	101
	1000	6.05	0.14	0.28	0.00	1.62	178	112	43	31	91
500	125		0.24	0.14	0.00	0.13		252	181	90	174
	250		0.22	0.11	0.00	0.11		360	134	94	196•
	500		0.21	0.14	0.00	0.12		295	159	82	179
	1000		0.23	0.17	0.00	0.13		281	150	58	163
1000	125			0.07	0.00	0.03			302	164	233•
	250			0.03	0.00	0.02			416	198	307•
	500			0.07	0.00	0.04			416	155	285•
	1000			0.01	0.00	0.01			454	153	304•

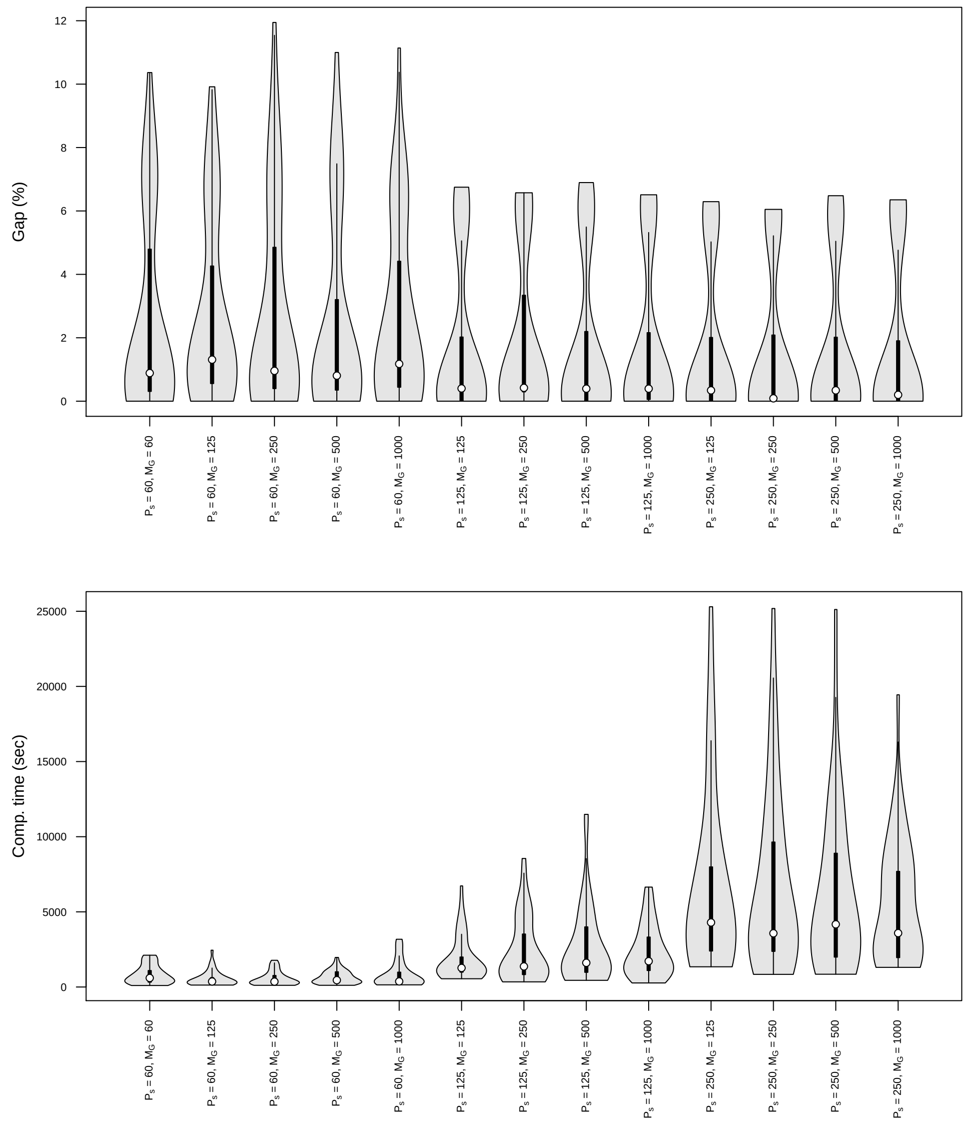


Figure 24 **Violin plots for the runs of the A-GA with different values of P_s and M_G**

