Transportation Research Part C 60 (2015) 540–565

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Optimal timetable development for community shuttle network with metro stations

Jie Xiong [a,1], Zhengbing He [a,2], Wei Guan [a,*], Bin Ran [b,c,3]

[a] Ministry of Education Key Laboratory for Urban Transportation Complex Systems Theory and Technology, Beijing Jiaotong University, No. 3 Shang Yuan Cun, Beijing 100044, China

[b] Wisconsin ITS Program, Department of Civil & Environmental Engineering, University of Wisconsin at Madison, 1415 Engineering Drive, Madison 53706, WI, United States

[c] School of Transportation, Southeast University, No. 2 Si Pai Lou, Nanjing 210096, China

A R T I C L E I N F O

Article history:
Received 29 October 2014
Received in revised form 29 August 2015
Accepted 9 October 2015

Keywords:
Timetable development
Community shuttle
Metro
Fleet size
Vehicle capacity

A B S T R A C T

This paper investigates an issue for optimizing synchronized timetable for community shuttles linked with metro service. Considering a passenger arrival distribution, the problem is formulated to optimize timetables for multiple community shuttle routes, with the objective of minimizing passenger's schedule delay cost and transfer cost. Two constraints, i.e., vehicle capacity and fleet size, are modeled in this paper. The first constraint is treated as soft, and the latter one is handled by a proposed timetable generating method. Two algorithms are employed to solve the problem, i.e., a genetic algorithm (GA) and a Frank–Wolfe algorithm combined with a heuristic algorithm of shifting departure times (FW-SDT). FW-SDT is an algorithm specially designed for this problem. The simulated and real-life examples confirm the feasibility of the two algorithms, and demonstrate that FW-SDT outperforms GA in both accuracy and effectiveness.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, transfers between ground buses and metros are very common for a journey from a community in a suburban area to downtown. Clearly, it is very necessary to set up an efficient and robust transit microcirculation system serving these communities to the metro stations nearby. Generally speaking, the transit planning process includes four basic activities, generally performed in such a sequence: route and network design, timetable development, vehicle scheduling and crew scheduling (Ceder and Wilson, 1986; Ceder, 2001, 2002). Among these activities, the timetable development is the simplest and most feasible way in the transit pre-planning process. It only needs the operator to preset or adjust the departure times of some trips, and then an optimal effect may be achieved. This paper thus presents an optimal timetable development problem and its corresponding solutions on a combined bus-metro network.

For timetable development problems, a simple even-headway timetable, which is formulated based on an optimal headway, is often developed in the transit network design problem (e.g. Chang and Schonfeld, 1991; Chien, 2000; Chien et al., 2001a, 2001b; Chien, 2005; Chien et al., 2007; Zhao and Zeng, 2007; Szeto and Wu, 2011). The models developed in these

* Corresponding author. Tel.: +86 139 1057 0120.
E-mail addresses: 10114191@bjtu.edu.cn (J. Xiong), he.zb@hotmail.com (Z. He), weig@bjtu.edu.cn (W. Guan), bran@wisc.edu (B. Ran).
[1] Tel.: +86 159 1066 7682.
[2] Tel.: +86 182 1096 7110.
[3] Tel.: +1 608 262 0052.

http://dx.doi.org/10.1016/j.trc.2015.10.007
0968-090X/© 2015 Elsevier Ltd. All rights reserved.

studies usually minimize the total cost (i.e. user and supplier costs) to develop routes and service frequencies leading to the bus timetables. Majority of the related problems are based on a single-mode transit system without considering the coordination among multi-mode transit. Ceder (2009, 2013) proposed an idea of designing an integrated smart feeder/shuttle service, and developed a simulation tool validated by ten different routing strategies. These studies focused on the route and network design of shuttles, while few focused on the coordinated timetable development. Mohaymany and Gholami (2010) and Xiong et al. (2013) designed feeders of rail transit, which include network/routing design and headway determination for a multi-mode transit network. Yet, the transfer cost was not considered in their researches. Some studies (e.g. Quadrifoglio et al., 2006, 2007) addressed the flex-route transit service design problem, in which the feeder of bus scheduling has been involved. Alshalalfah and Shalaby (2012) investigated the feasibility and benefits of using flex-route transit as a feeder transit service for rail stations instead of the fixed-route transit. Since these researches are not specially proposed for the timetable development, the timetables are simply formulated based on the solved optimal frequencies. It lacks the consideration of the passenger arrival rate or the coordinated scheduling problem.

Considering a passenger arrival distribution at bus stops will make the timetable development problem more complex. Ceder (1986, 2003) presented two methods to formulate timetables with smooth transitions, i.e. even average loads and even maximum load on individual vehicles. These models were mainly based on the real-time stop data while neglects the passenger's waiting cost or fleet size constraint. Newell (1971) assumed a given passenger arrival rate as a smooth function of time, and analytically showed that the frequency of transit vehicles and the number of passengers served per vehicle varied with time approximately as the square root of the passenger arrival rate. Osuna and Newell (1972) developed a control strategy that either held back a transit vehicle or dispatched it immediately with the objective of minimizing the passenger's waiting time. The findings and dispatching policy were examined and extended by Wirasinghe (1990, 2003). Palma and Lindsey (2001) proposed a method to design an optimal timetable for a given number of trips with the objective of minimizing the total schedule delay based on each rider's preferred travel time. The optimal analytical solutions were presented when the rider's arrival was uniformly distributed over time. Peeters and Kroon (2001) employed a mixed integer non-linear programming formulation to optimize a cyclic railway timetable. The objective is to minimize passenger's travel time, to maximize the robustness of the timetable, and to minimize the number of trains. Gallo and Di-Miele (2001) introduced a model for the special case of dispatching buses from a parking depot to minimize the shunting cost. However, these studies were mostly based on an idealized transit system in which only a single line is included and no consideration of realistic operational factors such as vehicle capacity and transferring among multi-mode transits.

When incorporating a multi-mode transit network into the timetable development problem, factors such as transfer time needs to be considered. Shrivastava and O'Mahony (2006) developed a new model combining the network design problem with the coordinated schedule problem. An optimized feeder bus network and a coordinated timetable of each route were determined simultaneously. The results of a case study indicated that the overall loading factor was improved considerably compared to the former model presented by Shrivastava and Dhingra (2001). Similar research can be seen in Shrivastava and Dhingra (2002) and Shrivastava and O'Mahony (2009). However, only even headways were considered and determined by solving optimal service frequencies of feeder buses. They might fail to meet dynamic temporal passenger demands on a microscopic level. Chakroborty et al. (1995) formulated an optimal scheduling problem for the urban transit network based on some certain passenger arrival rates at stops. The model considered both passenger's initial waiting cost and transfer cost, but only one transfer stop was involved in the problem formulation. Moreover, the model assumed vehicles could be loaded unlimitedly and the fleet size constraint was not considered. Both Ceder et al. (2001) and Wong et al. (2008) presented a mixed-integer-programming optimization model to formulate synchronized timetables. The former one (Ceder et al., 2001) aimed at maximizing the number of simultaneous arrivals on a bus network, while the latter (Wong et al., 2008) presented the problem to minimize the overall transfer waiting time based on a rail mass network. Kwan and Chang (2008) developed a model to formulate a new measurement for timetable synchronization by means of a total passenger dissatisfaction index and a total deviation index. To solve the model, a multi-objective evolutionary algorithms (MOEA) was combined with local search techniques. In addition, to seek for a flexible scheduling and control tool for the urban transit, Adamski (1998) and Dessouky et al. (1999) addressed a dynamic optimal dispatching problem at transfer terminals. Numerical simulations were used by both of them to validate the effectiveness of their proposed dynamic schedule control strategies.

This paper develops a synchronized timetable for shuttles linked with metro service based on certain passenger arrival distributions at stops. To our best knowledge, very few studies developed synchronized timetables based on temporal varying passenger demand and simultaneously considered both vehicle capacity and fleet size constraint. Neglecting these factors may make the problem simpler but unrealistic. Thus, it is more realistic to incorporate them into the problem.

A robust service provided by community shuttles linked with metros should satisfy a series of principles, e.g. meeting passenger's demands, reducing passenger's waiting time, transfer time, and avoiding overcrowding as much as possible. To this end, this paper thus seeks for an optimal synchronized timetable that minimizes the total cost of passenger's schedule delay cost and transfer cost. Two constraints are considered, i.e., vehicle capacity and fleet size, distinguishes our problem from the existing synchronized timetable development problems in the literature. In formulating the problem, the first constraint is modeled as a soft constraint, and the second one is handled by a proposed timetable generating method. To solve the problem, a GA is employed, and then a Frank–Wolfe algorithm combined with a heuristic algorithm of shifting departure times (FW-SDT) is proposed specifically. Two algorithms are compared in simulated and real-life examples. The results show the feasibility of the two algorithms, and demonstrate that FW-SDT performs better than GA in both accuracy and effectiveness.

The remainder of the paper is organized as follows: In Section 2, the problem is defined and a cost model with two realistic operational constraints, i.e. vehicle capacity and fleet size, is formulated. In Section 3, a method to generate timetables under a given fleet size is addressed. In Section 4, GA is employed to solve the problem. In Section 5, a new solution algorithm, i.e., FW-SDT is proposed for the defined problem. In Section 6, the performance of the algorithms is demonstrated in a simulated test network with results and numerical analysis, and followed by a real life example. Finally, in Section 7, conclusions and directions for future research are given.

2. Model formulation

The problem is based on a bus-metro combined network that consists of shuttle routes and metro stations, e.g. the network shown in Fig. 1. Assumptions are given as follows: (1) Every OD trip starts at a bus stop, and terminates at a metro station. (2) The travel time on each segment is constant. (3) For one route, all the shuttles are dispatched from one terminal (called outer terminal). Shuttles only dwell at the stops/stations where passengers boarding or alighting. The dwell time is independent of the number of boarding or alighting passengers. (4) For simplicity, passengers' walking time for transferring is set to be 0. (5) For the OD pairs that can be carried out by two or more paths, the OD demand assignment for each path is affected by two aspects: the travel time (including in-vehicle travel time, out-of-vehicle waiting time and transfer time) and the travel cost. The specific ratio is determined by a logistic function, which will be given in Section 2.1.

We explain assumption (1) as follows. For a route serving large communities in suburbs, the loading in both directions are usually significantly different, particularly during peak hours. That is, during the morning peak hours, passengers usually depart from bus stops and travel to downtown via metro stations, while conversely in the evening peak hours. Therefore, we treat that all the passengers' trips originate/terminate at the bus stops and terminate/originate at the metro stations.

For assumption (2), the travel time is assumed to be reliable, since this study only considers transit planning instead of operating. Moreover, the community traffic is usually more reliable than the urban traffic due to the simpler network topology and lower traffic demand. Therefore, in assumption (5), the OD assignment is only determined by the passenger's average travel time and the travel cost of the path, and the travel time variability is not considered.

2.1. OD assignment

A path-based assignment strategy is used for the OD demand assignment in this study. For an OD pair, a number of paths can be generated based on the shuttle network by using the Depth-first Search (DFS) algorithm if transferring among different routes is neglected. For detailed steps, one can refer to Xiong et al. (2013). However, many of the generated paths are infeasible in actual cases because of too many bypass or excessive transfers. Therefore, two constraints are considered here to eliminate the infeasible paths: (1) For an OD pair, the travel time of a path should be less than that of the path with the least travel time multiplied by $\rho(\rho > 1)$. (2) A feasible path involves at most two shuttle routes.

Constraint (1) eliminates the paths with excessive bypass. Constraint (2) means that there exists at most one transferring among different routes for the passengers who want to take the shuttle to the metro station. This assumption is reasonable because for a community, few passengers would like to transfer twice or more among shuttles to arrive at the metro service and then take the train to complete their journey. Assuming that there are k feasible paths for an OD pair, the loading proportion of the ith path ($1 \leqslant i \leqslant k$) is denoted as p_i^{OD}. Based on the above two constraints, the following logistic function is given as follows:

$$p_i^{OD} = \frac{e^{-V_i}}{\sum_{i=1}^{k} e^{-V_i}} \tag{1}$$

where V_i is the general cost of the ith path, which is formulated as:

$$V_i = a_1 \sum_j \frac{1}{n_{ij}} + a_2 t_i + a_3 \sum_j c_{ij} \tag{2}$$

where n_{ij} is the number of dispatched vehicle trips of the jth ($j \leqslant 2$) route involved in the ith path during the studied period. Specially, for $j = 1$, $1/n_{ij}$ indicates the average waiting time on the first route; for $j = 2$, $1/n_{ij}$ is treated as the average transfer delay between the two routes. t_i is the in-vehicle travel time of the ith path; c_{ij} is the travel cost (i.e. ticket price) of the jth route involved in the ith path. a_1, a_2, a_3 are non-negative coefficients denoting the weights of the above three items, respectively. Eqs. (1) and (2) show that the loading proportion has negative correlation to the passenger's travel time (including in-vehicle travel time, waiting time of the first vehicle, and transfer delay) and travel cost.

Considering the transferring in the network, there exist boarding or alighting demands in certain stops. Assuming that the segment from stop/station m to stop/station d is a segment of route l, the cumulative passenger arrival distribution for the segment, $f_{md}^{l}(t)$, is formulated as:

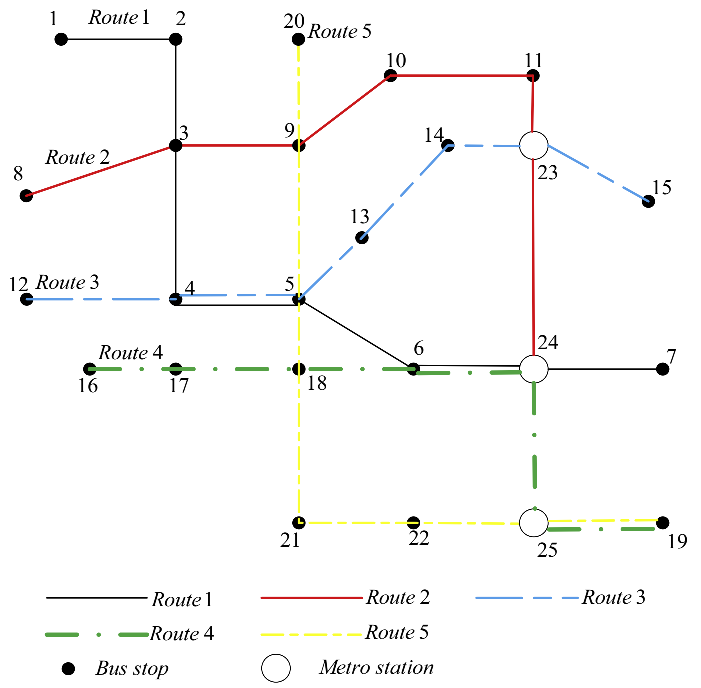

Fig. 1. An example of studied shuttle–metro combined network.

$$f_{md}^{l}(t) = \sum_{i}\varphi_{md}(t)\cdot q_{md}^{i,l}\cdot p_{i}^{md} + \sum_{d_1 \neq d}\sum_{i}\varphi_{md_1}(t)\cdot q_{md_1}^{i,l}\cdot p_{i}^{md_1} + \sum_{m_1 \neq m}\sum_{i}\varphi_{m_1 d}(t - t_{m_1}^{m})\cdot q_{m_1 d}^{i,l}\cdot p_{i}^{m_1 d} \tag{3}$$

where $\varphi_{md}(t)$ is the original travel demand from m to d (only when m is a shuttle stop and d is a metro station, $\varphi_{md}(t)$ is meaningful, otherwise, $\varphi_{md}(t) = 0$); $q_{md}^{i,l}$ takes 0 or 1. If the segment md of route l is a component of the ith path for the demand from m to d, $q_{md}^{i,l} = 1$, otherwise, its value is 0. $t_{m_1}^{m}$ is the travel time from m_1 to m. From Eq. (3), it is clear to see that the arrival distribution of the segment can be divided into three terms: the first one is the distribution of the travel demand from m to d; the second one is the distribution of the demand from m to $d_1 (d_1 \neq d)$ transferring at d; the third one is the distribution of the demand from $m_1 (m_1 \neq m)$ to d transferring at m. For the third term, the segment of route l is the latter part of the ith path. Thus, a phase of travel time, $t_{m_1}^{m}$, needs to be added in the distribution formulation.

2.2. Objective function

As stated above, the model considers the fleet size constraint, which reflects the limited vehicle resources in practice. Therefore, for the operator, its supply cost is fixed and thus is not needed to be included in the objective function. On the other hand, all the passengers' in-vehicle cost is also fixed given the OD assignment for a given network. The objective function thus only includes the following passenger's schedule delay cost and the transfer cost:

$$C_T = \sum_{l=1}^{L} C_D^l + \sum_{l=1}^{L} C_{TR}^l \tag{4}$$

where C_T is the total cost of the network; C_D^l is the passenger schedule delay cost on route l; C_{TR}^l is the passenger transfer cost of route l; L is the number of shuttle routes on the network.

