A Data-Driven Timetable Optimization of Urban Bus Line Based on Multi-Objective Genetic Algorithm

Jinjun Tang, Yifan Yang, Wei Hao, *Member, IEEE*, Fang Liu, and Yinhai Wang

***Abstract*— Reasonable bus timetable can reduce the operating costs of bus company and improve the quality of bus services. A data-driven method is proposed to optimize bus timetable in this study. Firstly, a bi-objective optimization model is constructed considering minimize the total waiting time of passengers and the departure times of bus company. Then, Global Positioning System (GPS) trajectories of buses and passenger information collected from Smart Card are fused and applied to calculate the key parameters or variables in optimization model, including time-dependent travel time, bus dwell time and passenger volume. Finally, by adopting a specific coding scheme, an improved Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is designed to quickly search Pareto optimal solutions. Furthermore, an experiment is conducted in Beijing city from one bus line to validate the effectiveness of the proposed method. Comparing with empirical scheduling method and traditional single-objective optimization base on GA, the results show that the proposed model could quickly provide high-quality and reasonable timetable schemes for the administrator in urban transit system.**

***Index Terms*— Urban transit, bus timetable, multi-objective, data-driven method, non-dominated sorting genetic algorithm-II (NSGA-II).**

NOMENCLATURE

I	the maximum number of service times the bus fleet can provide
J	the number of bus stations on bus line
K	the number of time periods that varies the average vehicle speed and passenger volume
i	the index of bus service ($1 \le i \le I$)
j	the index of bus station ($1 \le j \le J$)
k	the index of time period ($1 \le k \le K$)
t_i	decision variable, the departure time of the i-th bus leaving the first station
N_i	binary variable (0 if t_{i-1} is equal to t_i and 1 otherwise)
$H_{\min}$	the minimum interval between two adjacent vehicles arriving at a station
$H_{\max}$	the maximum interval between two adjacent vehicles arriving at a station
W_f	the departure time of the first bus leaving the first station
W_l	the departure time of the last bus leaving the first station
$R_j(t_i)$	given the departure time t_i at the first station, $R_j(t_i)$ is the time of the i-th bus arrives at j-th station
SP_k	speed breakpoint at which the speed changes, and $C_0 = W_f$, $C_K = W_l$
$TP^k_{j,j+1}$	travel time breakpoint of the path between the j-th and $(j+1)$-th station in the k-th time period
$V^k_{j,j+1}$	average speed of the path between the j-th and $(j+1)$-th station in the k-th time period
$L_{j,j+1}$	distance between the j-th and $(j+1)$-th station
$y^k_{j,j+1}$	travel time of the path between the j-th and $(j+1)$-th station in the k-th time period
$Y_{j,j+1}(t)$	the travel time function from the j-th station to the $(j+1)$-th station with the start time t. Given a start time t, a corresponding travel time can be obtained.
D^i_j	bus dwell time of the i-th bus at the j-th station
PB^i_j	the number of passengers boarding on the i-th bus at the j-th station
PA^i_j	the number of passengers alighting on the i-th bus at the j-th station
M	the maximum value of two adjacent interval paying by card
m	the value of interval between two adjacent paying events

Manuscript received February 11, 2020; revised July 2, 2020 and September 3, 2020; accepted September 8, 2020. Date of publication September 29, 2020; date of current version March 31, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 71701215 and Grant 51808057; in part by the Innovation-Driven Project of Central South University under Grant 2020CX041; in part by the Hunan Provincial Natural Science Foundation under Grant 2018JJ3553, Grant 2019JJ30026, and Grant 2020JJ4752; and in part by the Young Elite Scientists Sponsorship Program by Hunan Province of China under Grant 2018RS3074. The Associate Editor for this article was X. Qu. *(Corresponding author: Wei Hao.)*

Jinjun Tang and Yifan Yang are with the Smart Transport Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China (e-mail: jinjuntang@csu.edu.cn; xxxiouchen@163.com).

Wei Hao and Fang Liu are with the School of Transportation Engineering, Changsha University of Science and Technology, Changsha 410205, China (e-mail: haowei@csust.edu.cn; rcliufang@163.com).

Yinhai Wang is with the Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: yinhai@uw.edu).