As indicated by Table C1, although a reduced optimality gap can be obtained when P_s is fixed to 500 or 1000, the average computation time with the latter configuration often exceeds 3 h (marked with a •). Therefore, we preferred to set P_s to 60, 125, or 250 when using the A-GA for the experimental results presented in our paper. From Figure 24, we conclude that the main parameter affecting the performance of the proposed algorithm is the population size. On the one hand, a small population allows fast computation of high-quality solutions, even if the optimality gap can be affected by the population size. On the other hand, with a growing population, the optimality gap tends to be close to 0, but the corresponding computation time increases significantly. Taking into account our previous discussion on the results in Table C1, we select the following three parameter configurations: (i) $P_s = 60, M_G = 60$ as a fast-solving configuration with a good median value for the solution quality (optimality gap); (ii) $P_s = 250, M_G = 250$ as a configuration with a solution-quality median value close to zero, albeit requiring a significant (but still limited) computation time to compute its best-known solution; and (iii) $P_s = 125, M_G = 125$ as a good compromise configuration compared with configurations (i) and (ii), in terms of the computation time and optimality gap.

We continue our parameter tuning by evaluating different values of P_c and P_m, according to $P_s = 125, M_G = 125$. The computational results are summarized in Table C2, where we observe that the best solutions (the minimum optimality gap and computation time, marked with * in Table C2) were obtained with $P_c = 0.9$ and $P_m = 0.4$.

Table C2: Computational results for different values of P_c and P_m

Instance		Average Optimality Gap (%)					Average Computation Time (min)				
		A-60-0.25-12	A-60-0.5-12	A-60-1-12	A-60-2-12	Average	A-60-0.25-12	A-60-0.5-12	A-60-1-12	A-60-2-12	Average
P_c	0.6	6.69	0.78	0.53	0.19	2.05	49	54	10	8	30
	0.7	6.15	0.66	0.33	0.01	1.79	53	23	11	12	25
	0.8	6.67	0.41	0.51	0.16	1.94	47	29	9	10	24
	0.9	6.09	0.45	0.32	0.06	1.73*	68	22	20	12	30*
P_m	0.1	6.85	1.56	0.70	0.31	2.36	45	51	7	6	27
	0.2	7.07	1.07	0.74	0.09	2.24	45	16	10	9	20
	0.3	6.20	0.53	0.53	0.13	1.85	58	25	12	9	26
	0.4	6.09	0.45	0.32	0.06	1.73*	68	22	20	12	30*

In summary, for the A-GA, we selected the following optimum values of the studied key parameters: $P_s = 60, M_G = 60$; $P_s = 125, M_G = 125$, and $P_s = 250, M_G = 250$. Meanwhile, we set $P_c = 0.9$ and $P_m = 0.4$.

Appendix D: Computational results for Models A and B obtained with CPLEX and A-GA

Table D1 summarizes the computational results for the 60 instances for Models A and B obtained using CPLEX and the A-GA, as well as with the normal PC. In the CPLEX and Normal PC columns of Table D1, Gap1 and Gap2 represent the optimality gap computed with the lower bound obtained by the given configuration and the optimality gap computed with the best-known Lower Bound (LB), respectively. The optimality gap in the A-GA (i.e., Gap in Table D1) was obtained by comparing the A-GA average solution with the best-known LB. The grids marked with $*$ indicate that the UB could not be obtained.

Table D1: Computational results for models A and B (obtained using CPLEX/A-GA and the normal PC)

Instance code	CPLEX and Normal PC					A-GA and Normal PC								
	Maximum comp time 3 h					$P_s = 60, M_G = 60$			$P_s = 125, M_G = 125$			$P_s = 250, M_G = 250$		
	LB	UB	Gap1 (%)	Gap2 (%)	Time (min)	UB	Gap (%)	Time (min)	UB	Gap (%)	Time (min)	UB	Gap (%)	Time (min)
A-60-0.25-4	64101	83541	23.27	1.60	180	82982	0.94	6	82469	0.32	18	82371	0.20	55
A-60-0.25-8	9656	49741	80.59	32.99	180	41555	19.79	16	41064	18.83	36	40938	18.58	176
A-60-0.25-12	7251	29763	75.64	13.81	180	27726	7.48	28	27315	6.09	68	27253	5.88	230
A-60-0.5-4	82916	82916	0.00	0.00	51	83228	0.37	4	83075	0.19	9	82924	0.01	36
A-60-0.5-8	25173	42869	41.28	3.44	180	41836	1.05	8	41518	0.30	30	41400	0.01	108
A-60-0.5-12	27716	27716	0.00	0.00	84	27995	1.00	16	27841	0.45	22	27785	0.25	132
A-90-0.5-4	165480	214439	22.83	2.59	180	209909	0.49	6	209120	0.12	11	208876	0.00	50
A-90-0.5-8	17529	150949	88.39	41.54	180	106263	16.95	10	104332	15.42	36	103809	14.99	161
A-90-0.5-12	13537	105996	87.23	43.14	180	70543	14.56	16	69516	13.30	68	69272	12.99	236
A-120-0.5-4	167636	429402	60.96	15.31	180	365515	0.51	5	364317	0.18	20	363826	0.05	60