The schedule delay cost in this paper employs the formulation given in Palma and Lindsey (2001). The shuttle system operates n vehicles at times $T_1, \ldots, T_j, \ldots, T_n$, and each passenger must travel at T_j for some j. Each passenger is assumed to choose the vehicle that minimizes his/her delay cost. Before calculating the schedule delay cost, some time boundaries, $t_{j,j+1}$, $j = 0, 1, \ldots, n$, need to be identified first during the studied time span $[0, LN]$. For the passengers who want to depart at $t_{j,j+1}$, there is no difference for them to choose the shuttle departing at T_j or T_{j+1}. Therefore, $t_{j,j+1}$ can be formulated as:

$$t_{j,j+1} = \begin{cases} (\beta T_j + \gamma T_{j+1})/(\beta + \gamma) & \text{for } j = 1, \ldots, n-1 \\ 0 & \text{for } j = 0 \\ LN & \text{for } j = n \end{cases} \tag{5}$$

where $\beta,\ \gamma$-the delay cost per minute of arriving early or late. For the bus stop i, its individuals' arrival distribution is denoted as $f_i(t)$, which is a continuous distribution fitted using the historical passenger arrival data. It also denotes the preferred travel time of individuals during $[0, LN]$ at stop i. For a vehicle departing at T_j, it carries the individuals whose preferred travel time is between $t_{j-1,j}$ and $t_{j,j+1}$, and their schedule delay cost at stop i can be written as follows:

$$C_D(T_j) = \gamma \int_{t_{j-1,j}}^{T_j} (T_j - t) f_i(t)dt + \beta \int_{T_j}^{t_{j,j+1}} (t - T_j) f_i(t)dt \tag{6}$$

In this problem, for a route l with N^l vehicle trips, M bus stops and D metro stations, the timetable at the terminal is represented as: $\boldsymbol{T}^l = (T_1^l, T_2^l, \ldots, T_{N^l}^l)$, the corresponding $\boldsymbol{t}^l = (0, t_{12}^l, t_{23}^l, \ldots, t_{N^l-1,N^l}^l, LN)$. Denote by M_1 and M_2 the number of stops/stations with boarding or alighting demands, respectively. Therefore, the travel time spent from the terminal to the stops/stations existing boarding demands is denoted as: $\boldsymbol{s}^l = (s_1^l, s_2^l, \ldots, s_{M_1}^l)$. The schedule delay cost of route l is:

$$\begin{aligned} C_D^l\left(\boldsymbol{T}^l\right) = & \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\gamma \int_{s_m^l}^{T_1^l+s_m^l} (T_1^l + s_m^l - t) f_{md}^l(t)dt + \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\beta \int_{T_1^l+s_m^l}^{t_{12}^l+s_m^l} (t - T_1^l - s_m^l) f_{md}^l(t)dt + \cdots \\ & + \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\gamma \int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} (T_n^l + s_m^l - t) f_{md}^l(t)dt + \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\beta \int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} (t - T_n^l - s_m^l) f_{md}^l(t)dt + \cdots \\ & + \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\gamma \int_{t_{N^l-1,N^l}^l+s_m^l}^{T_{N^l}^l+s_m^l} (T_{N^l}^l + s_m^l - t) f_{md}^l(t)dt + \sum_{m=1}^{M_1}\sum_{d=1}^{M_2}\beta \int_{T_{N^l}^l+s_m^l}^{LN++s_m^l} (t - T_{N^l}^l - s_m^l) f_{md}^l(t)dt \end{aligned} \tag{7}$$

For simplicity, passenger's walking time for transferring is set to be 0, and the transfer delay is the difference between the arrival times of a bus and the corresponding train at a metro station. We use $\boldsymbol{sd}^l = (sd_1^l, sd_2^l, \ldots, sd_{D_1}^l)$ to represent the travel time spent from the terminal to every station, where D_1 denotes the number of metro stations for a shuttle needing to stop during the round trip. TR_k^d denotes the arrival time of the kth train at station d. Thus for route l, the transfer cost is:

$$\begin{aligned} C_{TR}^l\left(\boldsymbol{T}^l\right) = & \sum_{d=1}^{D_1}\sum_{m=1}^{M_d}\sum_{k}\delta_{k,1}^{d,l}\left(TR_k^d - T_1^l - sd_d^l\right)\int_{s_m^l}^{t_{12}^l+s_m^l} f_{md}^l(t)dt + \cdots + \sum_{d=1}^{D_1}\sum_{m=1}^{M_d}\sum_{k}\delta_{k,n}^{d,l}\left(TR_k^d - T_n^l - sd_d^l\right)\int_{t_{n-1,n}^l+s_m^l}^{t_{n,n+1}^l+s_m^l} f_{md}^l(t)dt \\ & + \cdots + \sum_{d=1}^{D_1}\sum_{m=1}^{M_d}\sum_{k}\delta_{k,N^l}^{d,l}\left(TR_k^d - T_N^l - sd_d^l\right)\int_{t_{N^l-1,N^l}^l+s_m^l}^{LN+s_m^l} f_{md}^l(t)dt \end{aligned} \tag{8}$$

M_d is the number of stops existing passengers to station d; $\delta_{k,n}^{d,l}$ takes 0 or 1. If TR_k^d is the arrival time of the first train after $T_n^l + sd_d^l$, $\delta_{k,n}^{d,l} = 1$, otherwise, its value is 0.

Since there are M stops and D stations on route l, the round trip can be divided into $2 \times (M + D - 1)$ segments. The segment loading of the nth vehicle trip on the segment $r(1 \leqslant r \leqslant 2 \times (M + D - 1))$, $Ld_{r,n}^l$, can be written as:

$$Ld_{r,n}^l = \sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\int_{t_{n-1,n}^l+s_r^l}^{t_{n,n+1}^l+s_r^l} f_{md}^l(t)dt - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\int_{t_{n-1,n}^l+s_r^l}^{t_{n,n+1}^l+s_r^l} f_{md}(t)dt \tag{9}$$

where M_1^r is the number of stops/stations with boarding demands before segment r, M_2^r is the number of stops/stations with alighting demands before segment r, s_r^l is the travel time from the terminal to segment r.

2.3. Constraints

The constraints of the problem are given as follows:

$$Ld_{r,n}^l \leqslant P \quad \forall r, n, l \tag{10}$$

$$0 \leqslant T_1^l < \cdots < T_n^l < T_{n+1}^l \cdots < T_{N^l}^l \leqslant LN \quad \forall n, l \tag{11}$$

$$T_{n-F}^l + rt^l \leqslant T_n^l \quad \forall n \in (F^l, N^l], l \quad \text{if } N^l > F^l \tag{12}$$

$$T_n^l \leqslant T_{n+F}^l - rt^l \quad \forall n \in (0, N^l - F^l], l \quad \text{if } N^l > F^l \tag{13}$$

$$\sum_k \delta_{k,n}^{d,l} = 1 \quad \forall n, d, l \tag{14}$$

$$(TR_k^d - T_n^l - sd_d^l) + W(1 - \delta_{k,n}^{d,l}) \geqslant 0 \quad \forall k, n, d, l \tag{15}$$

$$\begin{cases} (TR_k^d - T_n^l - sd_d^l)\delta_{k,n}^{d,l} \leqslant TR_k^d - TR_{k-1}^d & \forall k > 1, n, d, l \\ (TR_k^d - T_n^l - sd_d^l)\delta_{k,n}^{d,l} \leqslant TR_k^d & k = 1, \forall n, d, l \end{cases} \tag{16}$$

Constraint (10) is the vehicle capacity constraint, and P is the vehicle rated load. Constraint (11) restricts the departure order. Constraints (12) and (13) are fleet size constraints, and F^l is the fleet size of route l; rt^l is the round trip time of route l. Only when the number of vehicle trips N^l is more than the fleet size F^l, these two constraints take effects. Constraint (14)–(16) are transfer constraints. Constraint (14) ensures that there is only one train joint for a vehicle trip. Constraint (15) forces the $\delta_{k,n}^{d,l}$ of the trains arrive earlier than $T_n^l + sd_d^l$ to value 0. (W is a large positive number); Constraint (16) forces the $\delta_{k,n}^{d,l}$ of the trains arrive far behind of $T_n^l + sd_d^l$ to value 0. Thus constraint (14)–(16) together assure that if and only if TR_k^d is the arrival time of the first train after $T_n^l + sd_d^l$, $\delta_{k,n}^{d,l} = 1$.

The intensity of the constraint (10) is largely affected by the input of passenger data, and it is difficult to estimate without an enormous amount of computing. It means that we may do many unnecessary calculations and abandon many solutions if it is a strong constraint. To relieve the computational burden, we treat the vehicle capacity constraint as a soft constraint. The amount of violation of soft constraints is usually penalized and added to the objective function (Ibaraki et al., 2005). Here we add an overloading penalty to the original objective function. Before solving this penalty, we define a penalty function of the vehicle capacity, $g(x)$:

$$g(x) = \omega(\alpha^x - 1) \tag{17}$$

where x is the segment loading, and ω and α are coefficients. By adjusting ω and α, we can make $g(x)$ close to 0 when $0 \leqslant x \leqslant P$, and increase sharply when $x > P$. Based on Eq. (9), the overloading penalty on segment r of route l is:

$$\begin{aligned} C_P^{l,r}\left(\boldsymbol{T}^l\right) &= \sum_{n=1}^{N^l} g\left(Ld_{r,n}^l\right) \\ &= g\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\int_{s_r^l}^{t_{12}^l+s_r^l} f_{md}(t)dt - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\int_{s_r^l}^{t_{12}^l+s_r^l} f_{md}(t)dt\right) + \cdots \\ &\quad + g\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\int_{t_{n-1,n}^l+s_r^l}^{t_{n,n+1}^l+s_r^l} f_{md}(t)dt - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\int_{t_{n-1,n}^l+s_r^l}^{t_{n,n+1}^l+s_r^l} f_{md}(t)dt\right) + \cdots \\ &\quad + g\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\int_{t_{N^l-1,N^l}^l+s_r^l}^{LN+s_r^l} f_{md}(t)dt - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\int_{t_{N^l-1,N^l}^l+s_r^l}^{LN+s_r^l} f_{md}(t)dt\right) \end{aligned} \tag{18}$$

The penalty of each segment on route l is denoted by: $\boldsymbol{C_P^{l,R}} = \left(C_P^{l,1}, \ldots, C_P^{l,r}, \ldots, C_P^{l,2\times(M+D-1)}\right)$. For route l with $2\times(M+D-1)$ segments, the distance of each segment is denoted by: $\boldsymbol{dis}^l = \left(dis^{l,1}, \ldots, dis^{l,r}, \ldots, dis^{l,2\times(M+D-1)}\right)$. The overloading penalty of route l, C_P^l, is:

$$C_P^l\left(\boldsymbol{T}^l\right) = \boldsymbol{C_P^{l,R}} \cdot \left(\boldsymbol{dis^l}\right)^{\boldsymbol{T}} \tag{19}$$

Adding C_P^l into the original objective function, the problem becomes a multi-objective problem. The new objective function is:

$$C_T = \mu_1 \sum_{l=1}^{L} C_D^l + \mu_2 \sum_{l=1}^{L} C_{TR}^l + \mu_3 \sum_{l=1}^{L} C_P^l \tag{20}$$

where $\mu_1,\ \mu_2,\ \mu_3$ are weight coefficients. The constraints of the problem are Eqs. (11)–(16).

3. Generating timetables

For route l, we denote by N^l the number of vehicle trips, by F^l the fleet size, by rt^l the round trip time, and $[0, LN]$ denotes the study period. $\boldsymbol{T}^l$ is the timetable needing to be generated. The steps of generating the terminal timetable under the fleet size constraint (i.e. Eqs. (11)–(13)) are given as follows:

Step 1: Input N^l, F^l, rt^l, LN; set $\boldsymbol{T}^l = \phi$, go to step 2.

Step 2: Set $R^l = \lceil N^l / F^l \rceil$, which represents the number of loops for the vehicles to complete all the vehicle trips; set $N_e^l = N^l - (R^l - 1) \cdot F^l$, which represents the number of vehicle trips in the last loop. Initialize $t_1 = 0,\ i = 1$. If $R^l > 1$, go to step 3; otherwise, go to step 7.
Step 3: If $i > R^l - 1$, go to step 7; otherwise, initialize $j = 1$, and go to step 4.
Step 4: If $j > F^l$, set $i = i + 1$, and go to Step 3; otherwise, if $j \leqslant N_e^l$, set $t_2 = LN - rt^l \cdot (R^l - i)$; if $N_e^l < j \leqslant F^l$, set $t_2 = LN - rt^l \cdot (R^l - i - 1)$. Set $j = j + 1$, and go to step 5.
Step 5: Set $t = t_1 + rand(1) \cdot (t_2 - t_1)$, and update $\boldsymbol{T}^l = [\boldsymbol{T}^l, t]$. Go to step 6.
Step 6: Update t_1. If $i = 1$ and $j < F^l$, set $t_1 = \boldsymbol{T}^l(end)$; if $i = 1$ and $j = F^l$, set $t_1 = \max\left\{\boldsymbol{T}^l(1) + rt^l, \boldsymbol{T}^l(end)\right\}$; if $i > 1$, set $t_1 = \max\left\{\boldsymbol{T}^l\left((i-2) \cdot F^l + j\right) + rt^l, \boldsymbol{T}^l(end)\right\}$. Go to step 4.
Step 7: If $N_e^l > 0$, initialize $j = 1$, and go to step 8; otherwise, the algorithm ends, and output $\boldsymbol{T}^l$.
Step 8: If $j \leqslant N_e^l$, set $t_2 = LN$, and go to step 9; otherwise, the algorithm ends, and output $\boldsymbol{T}^l$.
Step 9: If $R^l = 1$, set $t_1 = 0$; otherwise, set $t_1 = \max\left\{\boldsymbol{T}^l\left((i-2) \cdot F^l + j\right) + rt^l, \boldsymbol{T}^l(end)\right\}$. Set $t = t_1 + rand(1) \cdot (t_2 - t_1)$, and update $\boldsymbol{T}^l = [\boldsymbol{T}^l, t],\ j = j + 1$. Go to step 8.

Let t_1 and t_2 denote the lower and the upper bounds of the current departure time respectively. The main idea of the method is to first update the bounds of the current departure time according to Eqs. (11)–(13) based on the former deterministic departure times, and then to generate the current departure time randomly in the interval $[t_1, t_2]$. By repeating this process, all the departure times will be generated sequentially.

By using the above algorithm, the terminal timetables of all the routes $l (1 \leqslant l \leqslant L)$ will be generated. Then a set that stores the timetables, $\boldsymbol{T} = \left(\boldsymbol{T}^1, \ldots, \boldsymbol{T}^l, \ldots, \boldsymbol{T}^L\right)$, is formulated.

After generating a feasible timetable at the terminal, the travel time matrix s^l needs to be solved first. Based on s^l, the terminal timetable can be extended to formulate the timetables for the other stops/stations afterwards. With the consideration of transferring among different shuttle routes, shuttles will dwell not only at stations but also at stops existing transferring demands. Therefore, before solving s^l, the stops/stations dwelled by a vehicle during its round trip should be listed sequentially. Then the timetables at the other stops/stations on route l are formulated based on s^l. They are denoted by: $\boldsymbol{TN}^l = \left((\boldsymbol{T}^l)^T + s^l(1), \ldots, (\boldsymbol{T}^l)^T + s^l(m), \ldots, (\boldsymbol{T}^l)^T + s^l(end)\right)$. The matrix that represents the timetables for all the routes is denoted by: $\boldsymbol{TN} = \left(\boldsymbol{TN}^1, \ldots, \boldsymbol{TN}^l, \ldots, \boldsymbol{TN}^L\right)$.

4. Genetic algorithm

Genetic algorithms are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection (Shrivastava and Dhingra, 2002). It mainly includes four operators in each generation: generating population, selection, crossover and mutation. The computation continues until the stopping criterion is satisfied. The algorithm reserves the best individual of each generation and obtains the optimal one from them.

4.1. Fitness function and selection

The GA designed for this problem employs a decimal encoding to represent an individual. In the generating population process, any randomly generated $\boldsymbol{T}$ is an individual. Its fitness value f can be set as the reciprocal of the objective value $f(\boldsymbol{T}) = 1/C_T(\boldsymbol{T})$. In the selection process, roulette is employed to select the individuals according to their fitness values. For any individual $\boldsymbol{T_i}$ with the fitness value f_i, the probability to be selected is $P_i = f_i / \sum f_i (1 \leqslant i \leqslant NIND)$, where $NIND$ is the population size. In each generation, a number of $NIND' = NIND \cdot GGAP$ individuals are selected for the following crossover and mutation, where $GGAP$ is the gap between the number of selected individuals and the population size.

4.2. Crossover

Crossover is the main operator in GA. For a set of individuals $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}$, the steps of crossover are given as follows:

Step 1: Input $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}, P_c$ (i.e. the crossover rate). Initialize $i = 1$. Go to step 2.
Step 2: If $i > NIND'$, the process of the crossover operator ends, and output the individual set $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}$; otherwise, go to step 3.
Step 3: Generate a number between 0 and 1 randomly: $rand(1) \in (0, 1)$. If $rand(1) > P_c$, set $i = i + 1$, and go to step 2; otherwise, go to step 4.

Step 4: Set $cp = \lceil rand(1,2) \cdot NIND' \rceil$, which denotes the indexes of the two chosen individuals. If $cp(1) \neq cp(2)$, go to step 5; otherwise, return to step 4.

Step 5: Crossover. Assume that $\{\boldsymbol{T}_i, \boldsymbol{T}_j\}$ are the two chosen individuals, and the new individuals are obtained by using Eq. (21).

$$\begin{cases} \boldsymbol{T}_i' = r_a \boldsymbol{T}_i + (1 - r_a)\boldsymbol{T}_j \\ \boldsymbol{T}_j' = (1 - r_a)\boldsymbol{T}_i + r_a \boldsymbol{T}_j \end{cases} \tag{21}$$

where r_a is a random number between 0 and 1. Go to Step 6.

Step 6: Replace $\boldsymbol{T_i}, \boldsymbol{T_j}$ with $\boldsymbol{T_i'}, \boldsymbol{T_j'}$. Update $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}$, and set $i = i + 1$. Go to step 2.

We conclude that both the new individuals $\boldsymbol{T_i'}, \boldsymbol{T_j'}$ obtained by Eq. (21) are feasible solutions for this problem (i.e. satisfying Eqs. (11)–(13)). This is because all these constraints are linear functions, and the set of feasible solutions is a convex set.

4.3. Mutation

Mutation is the main mechanism to maintain population diversity in GA. For a set of $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}$, the steps of mutation are given as follows:

Step 1: Input $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}, P_m$(i.e. the mutation rate). Initialize $i = 1$. Go to step 2.