Digital Object Identifier 10.1109/TITS.2020.3025031

$P\ (m)$	distribution type of the interval sequence paying by card
$\mu_1,\ \lambda_1$	distribution parameters of interval between two neighboring boarding events
$\mu_2,\ \lambda_2$	distribution parameters of interval between two neighboring alighting events
$\varphi_j(k)$	passenger arrival rate, person/minutes
$\gamma_j(k)$	passenger alighting rate, person/minutes
$G_j(i-1,i)$	interval time between the $(i-1)$-th and i-th bus arriving at the j-th station
Q_j^i	the total number of passengers on the i-th bus after leaving the j-th station
CAP	the maximum number of passengers a bus can carry
CB_j^i	the number of passengers waiting for the next bus due to insufficient seats on the previous one
f_1	the actual number of bus service times
f_2	the total waiting time of passengers
POP	Population size of NSGA-II
P_c	Crossover rate of GA
P_m	Mutation rate of GA
$tour$	Number of candidates for the tournament selection method
Gen	Iteration times of NSGA-II
ξ_1	The operating cost coefficient of completing one service (yuan/vehicle)
ξ_2	The waiting time cost coefficient of passengers (yuan/minutes)

I. Introduction

PUBLIC transit is an effective solution to alleviate urban congestion, protect the environment and save social resources. It is necessary to improve service quality and operational efficiency of transit system, and then attract more citizens from private vehicles to public transit. Generally, increasing the number of buses and departing frequency could enhance the quality of service but obviously add budget in operation. So, in the actual operation process, developing a reasonable bus scheduling plan based on data-driven methods is an effective strategy to improve the quality of services in the demand responsive public transport (DRPT) and enhance the operating efficiency of public transit companies.

Designing reasonable timetables is the basis of bus scheduling scheme. Bus timetable is a sequence of time to determine when the buses leave the terminal station to serve passengers. The purpose of designing reasonable bus timetable is not only to meet the passenger demand, but also to consider the operating cost of bus companies. It is a multi-objective optimization problem to meet two conflict parts: at the peak hours with highest passenger volume, the time interval of two adjacent buses tends to be reduced to shorten the passengers waiting time, but this would lead to increase of operating costs and the number of serving buses at the same time. However, increasing the number of buses will lead to excessive capacity during the non-peak hours, resulting in low utilizing rate of buses. Therefore, it is necessary to control the actual departure intervals within a reasonable range based on multi-objective programming method.

In the traditional bus scheduling system, the departure time intervals of each two buses is usually fixed, which are regulated by experienced engineers. Actually, the schedule of each bus line is closely related to multi factors such as passenger flow, bus dwell time, average travel time between stations and road traffic conditions. With the rapid development of detection and communication technology, automatic data collection systems have widely been used on buses, which include automatic fare collection devices (Smart Cards) record passenger payment information and the Global Positioning System (GPS) is equipped to collect trajectories of buses. These data sources are widely used in passenger demand prediction [1] and bus scheduling optimization [2]. However, there still exist challenges in how to effectively extract related factors from multi-source data to solve the problem of timetable optimization for urban transit system. This article proposes a data-driven timetable optimization method, and the main works of this study are as follows:

(1) Optimization model. A multi-objective timetable optimization model is constructed to minimize the passenger waiting time and number of departure times. For the constraints, operating period, interval limitation, arrival time and vehicle carrying capacity are considered in the model.
(2) Data-driven method. Three input variables for the optimization model are calculated from Smart Card and GPS data sources, including time-dependent travel time between two adjacent stations during a day, boarding and alighting passenger volume and number of passengers on the bus, and bus dwell time of each bus stop.
(3) Algorithm design. Since the decision variable in this study is the final timetable, an improved NSGA-II algorithm with a special coding scheme is designed to solve the multi-objective problem. The proposed algorithm considers characteristics of bus timetable and can quickly obtain high quality Pareto solutions.
(4) Experiments and comparison. Smart Card and GPS datasets in a real bus line were collected and explored to estimate input parameters and variables in the experiments. A comparison is implemented between the Pareto optimal solutions with experienced method and single-objective optimization. The experimental results can provide a scientific support for decision-making to balance the interests between passenger demand and bus companies.

The main contributions of this article are three aspects: (i) Introduce a concept of the Time-dependent Vehicle Routing Problem (TDVRP) into the background of this research, and construct the travel time function based on data analysis; (ii) Develop an optimization model to minimize waiting times and service times, considering the bus dwell time at each station and the passengers who fail to board the bus due to overload; (iii) Design the NSGA-II algorithm with a special encoding scheme to generate Pareto solutions, which reduces the difficulty of encoding and decoding. The experimental results show that the proposed optimization method is reasonable and effective.

The rest of this article is organized include: Section II introduces related research for bus scheduling problem. Section III introduces the data source used in optimization. Optimization model and multi-objective algorithm are presented in Section IV. In Section V, we apply a case study to testify the proposed method and conduct the parameter sensitivity analysis. Section VI summarizes the conclusion of this article.