A-120-0.5-8	19438	329826	94.11	55.84	180	187068	22.14	10	182373	20.13	43	181877	19.92	199
A-120-0.5-12	14987	1172333	98.72	98.41	180	124560	85.04	13	121538	84.67	80	120912	84.59	316
A-150-0.5-4	71708	615888	88.36	8.12	180	569209	0.59	4	566849	0.17	24	565931	0.01	65
A-150-0.5-8	19516	457214	95.73	55.69	180	288359	29.75	12	284927	28.90	46	282297	28.24	211
A-150-0.5-12	15364	1162656	98.68	98.34	180	192423	89.98	21	188556	89.77	70	187756	89.73	314
A-60-1-4	83496	83496	0.00	0.00	12	83788	0.35	2	83569	0.09	7	83532	0.04	16
A-60-1-8	41974	41974	0.00	0.00	50	42320	0.82	6	42063	0.21	19	41981	0.02	40
A-60-1-12	28295	28295	0.00	0.00	15	28524	0.80	7	28387	0.32	20	28358	0.22	54
A-90-1-4	209781	209781	0.00	0.00	115	210507	0.34	3	210035	0.12	4	209870	0.04	25
A-90-1-8	91202	122100	25.31	14.32	180	106714	1.96	6	105357	0.70	16	105136	0.49	69
A-90-1-12	64988	72547	10.42	3.46	180	71008	1.36	10	70413	0.53	34	70179	0.20	101
A-120-1-4	364869	364869	0.00	0.00	112	366124	0.34	3	365227	0.10	7	364882	0.00	25
A-120-1-8	30461	283005	89.24	39.35	180	185634	7.53	7	183520	6.47	22	183019	6.21	89
A-120-1-12	25419	161808	84.29	30.19	180	123887	8.82	10	122606	7.86	35	122030	7.43	166
A-150-1-4	466916	582264	19.81	2.56	180	567957	0.10	4	567812	0.08	6	567371	0.00	35
A-150-1-8	34975	390717	91.05	39.62	180	287629	17.97	9	285194	17.27	27	283382	16.74	125
A-150-1-12	29057	295980	90.18	44.66	180	193021	15.13	14	190999	14.24	34	189031	13.34	218
A-60-2-4	83804	83804	0.00	0.00	1	84076	0.32	1	83804	0.00	4	83804	0.00	13
A-60-2-8	42524	42524	0.00	0.00	2	42779	0.60	3	42632	0.25	4	42551	0.06	28
A-60-2-12	29142	29142	0.00	0.00	3	29184	0.14	4	29160	0.06	12	29144	0.01	33
A-90-2-4	210395	210395	0.00	0.00	4	211284	0.42	2	210493	0.05	4	210405	0.00	12
A-90-2-8	105315	105315	0.00	0.00	40	106670	1.27	3	105539	0.21	7	105369	0.05	29
A-90-2-12	70357	70357	0.00	0.00	29	71346	1.39	7	70716	0.51	21	70611	0.36	50
A-120-2-4	365473	365473	0.00	0.00	15	366729	0.34	2	365503	0.01	3	365500	0.01	13
A-120-2-8	178913	183205	2.34	0.00	133	185720	1.35	4	183741	0.29	12	183383	0.10	58
A-120-2-12	116361	148283	21.53	17.55	180	124532	1.83	6	123240	0.80	25	122827	0.46	73
A-150-2-4	567783	567783	0.00	0.00	26	570668	0.51	2	568252	0.08	6	567847	0.01	20
A-150-2-8	172942	345679	49.97	17.99	180	287834	1.51	7	286038	0.89	13	283947	0.16	63
A-150-2-12	52602	225835	76.71	16.09	180	194010	2.32	11	191341	0.96	20	189811	0.16	84
Ave.A	107412	255638	38.89	17.86	124	178490	9.18	8	177191	8.49	24	176697	8.25	97
B-60-1-4	74784	83535	10.48	0.00	180	85464	2.26	1	84328	0.94	6	84079	0.65	15
B-60-1-8	39038	42674	8.52	1.61	180	43496	3.47	5	42114	0.30	16	42071	0.20	36
B-60-1-12	27340	28378	3.66	0.29	180	28541	0.86	9	28424	0.45	13	28381	0.30	65
B-90-1-4	69462	755298	90.80	72.23	180	211629	0.87	3	210632	0.40	4	210307	0.25	14
B-90-1-8	12842	*	100	100	180	106868	2.10	8	104915	0.28	21	104737	0.11	55
B-90-1-12	12909	*	100	100	180	72481	3.37	9	71089	1.48	23	70337	0.42	83
B-120-1-4	77414	1354433	94.28	73.04	180	373653	2.26	4	367460	0.62	9	366585	0.38	19
B-120-1-8	15694	*	100	100	180	186667	8.05	8	184030	6.73	22	183208	6.31	85
B-120-1-12	112966	*	100	100	180	126518	10.71	15	123520	8.54	24	122136	7.51	170
B-60-2-4	84266	84266	0.00	0.00	12	87564	3.77	1	86570	2.66	1	86570	2.66	7
B-60-2-8	42524	42524	0.00	0.00	33	43526	2.30	3	42677	0.36	4	42674	0.35	17
B-60-2-12	29142	29142	0.00	0.00	20	30501	4.46	3	30198	3.50	8	30206	3.52	38
B-90-2-4	183463	210395	12.80	0.00	180	217592	3.31	1	216563	2.85	2	214785	2.04	14
B-90-2-8	101665	107025	5.01	1.60	180	110190	4.42	4	105997	0.64	7	105939	0.59	46
B-90-2-12	69705	71285	2.22	1.30	180	74463	5.51	5	71968	2.24	14	70736	0.54	39
B-120-2-4	312817	483169	35.26	24.36	180	396791	7.89	2	382241	4.39	3	381493	4.20	6
B-120-2-8	66196	*	100	100	180	188504	2.81	5	183954	0.41	11	184043	0.46	46
B-120-2-12	39390	781003	94.96	84.35	180	129476	5.58	9	127086	3.80	11	123935	1.35	90
B-150-2-4	197078	2066777	90.46	72.53	180	570523	0.48	2	570862	0.54	4	568493	0.12	17
B-150-2-8	32964	*	100	100	180	288127	1.61	6	286028	0.89	13	283681	0.07	51
B-150-2-12	32008	*	100	100	180	199226	4.88	11	191945	1.27	18	190121	0.33	71
Ave.A	77794	438565	54.69	49.11	157	170086	3.86	5	167267	2.06	11	166406	1.54	47
Ave.A &B	97045	303958	44.42	28.80	136	175549	7.32	7	173717	6.24	20	173095	5.90	79