Step 2: If $i > NIND'$, the process of the mutation operator ends, and output $\{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_i, \ldots, \boldsymbol{T}_N\}$; otherwise, go to step 3.

Step 3: Generate a number between 0 and 1 randomly, $rand(1) \in (0, 1)$. If $rand(1) > P_m$, set $i = i + 1$, and go to step 2; otherwise, set $cp1 = \lceil rand(1) \cdot NIND' \rceil$, which denotes the index of the chosen individual, assuming the individual is $\boldsymbol{T}_i$; set $cp2 = \left\lceil rand(1, length(\boldsymbol{T}_i)). \times \left(length(\boldsymbol{T}_i^1), \ldots, length(\boldsymbol{T}_i^l), \ldots, length(\boldsymbol{T}_i^L)\right)\right\rceil$, which is a matrix denoting the index of the chosen departure time for each route. Denote by $T_i^l(j)$ (i.e. the jth departure time of the ith individual for route l) the chosen departure time, and then go to step 4.

Step 4: Calculate the lower and upper bounds of $T_i^l(j)$, which are denoted by $T_i^l(j)_{min}$ and $T_i^l(j)_{max}$. Let $r_b = rand(1)$, and then calculate $T_i^l(j)'$ through Eq. (22). Go to Step 5.

$$T_i^l(j)' = \begin{cases} T_i^l(j) - \left(T_i^l(j) - T_i^l(j)_{\min}\right) \cdot r_b \cdot (1 - gen/MAXGEN)^2 & \text{for } r_b \leqslant 1/2 \\ T_i^l(j) + \left(T_i^l(j)_{\max} - T_i^l(j)\right) \cdot r_b \cdot (1 - gen/MAXGEN)^2 & \text{for } r_b > 1/2 \end{cases} \tag{22}$$

where gen is the current generation number, $MAXGEN$ is the maximum generation.

Step 5: Replace $T_i^l(j)$ with $T_i^l(j)'$, and set $i = i + 1$. Go to step 2.

Finally, the new obtained individuals are added into the last population. The mixed population ($NIND + NIND'$ individuals) is sorted in a descending order according to their fitness values. The first $NIND$ individuals are chosen as the next population.

5. Frank–Wolfe algorithm combined with a heuristic algorithm of shifting departure times

In this section, a new algorithm, Frank–Wolfe algorithm combined with a heuristic algorithm of shifting departure times (FW-SDT) is proposed to solve the optimal synchronized timetable development problem. The FW-SDT is divided into two stages: the first stage is using the Frank–Wolfe algorithm to minimize the sum of the schedule delay cost and the overloading penalty. The second stage is based on the result obtained in the first stage, by shifting departure times to reduce the transfer cost, while ensuring the other two costs nearly remain unchanged.

5.1. Frank–Wolfe algorithm

The Frank–Wolfe algorithm is an iterative first-order optimization algorithm for solving nonlinear optimization problems. Its main idea is to linearly approximate the objective function constantly, and to determine a direction in the solution space by solving a linear problem. A new solution is obtained by moving a certain distance based on the last solution along the direction.

The utilization of the Frank–Wolfe algorithm has two conditions:

(1) The objective function must be a continuously differentiable function.
(2) The set of all feasible solutions must be convex.

For the transfer cost, it involves a 0–1 programming problem, which makes the objective function definitely a non-continuously differentiable function (actually it has a step when T_n^l at each $(TR_k^d - sd_d^l)$). In this stage, the transfer cost is excluded from the objective function. We denote the sum of the other two costs on route l as: $C_{DP}^l = C_D^l + C_P^l$. Based on the above two conditions, two propositions are put forward:

Proposition 1. *The schedule delay cost C_D^l is a continuously differentiable function, for $\forall f(t)$.*

Proposition 2. *If $g(x)$ is continuously differentiable, the overloading penalty C_P^l is a continuously differentiable function, for $\forall f(t)$.*

The proofs of Propositions 1 and 2 are provided in Appendices A and B, respectively. Note that the $f(t)$ stated in the above two propositions should be a continuous distribution function. A discrete distribution is not applicable in this study.

From the above two propositions, we learn that if $g(x)$ is a continuously differentiable function, $C_{DP}^l = C_D^l + C_P^l$ is continuously differentiable. It means whether C_{DP}^l is continuously differentiable depends solely on $g(x)$, while not related to the passenger arrival distribution $f(t)$. This characteristic ensures that FW-SDT is suitable for any continuous arrival distribution function, including piecewise functions.

It is worth noting that the details of assigning passengers to vehicles should be neglected in the study. For instance, passengers are likely to be assigned to the first arrival vehicle at the stop in some studies. However, this assignment will lead to a non-continuously differentiable objective function because a mixed-integer programming problem is introduced in the modeling process. Therefore, in Section 2.1, we only calculate the utility function of each path for an OD pair, and determine the loading proportion of each path by a logistic function.

As described in Section 4.2, all of the constraints (11)–(13) are linear functions, thus the set of feasible solutions is definitely a convex set. Therefore, the Frank–Wolfe algorithm can be implemented. It uses the gradient at any point of the objective function to do the linear approximation. For any $n \in (1, N^l)$, the partial derivative of C_D^l for T_n^l is given as follows:

$$
\begin{aligned}
\frac{\partial C_D^l(\boldsymbol{T}^l)}{\partial T_n^l} = {} & \beta \sum_m \sum_d \left(\left(t_{n-1,n}^l + s_m^l\right) f_{md}^l\left(t_{n-1,n}^l + s_m^l\right) \frac{\partial t_{n-1,n}^l}{\partial T_n^l} - \left(T_{n-1}^l + s_m^l\right) f_{md}^l\left(t_{n-1,n}^l + s_m^l\right) \frac{\partial t_{n-1,n}^l}{\partial T_n^l} \right) \\
& + \gamma \sum_m \sum_d \left(\int_{t_{n-1,n}^l + s_m^l}^{T_n^l + s_m^l} f_{md}^l(t)dt - \left(T_n^l + s_m^l\right) f_{md}^l\left(t_{n-1,n}^l + s_m^l\right) \frac{\partial t_{n-1,n}^l}{\partial T_n^l} + \left(t_{n-1,n}^l + s_m^l\right) f_{md}^l\left(t_{n-1,n}^l + s_m^l\right) \frac{\partial t_{n-1,n}^l}{\partial T_n^l} \right) \\
& + \beta \sum_m \sum_d \left(\left(t_{n,n+1}^l + s_m^l\right) f_{md}^l\left(t_{n,n+1}^l + s_m^l\right) \frac{\partial t_{n,n+1}^l}{\partial T_n^l} - \int_{T_n^l + s_m^l}^{t_{n,n+1}^l + s_m^l} f_{md}^l(t)dt - \left(T_n^l + s_m^l\right) f_{md}^l\left(t_{n,n+1}^l + s_m^l\right) \frac{\partial t_{n,n+1}^l}{\partial T_n^l} \right) \\
& + \gamma \sum_m \sum_d \left(\left(t_{n,n+1}^l + s_m^l\right) f_{md}^l\left(t_{n,n+1}^l + s_m^l\right) \frac{\partial t_{n,n+1}^l}{\partial T_n^l} - \left(T_{n+1}^l + s_m^l\right) f_{md}^l\left(t_{n,n+1}^l + s_m^l\right) \frac{\partial t_{n,n+1}^l}{\partial T_n^l} \right)
\end{aligned}
\tag{23}
$$

There are four terms in Eq. (23). The first one indicates the increase in early arrival costs for passengers who ride the $(n-1)$th vehicle; the second indicates the increase in late arrival costs for passengers who ride the nth vehicle; the third denotes the increase in early arrival costs for passengers who ride the nth vehicle; and the fourth denotes the increase in late arrival costs for passengers who ride the $(n+1)$th vehicle. According to Eq. (5), $\partial t_{n-1,n}^l / \partial T_n^l = \gamma/(\beta+\gamma)$, $\partial t_{n,n+1}^l / \partial T_n^l = \beta/(\beta+\gamma)$, so Eq. (23) can be simplified as:

$$
\frac{\partial C_D^l(\boldsymbol{T}^l)}{\partial T_n^l} = \gamma \sum_m \sum_d \left(\int_{t_{n-1,n}^l + s_m^l}^{T_n^l + s_m^l} f_{md}^l(t)dt \right) - \beta \sum_m \sum_d \left(\int_{T_n^l + s_m^l}^{t_{n,n+1}^l + s_m^l} f_{md}^l(t)dt \right)
\tag{24}
$$

For $n = 1$ and N^l, the above process is similar. The final formulation of $\partial C_D^l / \partial T_n^l$ is given as follows:

$$
\frac{\partial C_D^l(\boldsymbol{T}^l)}{\partial T_n^l} = \begin{cases}
\gamma \sum_m \sum_d \left(\int_{s_m^l}^{T_1^l + s_m^l} f_{md}^l(t)dt \right) - \beta \sum_m \sum_d \left(\int_{T_1^l + s_m^l}^{t_{12}^l + s_m^l} f_{md}^l(t)dt \right) & \text{for } n = 1 \\
\gamma \sum_m \sum_d \left(\int_{t_{n-1,n}^l + s_m^l}^{T_n^l + s_m^l} f_{md}^l(t)dt \right) - \beta \sum_m \sum_d \left(\int_{T_n^l + s_m^l}^{t_{n,n+1}^l + s_m^l} f_{md}^l(t)dt \right) & \text{for } 1 < n < N^l \\
\gamma \sum_m \sum_d \left(\int_{t_{N^l-1,N^l}^l + s_m^l}^{T_N^l + s_m^l} f_{md}^l(t)dt \right) - \beta \sum_m \sum_d \left(\int_{T_N^l + s_m^l}^{LN + s_m^l} f_{md}^l(t)dt \right) & \text{for } n = N^l
\end{cases}
\tag{25}
$$

For $n \in (1, N^l)$, a slight shift of T_n^l will affect the loading of the $(n-1)$th, nth and $(n+1)$th vehicle. This can be confirmed by Eq. (9), and the partial derivatives of the three vehicles' loading on segment r for T_n^l are:

$$
\frac{\partial Ld_{r,n-1}^l(\boldsymbol{T}^l)}{\partial T_n^l} = \sum_{m=1}^{M_1^r} \sum_{d=1}^{M_2} f_{md}^l(t_{n-1,n}^l + s_r^l) \frac{\partial t_{n-1,n}^l}{\partial T_n^l} - \sum_{d=1}^{M_2^r} \sum_{m=1}^{Md} f_{md}^l(t_{n-1,n}^l + s_r^l) \frac{\partial t_{n-1,n}^l}{\partial T_n^l}
\tag{26}
$$

$$\frac{\partial Ld^l_{r,n}(\boldsymbol{T}^l)}{\partial T^l_n} = \sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\partial t^l_{n,n+1}}{\partial T^l_n} - f^l_{md}(t^l_{n-1,n}+s^l_r)\frac{\partial t^l_{n-1,n}}{\partial T^l_n}\right) - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\partial t^l_{n,n+1}}{\partial T^l_n} - f^l_{md}(t^l_{n-1,n}+s^l_r)\frac{\partial t^l_{n-1,n}}{\partial T^l_n}\right) \quad (27)$$

$$\frac{\partial Ld^l_{r,n+1}(\boldsymbol{T}^l)}{\partial T^l_n} = -\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\partial t^l_{n,n+1}}{\partial T^l_n} + \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\partial t^l_{n,n+1}}{\partial T^l_n} \quad (28)$$

From Eqs. (26)–(28), it is also clear to find that the sum of these three is 0. It means that a shift of T^l_n will not change the total loading of all vehicles. Based on this, the partial derivative of $C^{l,r}_P$ for T^l_n can be expressed as:

$$\frac{\partial C^{l,r}_P(\boldsymbol{T}^l)}{\partial T^l_n} = \frac{\partial g\left(Ld^l_{r,n-1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n-1}(\boldsymbol{T}^l)}\cdot\frac{\partial Ld^l_{r,n-1}(\boldsymbol{T}^l)}{\partial T^l_n} + \frac{\partial g\left(Ld^l_{r,n}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n}(\boldsymbol{T}^l)}\cdot\frac{\partial Ld^l_{r,n}(\boldsymbol{T}^l)}{\partial T^l_n} + \frac{\partial g\left(Ld^l_{r,n+1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n+1}(\boldsymbol{T}^l)}\cdot\frac{\partial Ld^l_{r,n+1}(\boldsymbol{T}^l)}{\partial T^l_n} \quad (29)$$

Substituting Eqs. (26)–(28) into Eq. (29), the expression of $\partial C^{l,r}_P/\partial T^l_n$ for $n \in (1, N^l)$ is obtained. For $n = 1$ and N^l, the cases are similar. Thus the final expression of $\partial C^{l,r}_P/\partial T^l_n$ is formulated in formulation (30):

$$\frac{\partial C^{l,r}_P(\boldsymbol{T}^l)}{\partial T^l_n} = \begin{cases} \left(\frac{\partial g\left(Ld^l_{r,1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,1}(\boldsymbol{T}^l)} - \frac{\partial g\left(Ld^l_{r,2}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,2}(\boldsymbol{T}^l)}\right)\cdot\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}(t^l_{12}+s^l_r) - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}(t^l_{12}+s^l_r)\right)\cdot\frac{\beta}{\beta+\gamma} & \text{for } n=1 \\ \frac{\partial g\left(Ld^l_{r,n-1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n-1}(\boldsymbol{T}^l)}\cdot\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}(t^l_{n-1,n}+s^l_r) - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}(t^l_{n-1,n}+s^l_r)\right)\cdot\frac{\gamma}{\beta+\gamma} \\ \quad + \frac{\partial g\left(Ld^l_{r,n}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n}(\boldsymbol{T}^l)}\cdot\begin{pmatrix} \sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\beta}{\beta+\gamma} - f^l_{md}(t^l_{n-1,n}+s^l_r)\frac{\gamma}{\beta+\gamma}\right) \\ -\sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(f^l_{md}(t^l_{n,n+1}+s^l_r)\frac{\beta}{\beta+\gamma} - f^l_{md}(t^l_{n-1,n}+s^l_r)\frac{\gamma}{\beta+\gamma}\right)\end{pmatrix} & \text{for } 1<n<N^l \\ \quad + \frac{\partial g\left(Ld^l_{r,n+1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n+1}(\boldsymbol{T}^l)}\cdot\left(-\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}(t^l_{n,n+1}+s^l_r) + \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}(t^l_{n,n+1}+s^l_r)\right)\cdot\frac{\beta}{\beta+\gamma} \\ \left(\frac{\partial g\left(Ld^l_{r,N^l-1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,N^l-1}(\boldsymbol{T}^l)} - \frac{\partial g\left(Ld^l_{r,N^l}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,N^l}(\boldsymbol{T}^l)}\right)\cdot\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}(t^l_{N^l-1,N^l}+s^l_r) - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}(t^l_{N^l-1,N^l}+s^l_r)\right)\cdot\frac{\gamma}{\beta+\gamma} & \text{for } n=N^l \end{cases} \quad (30)$$

For any segment $r(1 \leqslant r \leqslant 2\times(M+D-1))$ on route $l, \partial C^{l,r}_P/\partial T^l_n$ is presented in a matrix: $\partial C^{l,R}_P/\partial T^l_n = \left(\partial C^{l,1}_P/\partial T^l_n, \ldots, \partial C^{l,r}_P/\partial T^l_n, \ldots, \partial C^{l,2\times(M+D-1)}_P/\partial T^l_n\right)$, and the final $\partial C^l_P/\partial T^l_n$ is:

$$\frac{\partial C^l_P(\boldsymbol{T}^l)}{\partial \boldsymbol{T}^l_n} = \frac{\partial \boldsymbol{C}^{l,R}_P(\boldsymbol{T}^l)}{\partial \boldsymbol{T}^l_n}\cdot\left(\boldsymbol{dis}^l\right)^T \quad (31)$$

Then we have: $\partial C^l_{DP}/\partial T^l_n = \partial C^l_D/\partial T^l_n + \partial C^l_P/\partial T^l_n$. The gradient of C^l_{DP}, ∇C^l_{DP}, can be formulated as: $\nabla C^l_{DP} = \left(\partial C^l_{DP}/\partial T^l_1, \ldots, \partial C^l_{DP}/\partial T^l_n, \ldots, \partial C^l_{DP}/\partial T^l_{N^l}\right)$. For all the routes, there is: $\nabla C_{DP} = \left(\nabla C^1_{DP}, \ldots, \nabla C^l_{DP}, \ldots, \nabla C^L_{DP}\right)$.

When the gradient is formulated, the Frank–Wolfe algorithm can be implemented. Its main steps are given as follows:

Step 1: Input the maximum iteration N_{cmax}; the fleet size for all the routes: $\boldsymbol{F} = (F^1, \ldots, F^l, \ldots, F^L)$; the number of vehicle trips for all the routes: $\boldsymbol{N} = (N^1, \ldots, N^l, \ldots, N^L)$; the round trip time: $\boldsymbol{rt} = (rt^1, \ldots, rt^l, \ldots, rt^L)$. Generate a timetable $\boldsymbol{T}_0$ for all routes randomly. Initialize the optimal solution $\boldsymbol{T}^* = \boldsymbol{T}_0, i = 1$. Go to step 2.
Step 2: If $i > N_{cmax}$, the algorithm ends, and output $\boldsymbol{T}^*$; otherwise, go to step 3.
Step 3: Calculate $\boldsymbol{C}_{DP}(\boldsymbol{T}^*)$ and $\nabla\boldsymbol{C}_{DP}(\boldsymbol{T}^*)$. We replace $\boldsymbol{C}_{DP}(\boldsymbol{T})$ with $\boldsymbol{C}'_{DP}(\boldsymbol{T})$, where $\boldsymbol{C}'_{DP}(\boldsymbol{T}) = \boldsymbol{C}_{DP}(\boldsymbol{T}^*) + \nabla\boldsymbol{C}_{DP}(\boldsymbol{T}^*)\cdot(\boldsymbol{T}-\boldsymbol{T}^*)^T$, for $\forall\boldsymbol{T}$. Then the original problem has been converted to a linear problem. Go to step 4.
Step 4: Solve the linear problem: $\min(\nabla C_{DP}(\boldsymbol{T}^*)\cdot\boldsymbol{T}^T)$. The optimal solution is denoted by $\boldsymbol{T}^*_1$.
Step 5: Set $\lambda = 1/(2\cdot i)$, and update $\boldsymbol{T}^* = \boldsymbol{T}^* + \lambda(\boldsymbol{T}^*_1 - \boldsymbol{T}^*), i = i+1$. Go to step 2.

For step 4, the optimal solution of the linear problem, $\boldsymbol{T}_1^*$, could be always obtained when it values a boundary solution in the solution space. This is an important proposition for the Frank–Wolfe algorithm and has been proofed in many researches, and we do not prove it here specially. The following work is to list all the boundary solutions to solve $\boldsymbol{T}_1^*$.

The boundary solutions in the solution space are actually the solutions that make some of the inequality signs in Eqs. (11)–(13) convert to the equality sign. Inspired by this idea, an algorithm based on the Breadth-first Search (BFS) is suggested to orderly obtain all the boundary solutions orderly. That means that for route l, it firstly searches all the boundary solutions for the first to the (F^l)th departure time, denoting by $TB^l_{1\to F^l}$. For each element in $TB^l_{1\to F^l}$, we calculate all the boundary solutions for the (F^l+1)th to the $(2F^l)$th departure time, denoting by $TB^l_{F^l\to 2F^l}$. By combining $TB^l_{1\to F^l}$ and $TB^l_{F^l\to 2F^l}$, we obtain $TB^l_{1\to 2F^l}$. The above process repeats until $TB^l_{1\to N^l}$ is obtained. The detailed steps are given as follows:

Step 1: Input the number of vehicle trips N^l, the fleet size F^l, the round trip time rt^l, and the studied time span LN. Set $R^l = \lceil N^l/F^l \rceil, N^l_e = N^l - (R^l - 1)\cdot F^l$. Initialize $i = 1$, and $\boldsymbol{TB}^l = \phi$. Go to step 2.

Step 2: If $i > 1$, go to step 6; otherwise, create a matrix, $\boldsymbol{dl}$, to present the lower bounds of departure times. Initialize $\boldsymbol{dl} = zeros(1, min(F^l, N^l))$, and go to step 3.

Step 3: Create a matrix, $\boldsymbol{ul}$, to present the upper bounds of departure times. If $R^l = 1$, set all the elements in $\boldsymbol{ul}$ to be $LN : \boldsymbol{ul} = LN(1, N^l)$, go to step 4; otherwise, set the first N^l_e elements in $\boldsymbol{ul}$ to be $(LN - rt^l\cdot R^l)$, while the others are set to be $\left(LN - rt^l\cdot(R^l-1)\right) : \boldsymbol{ul} = \left[(LN - rt^l\cdot R^l)\cdot ones(1, N^l_e), \left(LN - rt^l\cdot(R^l-1)\right)\cdot ones(1, F^l - N^l_e)\right]$, and go to step 4.

Step 4: Set $\boldsymbol{tb} = \boldsymbol{dl}$, and update $\boldsymbol{TB}^l = [\boldsymbol{TB}^l; \boldsymbol{tb}]$. For $j = length(ul) : (-1) : 1$, do $tb(j) = ul(j)$, and update $\boldsymbol{TB}^l = [\boldsymbol{TB}^l; \boldsymbol{tb}]$. Go to step 5.

Step 5: If $R^l = 1$, the algorithm ends, and output $\boldsymbol{TB}^l$; otherwise, go to step 6.

Step 6: If $i < R^l$, update $\boldsymbol{ul} = \boldsymbol{ul} + rt^l$, and set $i = i+1$, do step 7 to 11; otherwise if $i = R^l$, set $i = i+1$, and go to step 12; otherwise, the algorithm ends, and output $\boldsymbol{TB}$.

Step 7: Set $\boldsymbol{TB_1^l} = \boldsymbol{TB}^l$, $\boldsymbol{TB}_1^{l'} = \boldsymbol{TB_1^l}\left(:, \left[size(\boldsymbol{TB_1^l}, 2) - F^l + 1 : size(\boldsymbol{TB_1^l}, 2)\right]\right)$. This assignment means that taking the last F^l columns of $\boldsymbol{TB_1^l}$ as a new input. Then initialize k = 1, $\boldsymbol{TB}^l = \phi$. Input $\boldsymbol{ul}$ obtained in step 6 or step 12. Go to step 8.

Step 8: If $k > size(\boldsymbol{TB_1^l}, 1)$, return to step 6 with the output $\boldsymbol{TB}^l$ obtained in step 11; otherwise, set $\boldsymbol{tb_k} = \boldsymbol{TB}_1^{l'}(k,:), \boldsymbol{dl} = \left[\boldsymbol{tb_k} + rt^l\right]([1 : length(\boldsymbol{ul})])$, and go to step 9.

Step 9: Check for each element in $\boldsymbol{dl}$. For $\forall j (1 \leqslant j \leqslant length(\boldsymbol{dl}))$, if $\boldsymbol{dl}(j) < \boldsymbol{tb_k}(end)$, do the assignment: $\boldsymbol{dl}(j) = \boldsymbol{tb}_k(end)$, and go to step 10.

Step 10: Set $\boldsymbol{TB_2^l} = \boldsymbol{dl}, \boldsymbol{tb}_2 = \boldsymbol{dl}$. For $j = length(\boldsymbol{ul}) : (-1) : 1$, do the assignment: $\boldsymbol{tb}_2(j) = \boldsymbol{ul}(j)$. If $\boldsymbol{tb_2} \neq \boldsymbol{TB_2^l}(end,:)$, update $\boldsymbol{TB_2^l} = [\boldsymbol{TB_2^l}; \boldsymbol{tb}_2]$. Go to step 11.

Step 11: Set $\boldsymbol{tb} = \left[repmat\left(\boldsymbol{TB_1^l}(k,:), size(\boldsymbol{TB_2^l}, 1), 1\right), \boldsymbol{TB_2^l}\right]$. Update $\boldsymbol{TB}^l = [\boldsymbol{TB}^l; \boldsymbol{tb}]$, and set $k = k+1$. Go to step 8.

Step 12: If $N^l_e = 0$, the algorithm ends, and output $\boldsymbol{TB}^l$; otherwise, set $\boldsymbol{ul} = LN\cdot ones(1, N^l_e)$, and return to step 7–11.

In the above steps, step 7–11 is treated as an internal function, which aims to produce $\boldsymbol{TB}^l_{(i\cdot F^l+1)\to(i+1)\cdot F^l}$ based on $\boldsymbol{TB}^l_{1\to i\cdot F^l}$, and combine them together to formulate $\boldsymbol{TB}^l_{1\to(i+1)\cdot F^l}$. $\boldsymbol{dl}$ and $\boldsymbol{ul}$ denote the lower and the upper bounds of the $(i\cdot F^l+1)\to(i+1)\cdot F^l$ departure times, respectively.

Recording all the boundary solutions in the matrix $\boldsymbol{TB}^l$ with N^l columns and Nb^l rows, where Nb^l is the number of boundary solutions. The linear problem, i.e., min $\left(\nabla \boldsymbol{C}^l_{DP}(\boldsymbol{T}^{l*})\cdot(\boldsymbol{T}^l)^T\right)$ becomes a problem of seeking for a boundary solution $\boldsymbol{tb}^{l*}$ from all the elements in $\boldsymbol{TB}^l$ that minimizes $\left(\nabla \boldsymbol{C}^l_{DP}(\boldsymbol{T}^{l*})\cdot(\boldsymbol{tb}^{l*})^T\right)$. For all the routes, the optimal solution $\boldsymbol{T}_1^*$ is formulated as $\boldsymbol{T}_1^* = \left(\boldsymbol{tb}^{1*}, \ldots, \boldsymbol{tb}^{l*}, \ldots, \boldsymbol{tb}^{L*}\right)$.

5.2. A heuristic algorithm of shifting departure times

As the transfer cost has not been involved in the Frank–Wolfe algorithm, in this stage, a heuristic algorithm of shifting departure times (SDT) is proposed to reduce the total cost further by reducing the transfer cost. Before using SDT, the gradient of the transfer cost needs to be solved.

For route $l, \forall n \in (1, N^l)$, the partial derivative of C^l_{TR} for T^l_n can be formulated:

$$\frac{\partial C_{TR}^l\left(\boldsymbol{T}^l\right)}{\partial T_n^l}=\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n-1}^{d,l}\left(TR_k^d-T_{n-1}^l-sd_d^l\right)f_{md}^l\left(t_{n-1,n}^l+s_m^l\right)\frac{\partial t_{n-1,n}^l}{\partial T_n^l}-\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\int_{t_{n-1,n}^l+s_m^l}^{t_{n,n+1}^l+s_m^l}f_{md}^l(t)dt$$
$$+\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n}^{d,l}\left(TR_k^d-T_n^l-sd_d^l\right)\left(f_{md}^l\left(t_{n,n+1}^l+s_m^l\right)\frac{\partial t_{n,n+1}^l}{\partial T_n^l}-f_{md}^l\left(t_{n-1,n}^l+s_m^l\right)\frac{\partial t_{n-1,n}^l}{\partial T_n^l}\right)$$
$$-\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n+1}^{d,l}\left(TR_k^d-T_{n+1}^l-sd_d^l\right)f_{md}^l\left(t_{n,n+1}^l+s_m^l\right)\frac{\partial t_{n,n+1}^l}{\partial T_n^l} \tag{32}$$

In Eq. (32), the first term denotes the increase in the total transfer cost of the individuals on the $(n-1)$th vehicle (caused by the increase of its loading). Similarly, the second term denotes the decrease of the loading on the nth vehicle. The third term denotes the increase of the total transfer cost of the individuals on the nth vehicle. The fourth term denotes the decrease of the total transfer cost of the individuals on the $(n+1)$th vehicle. For $n = 1$ and N^l, the cases are similar. Thus the final expression of $\partial C_{TR}^l/\partial T_n^l$ is formulated as:

$$\frac{\partial C_{TR}^l\left(\boldsymbol{T}^l\right)}{\partial T_n^l}=\begin{cases}\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,1}^{d,l}\left(TR_k^d-T_1^l-sd_d^l\right)f_{md}^l(t_{12}^l+s_m^l)\frac{\beta}{\beta+\gamma}-\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\int_{s_m^l}^{t_{12}^l+s_m^l}f_{md}^l(t)dt & \\ -\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,2}^{d,l}\left(TR_k^d-T_2^l-sd_d^l\right)f_{md}^l(t_{12}^l+s_m^l)\frac{\beta}{\beta+\gamma} & \text{for } n=1\\ \sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n-1}^{d,l}\left(TR_k^d-T_{n-1}^l-sd_d^l\right)f_{md}^l\left(t_{n-1,n}^l+s_m^l\right)\frac{\gamma}{\beta+\gamma}-\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\int_{t_{n-1,n}^l+s_m^l}^{t_{n,n+1}^l+s_m^l}f_{md}^l(t)dt & \\ +\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n}^{d,l}\left(TR_k^d-T_n^l-sd_d^l\right)\left(f_{md}^l\left(t_{n,n+1}^l+s_m^l\right)\frac{\beta}{\beta+\gamma}-f_{md}^l\left(t_{n-1,n}^l+s_m^l\right)\frac{\gamma}{\beta+\gamma}\right) & \text{for } 1<n<N^l\\ -\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,n+1}^{d,l}\left(TR_k^d-T_{n+1}^l-sd_d^l\right)f_{md}^l\left(t_{n,n+1}^l+s_m^l\right)\frac{\beta}{\beta+\gamma} & \\ \sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,N^l-1}^{d,l}\left(TR_k^d-T_{N^l-1}^l-sd_d^l\right)f_{md}^l\left(t_{N^l-1,N^l}^l+s_m^l\right)\frac{\gamma}{\beta+\gamma}-\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\int_{t_{N^l-1,N^l}^l+s_m^l}^{LN+s_m^l}f_{md}^l(t)dt & \text{for } n=N^l\\ -\sum_{d=1}^{D_1}\sum_{m=1}^{Md}\sum_k\delta_{k,N^l}^{d,l}\left(TR_k^d-T_{N^l}^l-sd_d^l\right)f_{md}^l\left(t_{N^l-1,N^l}^l+s_m^l\right)\frac{\gamma}{\beta+\gamma} & \end{cases} \tag{33}$$

For all $T_n^l(1\leqslant n\leqslant N^l)$, $\partial C_{TR}^l/\partial T_n^l$ is presented in $\nabla\boldsymbol{C}_{TR}^l=\left(\partial C_{TR}^l/\partial T_1^l,\dots,\partial C_{TR}^l/\partial T_n^l,\dots,\partial C_{TR}^l/\partial T_{N^l}^l\right)$. For route l, the gradient of the total cost, ∇C_T^l, is formulated as: $\nabla\boldsymbol{C}_T^l=\nabla\boldsymbol{C}_{DP}^l+\nabla\boldsymbol{C}_{TR}^l$. For all routes, there is $\nabla\boldsymbol{C}_T=\left(\nabla\boldsymbol{C}_T^1,\dots,\nabla\boldsymbol{C}_T^l,\dots,\nabla\boldsymbol{C}_T^L\right)$.

When $\nabla\boldsymbol{C}_T$ is obtained, SDT can be implemented. In each iteration of SDT, one departure time is shifted. For $\forall T_n^l\in\boldsymbol{T}$, $TR_n^{l,d}$ is the arrival time of the train joint at station d. $\boldsymbol{T\_Trans}_n^l=\left(TR_n^{l,1},\dots,TR_n^{l,d},\dots,TR_n^{l,D}\right)$ represents the arrivals times of the trains joint for it at all stations. We denote by $\boldsymbol{Cd1\_T_n^l}=\boldsymbol{T\_Trans_n^l}-\boldsymbol{sd}^l$ the candidate departure times that T_n^l an be shifted afterwards, by $\boldsymbol{Cd2\_T_n^l}=\boldsymbol{Cd1\_T_n^l}-\boldsymbol{H_R}$ the candidate departures times that T_n^l can be shifted forwards, where $\boldsymbol{H_R}=\left(H_R^1,\dots,H_R^d,\dots,H_R^D\right)$ denotes the train headway at each station. Considering Eqs. (11)–(13), the lower and the upper bounds of T_n^l are denoted by dT_n^l and uT_n^l, respectively. The main steps of SDT are given as follows:

Step 1: Input $\boldsymbol{T}^*$ and N_{smax}. Those are the optimal timetable obtained in the first stage, and the maximum iteration, respectively. Calculate $\nabla\boldsymbol{C_T}(\boldsymbol{T}^*)$ based on Eqs. (25), (30), and (33), and set $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)=\nabla\boldsymbol{C_T}(\boldsymbol{T}^*)$. Initialize $i = 1$. Go to
step 2.

Step 2: If $i>N_{smax}$, the algorithm ends, and output $\boldsymbol{T}^*$; otherwise, set $\boldsymbol{temp\_\nabla C_T}(\boldsymbol{T}^*)=\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end,:),\ i=i+1$, and go to step 3.

Step 3: If $sum(|\boldsymbol{temp\_\nabla C_T}|)=0$, the algorithm ends, and output $\boldsymbol{T}^*$; otherwise, do step 4–5.

Step 4: Find the maximum element in $|\boldsymbol{temp\_\nabla C_T}|$, assuming that it is the jth element in $|\boldsymbol{temp\_\nabla C_T}|$ and corresponds to the departure time T_n^l. Calculate $\boldsymbol{Cd1\_T_n^l}$, $\boldsymbol{Cd2\_T_n^l}$, dT_n^l, uT_n^l.

Step 5: If any of the following three conditions is satisfied, set $\boldsymbol{temp\_\nabla C_T}(\boldsymbol{T}^*)(j)=0$, and go to step 3; otherwise, break from this circulation, and go to step 6.

(1) $\boldsymbol{temp\_\nabla C_T}(j)<0$, and there exists at least one k that makes $\boldsymbol{Cd1\_T_n^l}(k)=\boldsymbol{T\_Trans_n^l}(k)$;

(2) $\boldsymbol{temp\_\nabla C_T}(j)<0$ and $\boldsymbol{Cd1\_T_n^l}(k)\notin[dT_n^l,uT_n^l]$, for $\forall k$;

(3) $\boldsymbol{temp\_\nabla C_T}(j)>0$ and $\boldsymbol{Cd2\_T_n^l}(k)\notin[dT_n^l,uT_n^l]$, for $\forall k$.

Step 6: Update $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*) = [\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*); \boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end,:)]$. If $\boldsymbol{temp\_\nabla C_T}(j) < 0$, set $\boldsymbol{Cd\_T_n^l} = \left\{ \boldsymbol{Cd1\_T_n^l}(k) \middle| Cd1\_T_n^l(k) \in [dT_n^l, uT_n^l].\right\}$, and sort $\boldsymbol{Cd\_T_n^l}$ in ascending order, then go to step 7; otherwise, set $\boldsymbol{Cd\_T_n^l} = \left\{ \boldsymbol{Cd2\_T_n^l}(k) \middle| \boldsymbol{Cd2\_T_n^l}(k) \in [dT_n^l, uT_n^l] \right\}$, and $\boldsymbol{Cd'\_T_n^l} = \boldsymbol{Cd\_T_n^l} + eps$ (eps is a very small positive number), sort $\boldsymbol{Cd\_T_n^l}$, $\boldsymbol{Cd'\_T_n^l}$ in descending order, and go to step 9.