Notation: Ave.A, Ave.B and Ave.A &B mean Average of Model A, Average of Model B and Average of Models A and B, respectively.

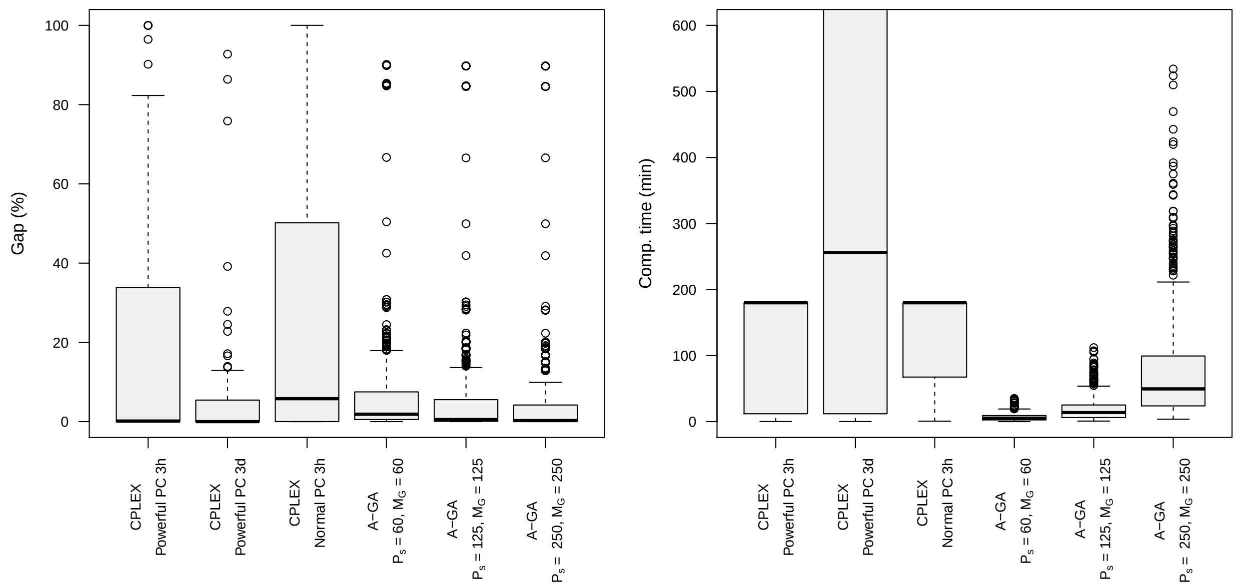
Table D2 compares the results obtained using CPLEX with the powerful PC (3 d) and the A-GA with the normal PC. Comparison 1 was performed with $P_s = 60, M_G = 60$; comparison 2 was performed with $P_s = 125, M_G = 125$; and comparison 3 was performed with $P_s = 250, M_G = 250$. Positive results for a comparison indicate that the A-GA objective-function or computation-time values were better than those for CPLEX, and negative values indicate the opposite. The numbers marked with ♦, ♠, and ► highlight the cases in which the CPLEX solutions were better than the A-GA solutions, considering a comparison gap improvement larger than 3%.

Table D2: Comparison between CPLEX and the A-GA

Instance	Comparison 1		Comparison 2		Comparison 3		Instance	Comparison 1		Comparison 2		Comparison 3	
Code	Gap1 (%)	Time1 (min)	Gap2 (%)	Time2 (min)	Gap3 (%)	Time3 (min)	Code	Gap1 (%)	Time1 (min)	Gap2 (%)	Time2 (min)	Gap3 (%)	Time3 (min)
A-60-0.25-4	-0.94	195	-0.32	183	-0.20	146	A-90-2-12	-1.39	2	-0.51	-11	-0.36	-40
A-60-0.25-8	2.99	4304	3.95	4284	4.20	4144	A-120-2-4	-0.34	-1	-0.01	-2	-0.01	-12
A-60-0.25-12	-0.90	4292	0.49	4252	0.70	4090	A-120-2-8	-1.35	6	-0.29	-1	-0.10	-48
A-60-0.5-4	-0.37	11	-0.19	6	-0.01	-21	A-120-2-12	-1.83	0	-0.80	-19	-0.46	-67
A-60-0.5-8	-1.05	1508	-0.30	1486	-0.01	1408	A-150-2-4	-0.51	1	-0.08	-3	-0.01	-17
A-60-0.5-12	-1.00	2	-0.45	-4	-0.25	-114	A-150-2-8	-1.51	42	-0.89	36	-0.16	-14
A-90-0.5-4	-0.49	257	-0.12	252	0.00	213	A-150-2-12	-2.32	256	-0.96	247	-0.16	183
A-90-0.5-8	-0.34	3650	1.20	3624	1.62	3499	**Ave.A**	-0.27	1434	0.43	1418	0.67	1345
A-90-0.5-12	-0.87	4304	0.39	4252	0.70	4084	B-60-1-4	-2.26	111	-0.94	106	-0.65	97