Step 7: Set $\boldsymbol{CN} = \phi$, $\boldsymbol{DN} = \phi$, $t1 = \boldsymbol{T}^*(j)$. For $k = 1 : length(\boldsymbol{Cd\_T_n^l})$, do the assignment: $\boldsymbol{T}^*(j) = \boldsymbol{Cd\_T_n^l}(k)$, and update $\boldsymbol{CN} = [\boldsymbol{CN}; \boldsymbol{C_T}(\boldsymbol{T}^*)]$, $\boldsymbol{DN} = [\boldsymbol{DN}; \partial C_T(\boldsymbol{T}^*)/\partial T_n^l]$. If $\boldsymbol{DN}(end) > 0$, break from the circulation, and go to step 8.

Step 8: If $k = 1$ and $\boldsymbol{DN}(end) > 0$, set $t2 = \boldsymbol{Cd\_T_n^l}(1)$, and go to step 12; otherwise, find the minimum element in $\boldsymbol{CN}$, assuming its index is k^*, do the assignment: $\boldsymbol{T}^*(j) = \boldsymbol{Cd\_T_n^l}(k^*)$, and go to step 13.

Step 9: Set $\boldsymbol{CN} = \phi$, $\boldsymbol{DN} = \phi$, $t1 = \boldsymbol{T}^*(j)$. For $k = 1 : length(\boldsymbol{Cd\_T_n^l})$, do the assignment: $\boldsymbol{T}^*(j) = \boldsymbol{Cd'\_T_n^l}(k)$, and update $\boldsymbol{CN} = [\boldsymbol{CN}; \boldsymbol{C}_T(\boldsymbol{T}^*)]$, $\boldsymbol{DN} = [\boldsymbol{DN}; \partial C_T(\boldsymbol{T}^*)/\partial T_n^l]$. If $\boldsymbol{DN}(end) < 0$, break from the circulation, and go to step 10.

Step 10: If $\boldsymbol{DN}(end) > 0$, set $\boldsymbol{T}^*(j) = \boldsymbol{Cd\_T_n^l}(k)$, and go to step 13; otherwise, go to step 11.

Step 11: If $k = 1$, set $t2 = \boldsymbol{Cd'\_Tn^l}(1)$, and go to step 12; otherwise, set $t1 = \boldsymbol{Cd\_T_n^l}(k-1)$, $t2 = \boldsymbol{Cd'\_T_n^l}(k)$, and go to step 12.

Step 12: Find a $t \in (\min(t1, t2), \max(t1, t2))$ that minimizes $\boldsymbol{C_T}(\boldsymbol{T}^*)$, and go to step 13.

Step 13: Update $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)$. Do $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j) = \partial C_T(\boldsymbol{T}^*)/\partial T_n^l$. If $n > 2$, do $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j-2) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n-2}^l$, $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j-1) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n-1}^l$; if $n = 2$, do $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j-1) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n-1}^l$. Go to step 14.

Step 14: Continue updating $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)$. If $n < N^l - 1$, do $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j+1) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n+1}^l$, $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j+2) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n+2}^l$; if $n = N^l - 1$, do $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)(end, j+1) = \partial C_T(\boldsymbol{T}^*)/\partial T_{n+1}^l$. Return to step 2.

For step 5, the explanations of the three conditions are given below. Condition (1): $\boldsymbol{temp\_\nabla C_T}(j) < 0$ means that C_T is decreasing at the current departure time, and it should be shifted to afterwards (let the trip depart later). While the latter part means that the current departure time just ensuring the transfer delay at a certain station equals to 0. A slight move to afterwards will cause a sudden increase of C_T. For this situation, we treat the current departure time cannot be shifted. Conditions (2) and (3) both mean that there is no candidate departure time.

For step 12, the partial derivatives of C_T for $t1$ and $t2$ always have opposite signs. Therefore, there exists a $t \in (\min(t1, t2), \max(t1, t2))$ that makes $C_T(\boldsymbol{T}^*)$ minimized.

For steps 13 and 14, when one trip departure time is shifted, we only need to update at most five elements in the last row of $\boldsymbol{S\_\nabla C_T}(\boldsymbol{T}^*)$. Because through Eqs. (25), (30) and (33), we know that if T_n^l (for $2 < j < N^l - 1$) is shifted, $\partial C_D^l/\partial T_{n-1}^l$, $\partial C_D^l/\partial T_n^l$, $\partial C_D^l/\partial T_{n+1}^l$ will be affected for C_D^l; $\partial C_P^l/\partial T_{n-2}^l$, $\partial C_P^l/\partial T_{n-1}^l$, $\partial C_P^l/\partial T_n^l$, $\partial C_P^l/\partial T_{n+1}^l$, $\partial C_P^l/\partial T_{n+2}^l$ will be affected for C_P^l; $\partial C_{TR}^l/\partial T_{n-1}^l$, $\partial C_{TR}^l/\partial T_n^l$, $\partial C_{TR}^l/\partial T_{n+1}^l$ will be affected for C_{TR}^l. Therefore, for C_T, at most five partial derivatives will be affected, and we only need to update them instead of updating the whole set of partial derivatives. That will save some computing time.

Based on the above steps, the flow chart of FW-SDT is given in Fig. 2.

From Fig. 2, it is clear to see that two conditions will terminate the algorithm:

(1) $i > N_{smax}$;
(2) $sum(|\boldsymbol{temp\_\nabla C_T}|) = 0$, which means all the elements in $\boldsymbol{temp\_\nabla C_T}$ are 0. The element in $\boldsymbol{temp\_\nabla C_T}$ is 0 if it satisfies one of the three conditions in step 5. So in this situation, there is no departure time can be shifted, and the algorithm ends.

6. Examples and numerical analysis

6.1. A simulated network example

The network in Fig. 1 is taken as the simulated network. In Fig. 1, there are 5 shuttle routes, 22 bus stops (Nos. 1–22) and 3 metro stations (Nos. 23–25). For each OD pair, there exists a certain distribution function, which is set to be a Gaussian function. Using Eqs. (1)–(3) to assign the demand of each OD pair, $f_{md}^l(t)$ for $\forall m, d, l$ is obtained. For stop m of route l, we can formulate the cumulative distribution of boarding demand at the stop, $f_m^l(t)$, by accumulating $f_{md}^l(t)$ for each d. Fig. 3 shows the cumulative distribution of boarding demand at each stop/station on Route 1 for both directions. Note that Fig. 3 only shows the distribution curve at the stops/stations where the shuttle needs to stop during its round trip.

In this example, $LN = 120$ (min), $s_m^l \leqslant 30$ for $\forall l, m$. The time span of all the stops is $120 + 30 = 150$ (min). $g(x) = (1.12^x - 1)/100$, fleet size matrix $\boldsymbol{F} = (4, 4, 5, 3, 5)$, the number of vehicle trips $\boldsymbol{N} = (7, 11, 11, 9, 14)$, the dwell time at each stop $t_s = 0.5$ (min), the vehicle speed $v_s = 20$ (km/h) and is set to be constant $\mathrm{dt} = 0.5$ (min). The link travel time is set to be the link length divided by the vehicle speed. The travel cost (ticket price) of each route $c_{ij} = 1$ yuan (the official currency of China) for $\forall i, j$. The other parameters are given as follows: $\beta = 0.4$, $\gamma = 0.6$, $a_1 = 0.2$, $a_2 = 2$, $a_3 = 1.8$, $\mu_1 = 1.5$ (yuan/min), $\mu_2 = 1$ (yuan/min), $\mu_3 = 0.6$. The parameters of GA are: $NIND = 40$, $MAXGEN = 30$, $GGAP = 0.9$, $P_c = 0.8$, $P_m = 0.15$. The maximum iteration of FW, $N_{cmax} = 60$; The maximum iteration of SDT, $N_{smax} = 60$.

We use three algorithms to solve the problem, respectively:

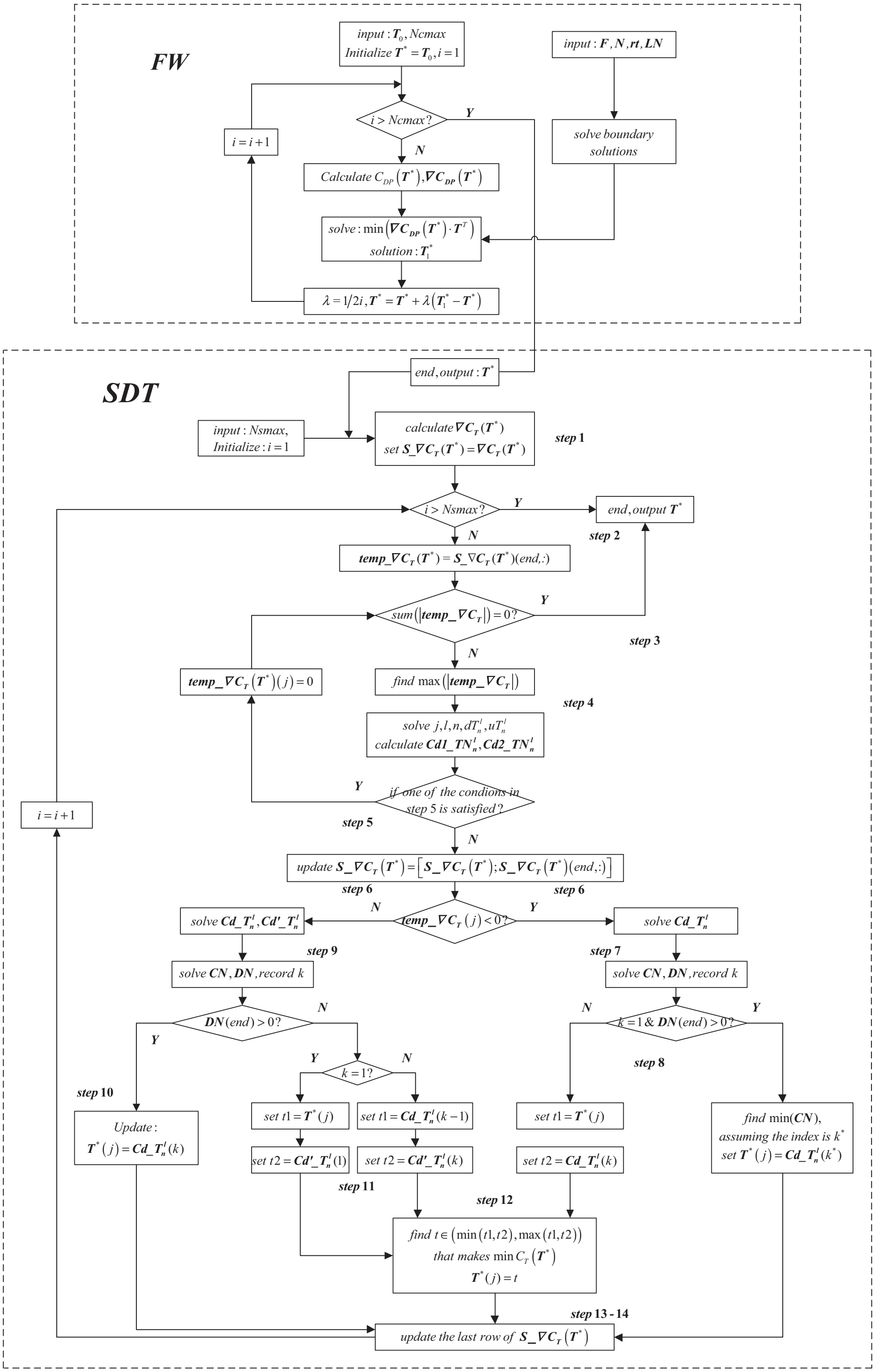

Fig. 2. Flow chart of FW-SDT.

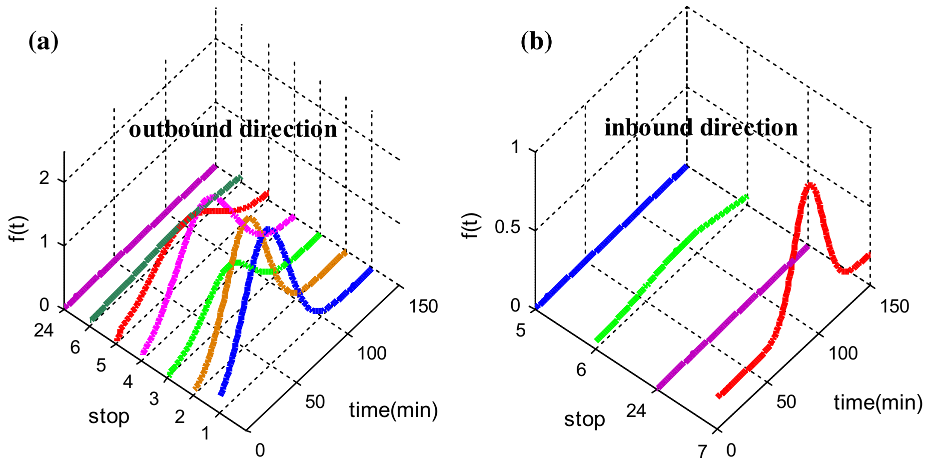

Fig. 3. Cumulative distribution of boarding demand at stops/stations on Route 1: (a) outbound direction and (b) inbound direction.

(1) GA: the evolutions of each cost and the total cost of each route are shown in Fig. 4.
(2) FW-SDT: the optimizing process of FW and SDT are shown in Figs. 5 and 6, respectively.
(3) GA-SDT: using GA instead of FW to optimize C_{DP} in the first stage. The evolutions are shown in Fig. 7.

In Fig. 4(a) and (b), we can see significant decrease in the first ten generations, and all curves become stable gradually after that.

In Fig. 5(a), the initial C_{DP} is quite large caused by a large overloading penalty, but it decreases sharply within the first ten circulations. In Fig. 5(b), all the costs basically remain stable after the 40th iteration. For the SDT's optimizing process in Fig. 6, the transfer cost decreases significantly while the overloading penalty nearly remains stable and the schedule delay cost increases slightly. Similarly, the total cost shares the same decreasing trend with the transfer cost.

Comparing Fig. 6 with Fig. 7(b), SDT in Fig. 7(b) does not perform as well as it performs in Fig. 6 though the total cost keeps decreasing. This is mainly due to the different input timetables obtained by FW and GA. Comparing Fig. 5(b) with Fig. 7(a), it's easy to see that the former result is more optimized, and it leads to a smaller transfer cost at the beginning of SDT stage. Experimental results show that although the output of FW influences the SDT's result to some extent, SDT still has a relatively stable performance.

The detailed objectives obtained by the above three algorithms are shown in Table 1.

From Table 1, FW-SDT performs best in terms of C_D, C_P, C_{TR}, C_T and its CPU time. The CPU time of GA-SDT is a little less than that of GA, while GA performs a little better than GA-SDT in the optimized results. FW-SDT uses 50.6% of the CPU time of GA and obtains a smaller total cost (72.8% of that obtained by GA). To illustrate the efficiency of FW-SDT, the runs for calculating C_D, C_P, C_{TR}, $\partial C_D/\partial T$, $\partial C_P/\partial T$, $\partial C_{TR}/\partial T$ of each route are provided in Table 2. Note that the number of runs here means the number of vehicle trips multiplied by the involved stops/segments. Therefore, the runs of different routes are different because of the different number of vehicle trips and stops/segments during the round trips. Then the runs of calculations within one generation of GA, FW and SDT are provided in Table 3.

In Table 3, it is clear that GA needs much more runs of calculations than FW-SDT in one generation because there are a number of individuals in the population within each generation of GA, while FW-SDT calculates each item based on the current one solution within each circulation. As a result, GA may take more than twice of the time spent by FW-SDT to obtain an optimized solution as well as that obtained by FW-SDT.

To test the effectiveness of GA and FW-SDT, sensitivity analyses of related parameters are made. Varying the population size (*NIND*) and the maximum generation (*MAXGEN*) of GA, the minimized C_T and the CPU time of each case are presented in Table 4. The results in each case of Table 4 are obtained by calculating the average of three times using GA. The units of the CPU times here and in the following tables are all seconds.

In Table 4, when *NIND* changes from 30 to 50, C_T keeps decreasing for all the cases. When the values of *NIND* are near 30, a small increase of *NIND* will lead to a relatively large decrease of C_T. This trend becomes inconspicuous as *NIND* increasing. Similar results can be also found between C_T and *MAXGEN*. Generally speaking, the change of *MAXGEN* cannot influence the final result as much as that of *NIND*. On the other hand, the CPU time almost grows linearly as the overall circulation number ($NIND \times MAXGEN$). Specially, there are two cases ($NIND = 50$, $MAXGEN = 40$ and $NIND = 50$, $MAXGEN = 50$) in Table 4 that obtain a little smaller C_T than the one obtained by FW-SDT previously. However, the CPU time of these cases are more than three times of that of FW-SDT.

Then we use FW-SDT to solve the problem repeatedly by varying the maximum iteration of FW (N_{cmax}) and the maximum iteration of SDT (N_{smax}). Table 5 presents the minimized C_T and the CPU time (s) of each case, and the results are also obtained by calculating the average of three times using FW-SDT.

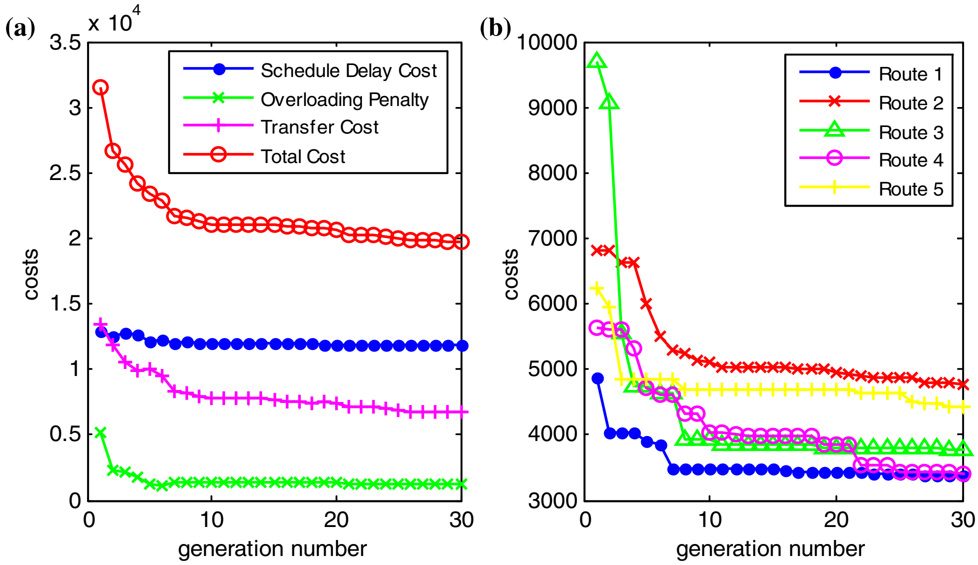

Fig. 4. Evolution process of GA: (a) the evolution of each cost and (b) the evolution of each route.

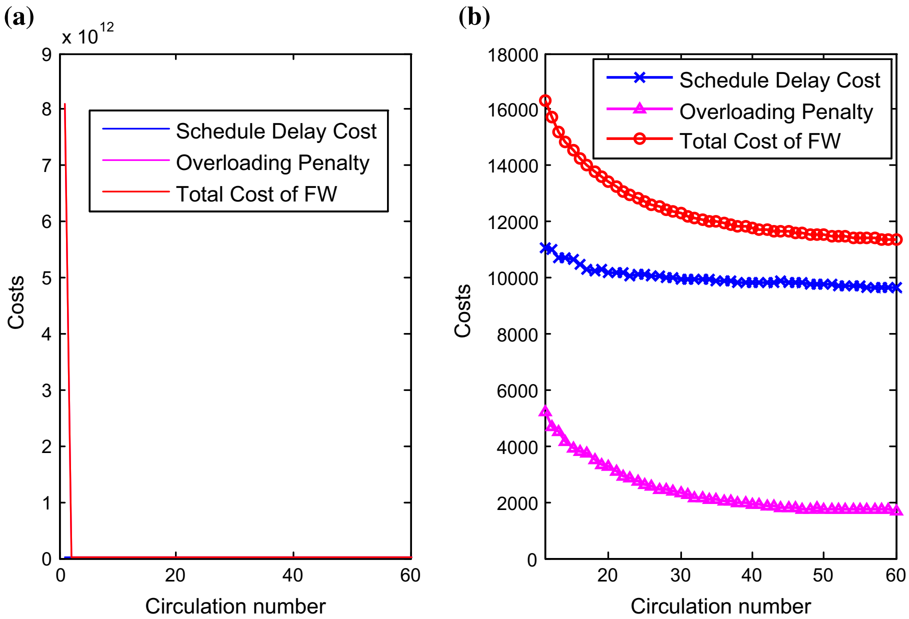

Fig. 5. Optimizing process of FW: (a) iteration 1–60 and (b) iteration 11–60.

In Table 5, when N_{cmax} is given, C_T decreases with N_{smax} increasing in general, while it does not always keep decreasing with N_{cmax} increasing for a given N_{smax}. This case suggests that when $N_{cmax} > 40$, increasing N_{cmax} cannot always ensure to obtain a more optimized final solution because of the exclusion of the transfer cost in FW. The CPU time also keeps a linear growth generally as the increase of overall circulation number ($N_{cmax} + N_{smax}$). Moreover, the same increase of N_{smax} will lead to a more increase of CPU time than that of N_{cmax} because SDT needs more runs of calculations in one circulation than FW (as shown in Table 3). Generally speaking, the results in Table 5 vary in a relatively small range and are smaller than GA's results in both C_T and CPU time. Therefore, we conclude that FW-SDT performs relatively stable and is better than GA in both efficiency and effectiveness.

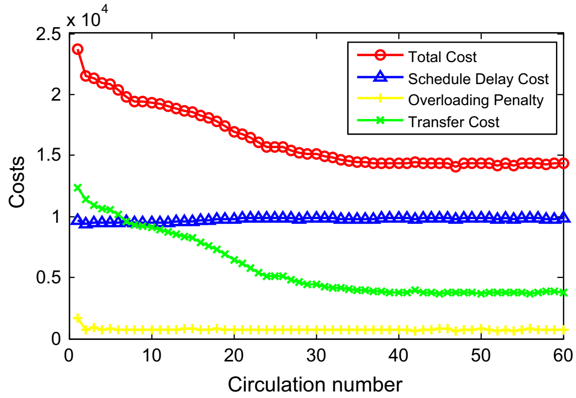

Fig. 6. Optimizing process of SDT.

6.2. Numerical analysis

For route l, when the fleet size is given and the vehicle trips are with an even headway, the number of vehicle trips is maximized. Thus the upper bound of the number of vehicle trips, ul_N^l, is formulated as:

$$ul\_N^l = \left\lfloor F^l \cdot LN/rt^l \right\rfloor \tag{34}$$

In the example, we set $\boldsymbol{F} = (4,4,5,3,5)$. The upper bound of the number of vehicle trips for each route, $\boldsymbol{ul\_N}$, is obtained: $\boldsymbol{ul\_N} = (15,15,24,14,19)$. When $\boldsymbol{F}$ remains unchanged, different costs are obtained by changing $\boldsymbol{N}$. The detailed results are shown in Table 6. The results in Table 6 are obtained by calculating the minimum of five times using FW. Notice that the transfer time is only related to the train's headway, while not related to the frequency of the shuttle, so C_{TR} is not considered in Table 6. Moreover, a_1 is set to value 0 here, which means that the OD assignment is unrelated to the bus frequency. The C_{DP} of each route for all the cases are presented in Fig. 8.

In Fig. 8, C_{DP} decreases significantly at the beginning for all the routes. When $\boldsymbol{N}$ approaches $\boldsymbol{ul\_N}$, the decreasing trend is not obvious and increases even occur (e.g. case 5–6 for the five routes). This is because when the fleet size is given, the increase of $\boldsymbol{N}$ has two effects: (1) Decreasing the average schedule delay cost and the overloading penalty. (2) Narrowing the range of feasible solutions. It means as $\boldsymbol{N}$ increasing, the intensity of fleet size constraint becomes stronger, and the headways of every two adjacent vehicles tend to be even. When $\boldsymbol{N} = \boldsymbol{ul\_N}$, the timetable is basically an even-headway timetable. Based on this, the relationship between $\boldsymbol{N}$ and C_{DP} could be very complex, and it is highly correlated with the passenger

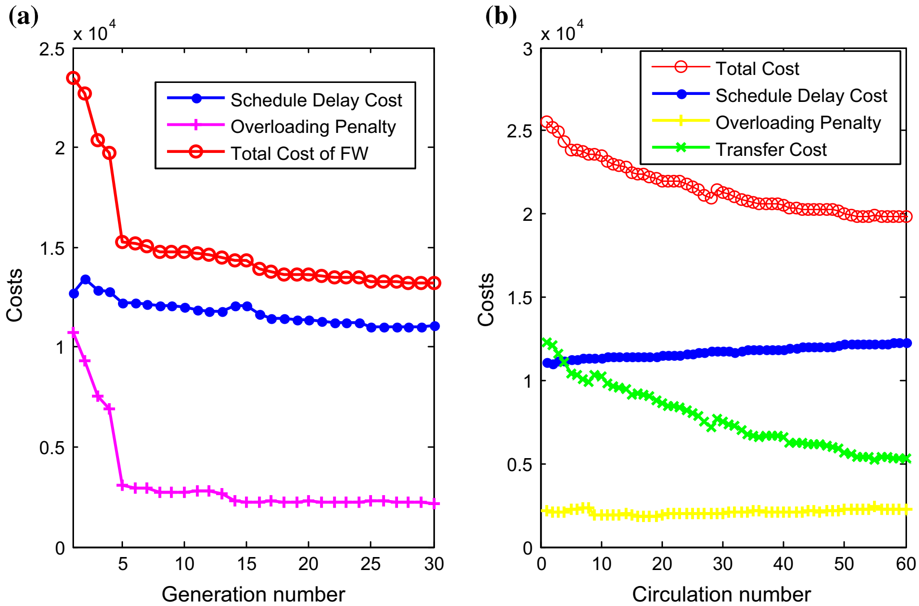

Fig. 7. Evolution process of GA-SDT: (a) GA's evolution process without transfer cost and (b) SDT's optimizing process based on GA's result.

Table 1
A comparison of costs obtained by GA, FW-SDT and GA-SDT.

	GA				FW-SDT				GA-SDT			
	C_D	C_P	C_{TR}	C_T	C_D	C_P	C_{TR}	C_T	C_D	C_P	C_{TR}	C_T
Route 1	2393.3	272.4	723.9	3388.7	1964.3	61.9	155.7	2181.9	2657.9	826.5	799.3	4283.7
Route 2	2563.4	176.9	2016.7	4758.3	2513.8	414.9	1556.20	4484.9	2891.5	346.7	1044.5	4282.6
Route 3	2437.3	302.3	1038.3	3778.2	1693.4	17.2	789.8	2500.3	2229.4	182.4	678.3	3090.1
Route 4	1777.4	243.4	1386.2	3408.4	1689.2	111.9	1128.0	2929.0	2084.1	739.1	1871.8	4695.0
Route 5	2685.2	203.3	1531.4	4419.9	1981.3	135.7	158.7	2275.6	2398.0	173.5	880.1	3451.7
Network	11856.6	1198.3	6696.5	19753.5	9842.0	741.5	3788.4	14371.9	12260.9	2268.2	5274.0	19803.0
CPU time (s)	471.1				238.5				448.1			

Table 2
Runs of related calculations of each route.

	C_D	C_P	C_{TR}	$\partial C_D/\partial T$	$\partial C_P/\partial T$	$\partial C_{TR}/\partial T$
Route 1	56	77	49	56	231	196
Route 2	99	121	66	99	363	264
Route 3	88	121	66	88	363	264
Route 4	63	81	63	63	243	252
Route 5	140	196	98	140	588	392
Network	446	596	342	446	1788	1368

Table 3
Runs of calculations within one generation of GA (*NIND* = 40), FW and SDT.

	C_D	C_P	C_{TR}	$\partial C_D/\partial T$	$\partial C_P/\partial T$	$\partial C_{TR}/\partial T$
GA	16,056	21,456	12,312	0	0	0
FW	446	596	0	446	1788	0
SDT	1449.5[a]	1937[a]	1111.5[a]	38.3[a]	153.5[a]	116.3[a]

[a] The runs of SDT are actually the expected runs of calculations because of the uncertain detailed process and the uncertain number of deviations need updating within each circulation. (See step 12–14 of SDT in Section 5.2.)

Table 4
Total cost and CPU time of each case by varying population size and the maximum generation of GA.

	NIND = 30		*NIND* = 35		*NIND* = 40		*NIND* = 45		*NIND* = 50	
	C_T	CPU time	C_T	CPU time	C_T	CPU time	C_T	CPU time	C_T	CPU time
MAXGEN = 20	20620.1	240.2	19392.7	277.6	18021.5	318.2	17526.1	359.8	17257.3	395.6
MAXGEN = 30	18907.2	354.5	18735.1	402.0	18092.2	459.9	16750.9	511.0	15289.7	565.5
MAXGEN = 40	19036.8	447.8	17922.0	493.3	16468.5	571.3	15394.5	649.1	14103.7	730.5
MAXGEN = 50	17120.4	551.0	16513.4	630.5	15392.1	728.2	14581.0	811.4	14219.9	903.5

Table 5
Total cost and CPU time of each case by varying N_{cmax} and N_{smax} of FW-SDT.

	N_{smax}=40		N_{smax}=50		N_{smax}=60		N_{smax}=70	
	C_T	CPU time	C_T	CPU time	C_T	CPU time	C_T	CPU time
$N_{cmax} = 40$	15525.3	156.4	15280.2	183.1	15352.5	210.9	14780.9	240.5
$N_{cmax} = 50$	16101.7	166.6	14893.9	187.5	14178.6	226.7	14145.6	253.7
$N_{cmax} = 60$	14883.1	175.6	14617.9	210.4	14351.2	239.6	14348.9	275.6
$N_{cmax} = 70$	15103.2	183.2	14829.3	216.0	14501.3	251.3	13965.1	288.9

Table 6
Detailed costs of all cases by changing $\boldsymbol{N}$.

Case no. N		1 (5,5,9,4,9)	2 (7,7,12,6,11)	3 (9,9,15,8,13)	4 (11,11,18,10,15)	5 (13,13,21,12,17)	6 (15,15,24,14,19)
Route 1	C_D	2566.9	1671.4	1471.4	1187.6	1039.6	1508.3
	C_P	199.7	127.9	57.4	39.0	24.5	69.8
	C_{DP}	2766.6	1799.3	1528.8	1226.6	1064.1	1578.1
Route 2	C_D	4187.6	3575.1	2609.5	2293.2	1859.1	2469.8
	C_P	1147.9	535.5	383.0	284.5	131.0	152.4
	C_{DP}	5335.6	4110.6	2992.6	2577.7	1990.1	2622.2
Route 3	C_D	2281.9	1395.9	1141.7	1094.3	896.7	1161.6
	C_P	70.7	11.7	4.2	3.6	2.2	15.9
	C_{DP}	2352.6	1407.6	1145.9	1097.8	898.9	1177.5
Route 4	C_D	3458.3	2736.1	1834.3	1575.0	1168.5	1417.3
	C_P	517.8	411.3	379.3	81.3	47.0	125.8
	C_{DP}	3976.1	3147.4	2213.6	1656.2	1215.5	1543.2
Route 5	C_D	3001.7	1901.5	1737.8	1487.6	1495.6	1885.2
	C_P	996.5	273.3	115.2	71.8	10.6	116.7
	C_{DP}	3998.3	2174.8	1852.9	1559.4	1506.2	2001.9

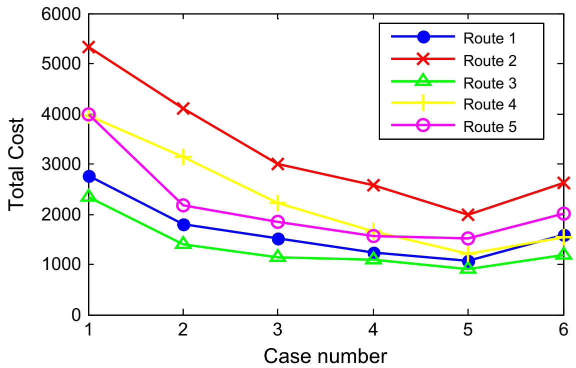

Fig. 8. C_{DP} of each route for all cases.

arrival distribution. Basically, when $\boldsymbol{N}$ is small or the passenger arrival distribution tends to be an uniform distribution, the effect (1) plays a dominant role. When $\boldsymbol{N}$ closes to $\boldsymbol{ul\_N}$ or the passenger arrival distribution has dramatic fluctuations, C_{DP} may be stable or even increase with $\boldsymbol{N}$ increasing. So when the fleet size is given, increasing the number of vehicle trips is not always a good idea. It will probably increase the passenger's travel cost and will definitely increase the operating cost.

Similarly, for route l, when the number of vehicle trips is given, the lower bound of the fleet size is:

$$dl\_F^l = \left\lceil rt^l \cdot N^l / LN \right\rceil \tag{35}$$

Here we set $\boldsymbol{N}$ = (7,11,11,9,14), and the lower bound of the fleet size for each route, $\boldsymbol{dl\_F}$, is solved: $\boldsymbol{dl\_F} = (2,3,3,2,4)$. By changing $\boldsymbol{F}$, different costs are obtained, which are shown in Table 7. The results in each case of Table 7 are also obtained by calculating the minimum of five times using FW-SDT. The total costs of each route for all the cases are presented in Fig. 9.

In Fig. 9, the total cost decreases as $\boldsymbol{F}$ increasing in general and the curves tend to be stable finally. This is because for a given $\boldsymbol{N}$, with $\boldsymbol{F}$ increasing, the intensity of the fleet size constraint becomes weaker, and then more and more solutions become feasible. When $\boldsymbol{F}$ increases to a certain value, the fleet size constraint does not take effect. In such a case, the curve of the total cost gets stable because of the same solution obtained theoretically.

6.3. A real life example

The real life study area is the *Huilongguan* Community, which is large in size and located in suburb of Beijing, China. In Fig. 10, four shuttle routes serve the community during peak hours, involving 40 bus stops (Nos. 1–40) and 3 metro stations (Nos. 41–43). The terminals of the routes are stop 1, 12, 23, 25, respectively. The passenger arrival distribution is obtained by checking the number of arrivals within every 15 min at each stop. It is treated to be uniformly distributed within every

Table 7
Detailed costs of all cases by changing ***F***.