A-120-0.5-4	-0.51	244	-0.18	229	-0.05	189	B-60-1-8	-3.47♦	195	-0.30	184	-0.20	164
A-120-0.5-8	2.41	4310	4.42	4277	4.63	4121	B-60-1-12	-0.86	57	-0.45	53	-0.30	1
A-120-0.5-12	1.35	4307	1.72	4240	1.80	4004	B-90-1-4	-0.87	1017	-0.40	1016	-0.25	1006
A-150-0.5-4	-0.59	2216	-0.17	2196	-0.01	2155	B-90-1-8	-2.06	4312	-0.24	4299	-0.07	4265
A-150-0.5-8	9.42	4308	10.27	4274	10.93	4109	B-90-1-12	-2.16	4311	-0.27	4297	0.79	4237
A-150-0.5-12	2.78	4299	2.98	4250	3.03	4006	B-120-1-4	-2.26	1889	-0.62	1884	-0.38	1874
A-60-1-4	-0.35	0	-0.09	-6	-0.04	-14	B-120-1-8	67.84	4312	69.15	4298	69.57	4235
A-60-1-8	-0.82	7	-0.21	-6	-0.02	-27	B-120-1-12	2.23	4305	4.39	4296	5.43	4150
A-60-1-12	-0.80	-5	-0.32	-18	-0.22	-52	B-60-2-4	-3.77♦	3	-2.66	3	-2.66	-2
A-90-1-4	-0.34	7	-0.12	6	-0.04	-15	B-60-2-8	-2.30	10	-0.36	9	-0.35	-4
A-90-1-8	-1.96	621	-0.70	611	-0.49	558	B-60-2-12	-4.46♦	7	-3.50♠	3	-3.52►	-28
A-90-1-12	-1.34	980	-0.50	956	-0.17	889	B-90-2-4	-3.31♦	134	-2.85	133	-2.04	121
A-120-1-4	-0.34	10	-0.10	6	0.00	-11	B-90-2-8	-4.42♦	174	-0.64	171	-0.59	132
A-120-1-8	-1.44	4313	-0.37	4298	-0.11	4231	B-90-2-12	-5.51♦	165	-2.24	156	-0.54	131
A-120-1-12	-0.77	4310	0.18	4285	0.62	4154	B-120-2-4	-7.89♦	1513	-4.39♠	1512	-4.20►	1509
A-150-1-4	-0.10	71	-0.08	69	0.00	40	B-120-2-8	-2.81	4315	-0.41	4309	-0.46	4274
A-150-1-8	-0.82	2811	-0.11	2793	0.41	2695	B-120-2-12	-4.48♦	4311	-2.70	4309	-0.25	4230
A-150-1-12	-1.19	4306	-0.29	4286	0.60	4102	B-150-2-4	-0.48	1948	-0.54	1946	-0.12	1933
A-60-2-4	-0.32	-1	0.00	-4	0.00	-13	B-150-2-8	26.24	4314	26.97	4307	27.79	4269
A-60-2-8	-0.60	-3	-0.25	-4	-0.06	-28	B-150-2-12	-0.09	4309	3.52	4302	4.46	4249
A-60-2-12	-0.14	-4	-0.06	-12	-0.01	-33	**Ave.B**	2.04	1986	3.83	1981	4.35	1945
A-90-2-4	-0.42	-1	-0.05	-3	0.00	-12	**Ave.A &B**	0.54	1627	1.62	1615	1.96	1555
A-90-2-8	-1.27	1	-0.21	-4	-0.05	-25							