Case no. F		1 (2,3,3,2,4)	2 (3,4,4,3,5)	3 (4,5,5,4,6)	4 (5,6,6,5,7)	5 (6,7,7,6,8)	6 (7,8,8,7,9)
Route 1	C_D	2294.7	1920.6	1964.3	1824.8	1882.2	1824.8
	C_P	270.6	58.9	61.9	89.4	44.0	49.4
	C_{TR}	780.2	423.1	155.7	163.7	116.7	155.7
	C_T	3345.6	2402.5	2181.9	2077.9	2042.9	2029.9
Route 2	C_D	2509.4	2321.3	2059.0	2295.4	1958.0	1902.4
	C_P	345.7	122.1	46.3	39.3	47.2	52.2
	C_{TR}	2127.3	1860.1	1494.5	961.2	1048.9	1022.5
	C_T	4982.4	4303.5	3599.8	3295.9	3054.1	2977.0
Route 3	C_D	1830.0	1799.4	1631.6	1582.3	1416.2	1451.3
	C_P	92.5	35.7	14.4	18.9	15.4	14.3
	C_{TR}	1120.5	1008.9	537.9	488.8	503.2	518.0
	C_T	3043.0	2844.0	2183.8	2090.1	1934.9	1983.5
Route 4	C_D	1585.0	1320.0	1381.9	1588.0	1556.1	1455.4
	C_P	237.6	160.0	96.7	30.6	42.1	42.0
	C_{TR}	1380.4	1328.2	1320.7	1082.4	1020.0	1075.7
	C_T	3202.9	2808.1	2799.3	2701.0	2618.2	2573.1
Route 5	C_D	2073.7	1675.1	1486.5	1321.4	1342.3	1295.7
	C_P	187.2	168.0	52.3	43.9	35.6	22.2
	C_{TR}	148.4	338.1	381.0	486.3	455.1	500.7
	C_T	2409.3	2181.2	1919.8	1851.7	1833.0	1818.6

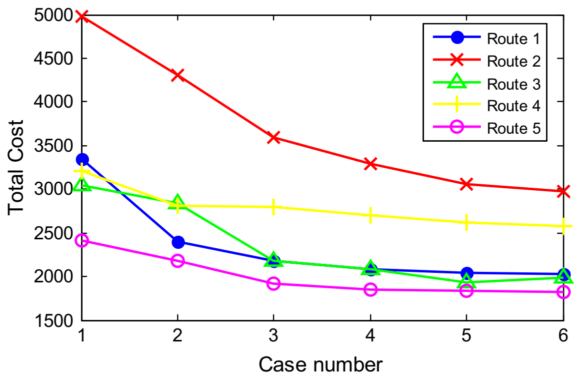

Fig. 9. C_T of each route for all cases.

15 min. The distribution of each OD demand is simply set to be the arrival distribution at the bus stop multiplied by a percentage. The percentage is determined by questionnaire surveys at the bus stops. Our objective is to optimize the timetables at the terminals during the morning peak hours (6:30–9:00 am). The current fleet size and the number of vehicle trips remain unchanged. We use GA and FW-SDT to obtain the optimized timetables, respectively. The parameters of GA, FW and SDT are: $NIND = 40$, $MAXGEN = 30$, $N_{cmax} = 60$, $N_{smax} = 60$. Related results are shown in Table 8 and the runs of calculations of each algorithm are shown in Table 9. Table 10 provides the current timetable and the optimized one obtained by FW-SDT.

From Table 8, the solutions solved by GA and FW-SDT are both better than the current one. GA reduces the total cost by 35.7%, while FW-SDT reduces it by 44.4%. In practice, we can reduce the passenger's travel cost by nearly a half just by adjusting departure times, without any more investment of vehicles or operating costs. That is a simple and feasible way in the transit planning stage. From Table 9, it is clear that FW-SDT takes less than a half of GA's CPU time to obtain a more optimized solution due to less runs of calculations compared to GA. By comparing the CPU times in Tables 9 and 1, the CPU times of both GA and FW-SDT increase when solving the real life example. However, the increase of GA's CPU time is much larger than the increase of FW-SDT's CPU time. That is because as the scale of the problem increasing, the runs of calculations within one generation of GA increase much more rapidly than that of FW-SDT. Therefore, FW-SDT is an effective and efficient algorithm to develop timetables and can be applied to the real scale network.

To solve the problem, the vehicle schedules are obtained simultaneously. If a vehicle trip can be carried out by more than one vehicle, we assume that it is carried out by the first arrived vehicle. From Table 10, the corresponding vehicle schedule is: Shuttle 121 needs 5 vehicles, vehicle 1: 1–6–11; vehicle 2: 2–7–12; vehicle 3: 3–8–13; vehicle 4: 4–9–14; vehicle 5: 5–10–

Fig. 10. The real life study area.

Table 8
A comparison of costs of different solutions.

	Current				GA				FW-SDT			
	C_D	C_P	C_{TR}	C_T	C_D	C_P	C_{TR}	C_T	C_D	C_P	C_{TR}	C_T
Shuttle 121	2211.3	152.6	5718.2	8074.6	2996.0	366.4	2335.1	5697.5	2275.3	338.0	1816.2	4429.5
Shuttle 102	3976.3	5926.3	4223.8	14201.3	5468.8	248.0	2849.1	8565.9	4314.1	225.5	2564.7	7104.3
Shuttle 31	529.8	0.4	1308.5	1838.4	704.9	0.6	18.6	724.1	718.0	0.7	11.3	730.0
Shuttle 101	3619.0	1677.2	7043.1	12337.6	3847.2	396.6	4219.4	8463.3	3811.9	196.4	4004.9	8013.2
Network	10336.4	7756.5	18293.7	36452.0	13017.0	1011.5	9422.3	23450.8	11119.3	760.6	8397.1	20277.0

Table 9
Runs of calculations and CPU time of GA, FW and SDT.

	C_D	C_P	C_{TR}	$\partial C_D/\partial T$	$\partial C_P/\partial T$	$\partial C_{TR}/\partial T$	CPU time (s)
GA	$1.23 * 10^6$	$1.35 * 10^6$	$1.19 * 10^6$	0	0	0	995.9
FW	$6.13 * 10^4$	$6.75 * 10^4$	0	$6.13 * 10^4$	$2.03 * 10^5$	0	389.5
SDT	$1.99 * 10^5$	$2.19 * 10^5$	$1.93 * 10^5$	$1.88 * 10^4$	$6.21 * 10^4$	$7.29 * 10^4$	

Table 10
Original timetable and the optimized timetable solved by FW-SDT in the real life example.

Trip no.	Shuttle 121		Shuttle 102		Shuttle 31		Shuttle 101	
	Original	Optimized	Original	Optimized	Original	Optimized	Original	Optimized
1	6:35	6:34	6:30	6:30	6:45	6:30	6:30	6:31
2	6:50	6:42	6:44	6:38	6:55	6:39	6:45	6:43
3	7:02	6:55	6:55	6:46	7:05	6:49	7:00	6:55
4	7:10	7:02	7:05	6:58	7:15	6:56	7:15	7:00
5	7:18	7:11	7:15	7:06	7:25	7:05	7:28	7:06
6	7:26	7:24	7:25	7:18	7:35	7:16	7:38	7:14
7	7:35	7:37	7:35	7:29	7:45	7:31	7:45	7:24
8	7:45	7:45	7:45	7:38	7:55	7:42	7:52	7:38
9	7:55	7:52	7:55	7:51	8:05	7:52	8:00	7:46
10	8:05	7:59	8:05	8:02	8:15	8:00	8:08	7:53
11	8:15	8:14	8:15	8:11	8:25	8:08	8:15	8:01
12	8:25	8:24	8:25	8:21	8:35	8:17	8:22	8:09
13	8:35	8:32	8:38	8:36	8:45	8:28	8:30	8:19
14	8:45	8:43	8:52	8:53	8:55	8:43	8:40	8:32
15	9:00	8:53	–	–	–	–	8:50	8:43
16	–	–	–	–	–	–	9:00	8:58

15. Shuttle 102 needs 5 vehicles, vehicle 1: 1–6–9–14; vehicle 2: 2–7–12; vehicle 3: 3–8–13; vehicle 4: 4–10; vehicle 5: 5–11. Shuttle 31 needs 2 vehicles, vehicle 1: 1–3–5–7–9–11–13; vehicle 2: 2–4–6–8–10–12–14. Shuttle 101 needs 6 vehicles, vehicle 1: 1–7–13; vehicle 2: 2–8–14; vehicle 3: 3–9–15; vehicle 4: 4–10–16; vehicle 5: 5–11; vehicle 6: 6–12.

7. Conclusions

Based on real-time passenger arrival distributions, the paper presents an optimal synchronized timetable development problem for community shuttles linked with metro service, subject to two constraints, i.e., the vehicle capacity constraint and the fleet size constraint. The first constraint is treated as a soft constraint and the latter is handled by a proposed timetable generating method. To solve the problem, a genetic algorithm (GA) is first presented, and a Frank–Wolfe algorithm combined with a heuristic algorithm of shifting departure times (FW-SDT) is put forward. The two algorithms are tested in simulated and real life examples. Important conclusions are summarized as follows:

(1) For a continuous passenger arrival distribution function,$f(t)$, the schedule delay cost is a continuously differentiable function. If the penalty function,$g(x)$, is continuously differentiable, the overloading penalty is also a continuously differentiable function. The sum of the two functions is continuously differentiable or not only depends on $g(x)$, while not related to $f(t)$. If $g(x)$ is continuously differentiable, FW-SDT can be used to solve the problem. FW-SDT is suitable for any continuous passenger's arrival distribution.
(2) Both GA and FW-SDT are feasible to solve the problem, and FW-SDT is more effective and efficient than GA. The efficiency of FW-SDT is attributed to much less runs of related calculations within one generation. From the simulated network example, FW-SDT performs more stable than GA, and always obtains a better solution.
(3) When the fleet size is given, increasing the number of vehicle trips blindly is unwise. When the number of vehicle trips closes to its upper bound or the passenger arrival distribution has dramatic fluctuations, increasing vehicle trips may not reduce the passenger's travel cost and may even make it increase. In addition, increasing vehicle trips will definitely increase the operating cost. Determining the number of vehicle trips needs to consider its upper bound and the passenger arrival distribution.
(4) When the number of vehicle trips is given, increasing the fleet size will make the passenger's total cost decrease significantly at first. As the fleet size increasing further, the curve of the total cost becomes stable because of the same solution obtained theoretically.
(5) In the real life example, GA reduces the total cost by 35.7%, while FW-SDT reduces it by 44.4%. The CPU time of FW-SDT is less than that of GA and increases slower when the scale of the problem getting larger. Therefore, FW-SDT is an easy and efficient way to optimize timetables and can be practically applied.

Future research could be done from the following aspects:

(1) Accurate, real-time, and time-varying passenger arrival distribution functions at stops/stations are very difficult to obtain in practice. It will cost a lot of manpower and resources to conduct the travel investigation and collect the passenger's trip data. Moreover, if the simulated arrival distribution is not accurate enough to represent the actual case, it is difficult to achieve an optimal result. Therefore, future research should analyze to what extent this limitation can affect the effectiveness of the model output for real bus operation purpose. Hopefully, with the application of some advanced technologies in the transit system, this problem may become easier to solve in the near future.
(2) Stochastic bus travel time will impact on the optimization process in various aspects and travel time variability may also affect how passengers choose a path. The issue of travel time reliability is neglected in the study because of the relatively simple network topology and low demand in the community in our study. Moreover, the dwell time is also set to be a constant in the paper, while it actually varies with the number of boarding or alighting. Therefore, considering the stochastic travel time and the varying dwell time in the optimization process is an important issue for the future.
(3) This study focuses on optimizing the timetable when the number of vehicle trips and the fleet size are both given. Actually, when the fleet size is given, it should have a certain number of vehicle trips that ensures the passenger's travel cost minimized. Exploring the relationship between the number of vehicle trips and the fleet size is a future work.

Acknowledgements

This paper is sponsored by the NSFC Project (No. 71131001) and the National Basic Research Program of China (Nos. 2012CB725403 and 2012CB725405). The authors would like to acknowledge the valuable comments from Dr. Yang Cheng and the help from the other members in Wisconsin ITS Program.

Appendix A

Proposition 1. *The schedule delay cost C_D^l is a continuously differentiable function, for $\forall f(t)$.*

Proof. First, we proof its continuity. For route l, if we shift a random departure time to the right slightly, which means T_n^l is replaced by $T_n^l + \Delta t$, for $\forall n \in (1, N^l)$, $\Delta t \to 0^+$, and then $t_{n-1,n}^l$ is replaced by $t_{n-1,n}^l + \gamma/(\beta+\gamma)\cdot\Delta t$, $t_{n,n+1}^l$ is replaced by $t_{n,n+1}^l + \beta/(\beta+\gamma)\cdot\Delta t$, and the new timetable is $\boldsymbol{T}^{l'}$. According to Eq. (7), the difference between $C_D^l(\boldsymbol{T}^l)$ and $C_D^l(\boldsymbol{T}^{l'})$ is:

$$\begin{aligned}
C_D^l(\boldsymbol{T}^{l'}) - C_D^l(\boldsymbol{T}^l) = {} & \beta\sum_m\sum_d \int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l} \left(t - T_{n-1}^l - s_m^l\right) f_{md}^l(t)dt \\
& + \gamma\sum_m\sum_d \int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} \left(T_n^l + \Delta t + s_m^l - t\right) f_{md}^l(t)dt \\
& + \beta\sum_m\sum_d \int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} \left(t - T_n^l - \Delta t - s_m^l\right) f_{md}^l(t)dt \\
& + \gamma\sum_m\sum_d \int_{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l}^{T_{n+1}^l+s_m^l} \left(T_{n+1}^l + s_m^l - t\right) f_{md}^l(t)dt \\
& - \beta\sum_m\sum_d \int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+s_m^l} \left(t - T_{n-1}^l - s_m^l\right) f_{md}^l(t)dt - \gamma\sum_m\sum_d \int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} \left(T_n^l + s_m^l - t\right) f_{md}^l(t)dt \\
& - \beta\sum_m\sum_d \int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} \left(t - T_n^l - s_m^l\right) f_{md}(t)dt - \gamma\sum_m\sum_d \int_{t_{n,n+1}^l+s_m^l}^{T_{n+1}^l+s_m^l} \left(T_{n+1}^l + s_m^l - t\right) f_{md}(t)dt
\end{aligned} \tag{A.1}$$

Eq. (A.1) can be also written as:

$$\begin{aligned}
C_D^l(\boldsymbol{T}^{l'}) - C_D^l(\boldsymbol{T}^l) = {} & \beta\sum_m\sum_d \left(\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l} t f_{md}^l(t)dt - \int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+s_m^l} t f_{md}^l(t)dt \right. \\
& \left. + \int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} t f_{md}^l(t)dt - \int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} t f_{md}^l(t)dt \right) - \gamma\sum_m\sum_d \left(\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} t f_{md}^l(t)dt \right. \\
& \left. - \int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} t f_{md}^l(t)dt + \int_{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l}^{T_{n+1}^l+s_m^l} t f_{md}^l(t)dt - \int_{t_{n,n+1}^l+s_m^l}^{T_{n+1}^l+s_m^l} t f_{md}^l(t)dt \right) \\
& - \beta\sum_m\sum_d \left((T_{n-1}^l + s_m^l) \left(\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt - \int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+s_m^l} f_{md}^l(t)dt \right) \right) \\
& + \gamma\sum_m\sum_d \left((T_{n+1}^l + s_m^l) \left(\int_{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l}^{T_{n+1}^l+s_m^l} f_{md}^l(t)dt - \int_{t_{n,n+1}^l+s_m^l}^{T_{n+1}^l+s_m^l} f_{md}^l(t)dt \right) \right) \\
& - \beta\sum_m\sum_d \left((T_n^l + s_m^l) \left(\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt - \int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} f_{md}^l(t)dt \right) \right. \\
& \left. + \Delta t \int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt \right) + \gamma\sum_m\sum_d \left((T_n^l + s_m^l) \left(\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt - \int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} f_{md}^l(t)dt \right) \right. \\
& \left. + \Delta t \int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt \right)
\end{aligned} \tag{A.2}$$

From the above analysis, there are $t_{n-1,n}^l + \gamma/(\beta+\gamma)\Delta t + s_m^l \to t_{n-1,n}^l + s_m^l$, $T_n^l + \Delta t \to T_n^l$, $t_{n,n+1}^l + \beta/(\beta+\gamma)\Delta t + s_m^l \to t_{n,n+1}^l + s_m^l$, so every term in Eq. (A.2) infinitely closes to 0. That means $\lim_{\Delta t\to 0^+}\left(C_D^l(\boldsymbol{T}^{l'}) - C_D^l(\boldsymbol{T}^l)\right) = 0$, for $\forall n \in (1, N^l)$. For $n = 1$ or N^l, the case is similar, and we do not give the proof here. Similarly, there also exists $\lim_{\Delta t\to 0^-}\left(C_D^l(\boldsymbol{T}^l) - C_D^l(\boldsymbol{T}^{l'})\right) = 0$, so C_D^l is a continuously function.

Then we proof its derivability. When $\Delta t \to 0^+$, there is:

$$\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^+} = \lim_{\Delta t\to 0^+} \frac{C_D^l(\boldsymbol{T}^{l'}) - C_D^l(\boldsymbol{T}^l)}{\Delta t} \tag{A.3}$$

Based on Eq. (A.2), there is:

$$\begin{aligned}\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^+} =\ &\beta\sum_m\sum_d\left(\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l} tf_{md}^l(t)dt-\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+s_m^l} tf_{md}^l(t)dt+\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} tf_{md}^l(t)dt-\int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} tf_{md}^l(t)dt\right)\Big/\Delta t\\
&-\gamma\sum_m\sum_d\left(\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} tf_{md}^l(t)dt-\int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} tf_{md}^l(t)dt+\int_{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l}^{T_{n+1}^l+s_m^l} tf_{md}^l(t)dt-\int_{t_{n,n+1}^l+s_m^l}^{T_{n+1}^l+s_m^l} tf_{md}^l(t)dt\right)\Big/\Delta t\\
&-\beta\sum_m\sum_d\left((T_{n-1}^l+s_m^l)\left(\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt-\int_{T_{n-1}^l+s_m^l}^{t_{n-1,n}^l+s_m^l} f_{md}^l(t)dt\right)\right)\Big/\Delta t\\
&+\gamma\sum_m\sum_d\left((T_{n+1}^l+s_m^l)\left(\int_{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l}^{T_{n+1}^l+s_m^l} f_{md}^l(t)dt-\int_{t_{n,n+1}^l+s_m^l}^{T_{n+1}^l+s_m^l} f_{md}^l(t)dt\right)\right)\Big/\Delta t\\
&-\beta\sum_m\sum_d\left((T_n^l+s_m^l)\left(\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt-\int_{T_n^l+s_m^l}^{t_{n,n+1}^l+s_m^l} f_{md}^l(t)dt\right)\Big/\Delta t+\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt\right)\\
&+\gamma\sum_m\sum_d\left((T_n^l+s_m^l)\left(\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt-\int_{t_{n-1,n}^l+s_m^l}^{T_n^l+s_m^l} f_{md}^l(t)dt\right)\Big/\Delta t+\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt\right)\end{aligned} \tag{A.4}$$

When $\Delta t\to 0$, $F(\Delta t)\to 0$, there is: $\lim_{\Delta t\to 0}F(\Delta t)/\Delta t=\lim_{\Delta t\to 0}(F')|_{\Delta t}$ (*L'Hopsital's rule*), and Eq. (A.4) can be written as:

$$\begin{aligned}\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^+} =\ &\sum_m\sum_d\begin{pmatrix}\frac{\beta\gamma}{\beta+\gamma}\left(t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l\right)+\frac{\gamma^2}{\beta+\gamma}\left(t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l\right)\\ -\frac{\beta\gamma}{\beta+\gamma}\left(T_{n-1}^l+s_m^l\right)-\frac{\gamma^2}{\beta+\gamma}\left(T_n^l+s_m^l\right)\end{pmatrix}\cdot f\left(t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l\right)\\
&+\sum_m\sum_d\begin{pmatrix}\frac{\beta^2}{\beta+\gamma}\left(t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l\right)+\frac{\beta\gamma}{\beta+\gamma}\left(t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l\right)\\ -\frac{\beta\gamma}{\beta+\gamma}\left(T_{n+1}^l+s_m^l\right)-\frac{\beta^2}{\beta+\gamma}\left(T_n^l+s_m^l\right)\end{pmatrix}\cdot f\left(t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l\right)\\
&-\beta\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt+\gamma\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt\end{aligned} \tag{A.5}$$

Then substitute $t_{n-1,n}^l=(\beta T_{n-1}^l+\gamma T_n^l)/(\beta+\gamma)$, $t_{n,n+1}^l=(\beta T_n^l+\gamma T_{n+1}^l)/(\beta+\gamma)$ into Eq. (A.5), the first two terms of Eq. (A.5) are both equal to 0, and finally:

$$\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^+} = -\beta\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt+\gamma\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt \tag{A.6}$$

Similarly, there is: $\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^-}=-\beta\int_{T_n^l+\Delta t+s_m^l}^{t_{n,n+1}^l+\beta/(\beta+\gamma)\Delta t+s_m^l} f_{md}^l(t)dt+\gamma\int_{t_{n-1,n}^l+\gamma/(\beta+\gamma)\Delta t+s_m^l}^{T_n^l+\Delta t+s_m^l} f_{md}^l(t)dt$, and $\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^-}=\left.\frac{\partial C_D^l}{\partial T_n^l}\right|_{\Delta t\to 0^+}$. So C_D^l is differentiable. All the above formulations are not subject to $f(t)$, so C_D is a continuously differentiable function, for $\forall f(t)$. □

Appendix B

Proposition 2. *If $g(x)$ is continuously differentiable, the overloading penalty function C_P^l is a continuously differentiable function, for $\forall f(t)$.*

Proof. First, we proof its continuity. For route l with the timetable $\boldsymbol{T}^l$, the loading on the segment r is denoted as $Ld_r^l(\boldsymbol{T}^l)$. Shift T_n^l to $T_n^l+\Delta t$, for $\forall n\in(2,N^l-1)$, $\Delta t\to 0^+$. Similarly, the difference between $Ld_r^l(\boldsymbol{T}^l)$ and $Ld_r^l(\boldsymbol{T}^{l'})$ can be formulated according to Eq. (9):

$$\begin{aligned} Ld_r^l(\boldsymbol{T}^{l'}) - Ld_r^l(\boldsymbol{T}^l) = &\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(\int_{t^l_{n-2,n-1}+s^l_r}^{t^l_{n-1,n}+\gamma/(\beta+\gamma)\Delta t+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n-2,n-1}+s^l_r}^{t^l_{n-1,n}+s^l_r} f^l_{md}(t)dt\right) \\ &- \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(\int_{t^l_{n-2,n-1}+s^l_r}^{t^l_{n-1,n}+\gamma/(\beta+\gamma)\Delta t+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n-2,n-1}+s^l_r}^{t^l_{n-1,n}+s^l_r} f^l_{md}(t)dt\right) \\ &+ \sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(\int_{t^l_{n-1,n}+\gamma/(\beta+\gamma)\Delta t+s^l_r}^{t^l_{n,n+1}+\beta/(\beta+\gamma)\Delta t+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n-1,n}+s^l_r}^{t^l_{n,n+1}+s^l_r} f^l_{md}(t)dt\right) \\ &- \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(\int_{t^l_{n-1,n}+\gamma/(\beta+\gamma)\Delta t+s^l_r}^{t^l_{n,n+1}+\beta/(\beta+\gamma)\Delta t+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n-1,n}+s^l_r}^{t^l_{n,n+1}+s^l_r} f^l_{md}(t)dt\right) \\ &+ \sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(\int_{t^l_{n,n+1}+\beta/(\beta+\gamma)\Delta t+s^l_r}^{t^l_{n+1,n+2}+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n,n+1}+s^l_r}^{t^l_{n+1,n+2}+s^l_r} f^l_{md}(t)dt\right) \\ &- \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(\int_{t^l_{n,n+1}+\beta/(\beta+\gamma)\Delta t+s^l_r}^{t^l_{n+1,n+2}+s^l_r} f^l_{md}(t)dt - \int_{t^l_{n,n+1}+s^l_r}^{t^l_{n+1,n+2}+s^l_r} f^l_{md}(t)dt\right) \end{aligned} \tag{B.1}$$

Eq. (B.1) denotes that shifting the nth($n \in (2, N^l - 1)$) departure time of $\boldsymbol{T}^l$, three vehicles' loading will be affected, i.e. the $(n-1)$th, the nth and the $(n+1)$th vehicle. The six terms in Eq. (B.1) reflect the change of the three vehicles' loading. Every term in Eq. (B.1) infinitely closes to 0, and there is $\lim_{\Delta t\to 0^+}\left(Ld_r^l(\boldsymbol{T}^{l'}) - Ld_r^l(\boldsymbol{T}^l)\right) = 0$. The cases for $n = 1, 2, N^l - 1, N^l$ are similar. If $g(x)$ is continuously differentiable, there is: $\lim_{\Delta t\to 0^+}\left(C_P^{l,r}(\boldsymbol{T}^{l'}) - C_P^{l,r}(\boldsymbol{T}^l)\right) = \lim_{\Delta t\to 0^+}\left(g\left(Ld_r^l(\boldsymbol{T}^{l'})\right) - g\left(Ld_r^l(\boldsymbol{T}^l)\right)\right) = 0$. Similarly, $\lim_{\Delta t\to 0^-}\left(C_P^{l,r}(\boldsymbol{T}^l) - C_P^{l,r}(\boldsymbol{T}^{l'})\right) = 0$. As $\boldsymbol{dis}^l$ is not related to $\boldsymbol{T}^l, C_P^{l,r}(\boldsymbol{T}^l) = \boldsymbol{C_P^{l,R}}(\boldsymbol{T}^l)\cdot(\boldsymbol{dis}^l)^T$ is a continuously function.

Then we proof its derivability. When $\Delta t \to 0^+$, there is:

$$\left.\frac{\partial C_P^{l,r}}{\partial T_n^l}\right|_{\Delta t\to 0^+} = \lim_{\Delta t\to 0^+}\frac{C_P^{l,r}(\boldsymbol{T}^{l'}) - C_P^{l,r}(\boldsymbol{T}^l)}{\Delta t} = \lim_{\Delta t\to 0^+}\frac{\left.C_P^{l,r}(\boldsymbol{T}^{l'})'\right|_{\Delta t}}{\left.\Delta t'\right|_{\Delta t}} = \lim_{\Delta t\to 0^+}\left.C_P^{l,r}(\boldsymbol{T}^{l'})'\right|_{\Delta t} \tag{B.2}$$

Based on Eq. (B.1), there is:

$$\begin{aligned} \left.\frac{\partial C_P^{l,r}}{\partial T_n^l}\right|_{\Delta t\to 0^+} = &\frac{\partial g\left(Ld^l_{r,n-1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n-1}(\boldsymbol{T}^l)}\cdot\left(\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}\left(t^l_{n-1,n}+\frac{\gamma}{\beta+\gamma}\Delta t+s^l_r\right) - \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}\left(t^l_{n-1,n}+\frac{\gamma}{\beta+\gamma}\Delta t+s^l_r\right)\right)\cdot\frac{\gamma}{\beta+\gamma} \\ &+ \frac{\partial g\left(Ld^l_{r,n}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n}(\boldsymbol{T}^l)}\cdot\left(\begin{aligned}&\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2}\left(f^l_{md}\left(t^l_{n,n+1}+\tfrac{\beta}{\beta+\gamma}\Delta t+s^l_r\right)\tfrac{\beta}{\beta+\gamma} - f^l_{md}\left(t^l_{n-1,n}+\tfrac{\gamma}{\beta+\gamma}\Delta t+s^l_r\right)\tfrac{\gamma}{\beta+\gamma}\right) \\ &-\sum_{d=1}^{M_2^r}\sum_{m=1}^{Md}\left(f^l_{md}\left(t^l_{n,n+1}+\tfrac{\beta}{\beta+\gamma}\Delta t+s^l_r\right)\tfrac{\beta}{\beta+\gamma} - f^l_{md}\left(t^l_{n-1,n}+\tfrac{\gamma}{\beta+\gamma}\Delta t+s^l_r\right)\tfrac{\gamma}{\beta+\gamma}\right)\end{aligned}\right) \\ &+ \frac{\partial g\left(Ld^l_{r,n+1}(\boldsymbol{T}^l)\right)}{\partial Ld^l_{r,n+1}(\boldsymbol{T}^l)}\cdot\left(-\sum_{m=1}^{M_1^r}\sum_{d=1}^{M_2} f^l_{md}\left(t^l_{n,n+1}+\frac{\beta}{\beta+\gamma}\Delta t+s^l_r\right) + \sum_{d=1}^{M_2^r}\sum_{m=1}^{Md} f^l_{md}\left(t^l_{n,n+1}+\frac{\beta}{\beta+\gamma}\Delta t+s^l_r\right)\right)\cdot\frac{\beta}{\beta+\gamma} \end{aligned} \tag{B.3}$$

The $\left.\frac{\partial C_P^{l,r}}{\partial T_n^l}\right|_{\Delta t\to 0^-}$ can be obtained similarly, and there is $\left.\frac{\partial C_P^{l,r}}{\partial T_n^l}\right|_{\Delta t\to 0^-} = \left.\frac{\partial C_P^{l,r}}{\partial T_n^l}\right|_{\Delta t\to 0^+}$. So if $g(x)$ is differentiable, C_P^l is a differentiable function. All the above formulations are not subject to $f(t)$, so if $g(x)$ is continuously differentiable, C_P^l is a continuously differentiable function, for $\forall f(t)$. $\square$

References

Adamski, A., 1998. Simulation support tool for real-time dispatching control in public transport. Transp. Res. 32A (2), 73–87.

Alshalalfah, B., Shalaby, A., 2012. Feasibility of flex-route as a feeder transit service to rail stations in the suburbs: Case study in Toronto. J. Urban Plan. Develop. 138 (1), 90–100.

Ceder, A., 1986. Methods for creating bus timetables. Transp. Res. 21A (1), 59–83.

Ceder, A., 2001. Public transport scheduling. In: Hensher, D., Button, K. (Eds.), Handbooks in Transport-Handbook 3: Transport Systems and Traffic Control. Elsevier Ltd, pp. 539–558.

Ceder, A., 2002. Urban transit scheduling: framework, review, and examples. ASCE J. Urban Plan. Develop. 128 (4), 225–244.

Ceder, A., 2003. Public transport timetabling and vehicle scheduling. In: Lam, W., Bell, M. (Eds.), Advanced Modeling for Transit Operations and Service Planning. Elsevier Ltd, pp. 31–57.

Ceder, A., 2009. Stepwise multi-criteria and multi-strategy design of public transit shuttles. J. Multi-Crit. Dec. Anal. 16 (1-2), 21–38.
Ceder, A., 2013. Integrated smart feeder/shuttle transit service: simulation of new routing strategies. J. Adv. Transp. 47 (6), 595–618.
Ceder, A., Wilson, N.H.M., 1986. Bus network design. Transp. Res. 20B (4), 331–344.
Ceder, A., Golany, B., Tal, O., 2001. Creating timetables with maximal synchronization. Transp. Res., Part A 35, 913–928.
Chakroborty, P., Deb, K., Subrahmanyam, P.S., 1995. Optimal scheduling of urban transit systems using genetic algorithms. J. Transp. Eng. 121 (6), 544–553.
Chang, S.K., Schonfeld, P.M., 1991. Multiple period optimization of bus transit systems. Transp. Res., Part B 25B, 453–478.
Chien, S., 2000. Optimal feeder bus routes on irregular street networks. J. Adv. Transp. 34, 213–248.
Chien, S., 2005. Optimization of headway, vehicle size and route choice for minimum cost feeder service. Transp. Plan. Technol. 28, 359–380.
Chien, S., Spasovic, L.N., Elefsiniotis, S.S., Chhonkar, R.S., 2001a. Evaluation of feeder bus systems with probabilistic time-varying demands and non-additive time costs. Transportation Research Record 1760,Transportation Research Board, Washington, DC, pp. 47–55.
Chien, S., Yang, Z., Hou, E., 2001b. Genetic algorithm approach for transit route planning and design. J. Transp. Eng. 127, 200–207.
Chien, S., Daripally, S.K., Kim, K., 2007. Development of a probabilistic model to optimize disseminated real-time bus arrival information for pre-trip passengers. J. Adv. Transp. 41 (2), 195–215.
Dessouky, M., Hall, R., Nowroozi, A., Mourikas, K., 1999. Bus dispatching at timed transfer transit stations using bus tracking technology. Transp. Res. 7C (4), 187–208.
Gallo, G., Di-Miele, F., 2001. Dispatching buses in parking depots. Transp. Sci. 35 (3), 322–330.
Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., Yagiura, M., 2005. Effective local search algorithms for routing and scheduling problem with general time-window constraints. Transp. Sci. 39 (2), 206–232.
Kwan, C.M., Chang, C.S., 2008. Timetable synchronization of mass rapid transit system using multi-objective evolutionary approach. IEEE Trans. Syst., Man, Cybernet., Part C 38 (5), 636–648.
Mohaymany, A.S., Gholami, A., 2010. Multimodal feeder network design problem: ant colony optimization approach. J. Transp. Eng. 136 (4), 323–331.
Newell, G.F., 1971. Dispatching policies for a transportation route. Transp. Sci. 5 (1), 91–105.
Osuna, E.E., Newell, G.F., 1972. Control strategies for an idealized public transportation system. Transp. Sci. 6 (1), 52–72.
Palma, A.D., Lindsey, R., 2001. Optimal timetables for public transportation. Transp. Res. Part B 35, 789–813.
Peeters, L., Kroon, L., 2001. A cycle based optimization model for the cyclic railway timetabling problem. In: Voss, S., Daduna, J.R. (Eds.), Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, vol. 505. Springer-Verlag, pp. 275–296.
Quadrifoglio, L., Dessouky, M.M., Palmer, K., 2007. An insertion heuristic for scheduling mobility allowance shuttle transit (MAST) services. J. Scheduling. 10 (1), 25–40.
Quadrifoglio, L., Hall, R.W., Dessouky, M.M., 2006. Performance and design of mobility allowance shuttle transit services: bounds on the maximum longitudinal velocity. Transp. Sci. 40 (3), 351–363.
Shrivastava, P., Dhingra, S.L., 2001. Development of feeder routes for suburban railway stations using heuristic approach. J. Transp. Eng. 127 (4), 334–341.
Shrivastava, P., Dhingra, S.L., 2002. Development of coordinated schedules using genetic algorithms. J. Transp. Eng. 128, 89–96.
Shrivastava, P., O'Mahony, M., 2006. A model for development of optimized feeder routes and coordinated schedules – a genetic algorithms approach. Transp. Policy 13 (5), 413–425.
Shrivastava, P., O'Mahony, M., 2009. Use of a hybrid algorithm for modeling coordinated feeder bus route network at suburban railway station. J. Transp. Eng. 135 (1), 1–8.
Wirasinghe, S.C., 1990. Re-examination of Newell's dispatching policy and extension to a public transportation route with many to many time varying demand. In: Transportation and Traffic Theory. Elsevier Ltd, pp. 363–378.
Wirasinghe, S.C., 2003. Initial planning for an urban transit system. In: Lam, W., Bell, M. (Eds.), Advanced Modeling for Transit Operations and Service Planning. Elsevier Ltd, pp. 1–29.
Wong, C.W., Yuen, W.Y., Fung, K.W., Leung, M.Y., 2008. Optimizing timetable synchronization for rail mass transit. Transp. Sci. 42 (1), 57–69.
Szeto, W.Y., Wu, Y., 2011. A simultaneous bus route design and frequency setting problem for Tin Shui Wai, Hong Kong. Euro. J. Operat. Res. 209, 141–155.
Xiong, J., Guan, W., Song, L.Y., Huang, A.L., Shao, C.F., 2013. Optimal routing design of community shuttle joint for metro stations. J. Transp. Eng. 139 (12), 1211–1223.
Zhao, F., Zeng, X., 2007. Optimization of user and operator cost for large-scale transit network. J. Transp. Eng. 133, 240–251.