Figure 25 shows boxplots for the following distributions: (a) the optimality gap, computed using the best-known lower bound and reported as a percentage, and (b) the computation time, expressed in minutes. These distributions were obtained using the solving approaches of Figure 12, as described on the x-axis of Figure 25. In Figure 25b, the upper limit of the y-axis is restricted, because the computation time of CPLEX with the Powerful PC (3 d) is significantly longer than the others. In each boxplot of Figure 25, the whiskers represent the minimum and maximum values for the corresponding distribution, each box shows an interquartile range, each tick black line indicates a median value, and the circles are the outliers.

The boxplots shown in Figure 25 confirm that the A-GA computes high-quality solutions, as the optimality gaps of the selected A-GA configurations are comparable to those obtained by CPLEX with the Powerful PC and a maximum computation time of 3 d. Specifically, the A-GA with P_s=250 and M_G=250 had the optimality gap distribution with a smaller maximum value than the best one obtained by CPLEX. However, the case of P_s=250 and M_G=250 requires a significant computation time, which sometimes slightly exceeds



(a) Optimality gap (%) (b) Computation time (min)

Figure 25 **Boxplots for the distributions of the optimality gap (%) and computation time (min)**

3 h. The other A-GA configurations have slightly worse optimality gaps than the A-GA with P_s=250 and M_G=250, even if they are the fastest configurations to compute high-quality solutions.

Credit Author Statement

Lianhua Tang: Conceptualization, Model analysis, Methodology, Software, Writing - original draft, Writing - review & editing;

Andrea D'Ariano: Methodology, Validation, Supervision, Writing - original draft, Writing - review & editing;

Xingfang Xu: Conceptualization, Supervision, Funding acquisition;

Yantong Li: Model analysis, Writing - original draft, Writing - review & editing;

Xiaobing Ding: Data curation, Investigation;

Marcella Samà: Statistical analysis, Writing - review & editing.