II. Related Work

The research works for the public transit system planning can be divided into four parts or stages [3], [4]: (1) *transportation route network design* (TRND), (2) *setting timetable*, (3) *vehicle scheduling*, (4) *crew scheduling*, and (5) *real-time control*. Next, we briefly introduce the main content of each stage. The purpose of the TRND is to design a set of bus routes for the public transit system [5]–[7]. Yu *et al.* [6] established a direct passenger density model for traffic network design and solved the proposed model with ant colony optimization (ACO). Timetable design is to determine when each bus leaves the first bus station [8]–[10]. Gkiotsalitis and Alesiani [8] combined the uncertainty of passenger demand and travel time to model the robust timetable, with the objective to minimize the possible losses in the worst case. Vehicle scheduling aims at assigning a set of trips to each vehicle, and each trip has fixed start and end time in the bus schedule [11]–[13]. Tang *et al.* [11] proposed a deficit function (DF) methodology to reduce the required number of vehicles for covering all trips, and used some strategies to adjust a few trips. In addition, more introduction of DF method is in [12]. Multiple depot vehicle scheduling problem (MDVSP) is studied in [13], in which a two-phase method is applied to find a set of scheduling solutions that covers every trip. Crew scheduling is to assign the efficient duty combination for crews to cover all activities [14]–[17]. Fuentes *et al.* [15] proposed an ad-hoc mathematical decomposition method based on time-personnel clustering to solve the crew scheduling problem. Real-time control strategy is implemented to ensure the efficient service by real-time supervision and management during the operation period [18], [19]. Delgado *et al.* [18] proposed a mathematical programming model to control vehicles operating on a transit corridor with the goal of minimizing total delays.

There are two types of the setting timetable problem, one is to determine the departure frequency and fleet size allocated to different routes in each period, and the other is to formulate the departure time of buses at the first stops on the bus network. Also, some studies combine the timetable problem with other issues. The solution method of setting timetable includes analytical simulation [20], [21], exact algorithms [22], [23] and heuristics [24]–[31]. Based on the cooperative game theory, Dai *et al.* [20] proposed a bus holding strategy to improve bus bunching phenomenon and adjust bus headways. Sivakumaran *et al.* [21] used an analytical method to minimize the weighted sum of waiting time, transfer time and operator cost. Mesa *et al.* [22] produced the cluster pre-processing method to optimize the user inconvenience. Martínez *et al.* [23] combined the Tabu search algorithm with CPLEX to minimize total travel time, with the constraints of passenger satisfaction and fleet size. A computer-based timetable and bus schedule optimization method was proposed to improve the utilization of buses and the crew assignment in [24]. Varga *et al.* [25] proposed a multi-objective model to optimize the headway and timetable of buses, and predict the arrival time at a bus stop by estimating the dwell time. In addition to buses, timetable optimization has applications in other areas. Nitisiri *et al.* [26] presented a Parallel Multi-objective Evolutionary Algorithm for the railway train timetabling problem with two objectives. One is to minimize the number of operating cycles during operation, and the other is to minimize the average waiting time of the rail passengers. In [27], a Particle Swarm Optimization and Simulated Annealing (PSO-SA) algorithm is designed to optimize the train timetable problem, with the goal of improving the transfer synchronization between different railway lines. Yang *et al.* [28] proposed a two-stage stochastic programming model for the metro timetable, aiming at minimizing the total energy consumption of traction. Louwerse and Huisman [29] focused on the railway timetable in case of major disruptions, considering partial and complete blockade of the railway line. An integer programming model is proposed in [30] to generate train timetable, in which genetic algorithm based on a binary coding method is used.

On the one hand, the time-dependent travel time was rarely considered in the timetable optimization problem. The travel speed in [32] is constant whether in different road segments or in different time periods. For rail transit, the travel time between two consecutive stops is set as a constant value. While in urban transport, it is affected by traffic conditions. An exception is [2]. However, although it defines the time-dependent traffic and gives the corresponding numerical matrix, it essentially takes a fixed value into the model for calculation. In fact, TDVRP is a common type of vehicle routing problem [33]. We can incorporate the calculation method of its travel time function into this research.

On the other hand, although a large number of multi-objective models have been proposed for setting timetable, relatively few studies used Evolutionary Algorithms (EAs) to solve it. Moreover, most studies convert the multi-objective problem into one or multiple single-objective optimization problems, based on the weight sum approach. Also, there are some researches using EAs methods in public transport field. In the airport scheduling problem, Lu *et al.* [34] established a timetable optimization model by considering the satisfaction of airport loyal passengers on bus service quality, and then applied NSGA-II to generate Pareto optimal solutions. For the vehicle scheduling problem, Zuo *et al.* [35] first produced a set of candidate vehicle blocks, and then designed an improved NSGA-II to obtain the Pareto solutions for vehicle scheduling. However, it is difficult to find the application in setting timetable. Therefore, this article uses a multi-objective genetic algorithm to generate multiple Pareto optimal solutions for the timetable optimization problem, which makes it easier to balance the costs between passengers and operating companies.

III. Overview of Optimization

A. Model Framework

Figure 1 introduces an example to clearly describe the optimization problem. The solid circles indicate the stations where the buses have passed, and the hollow circles indicate the remaining stations they will travel. We consider the difference between the departure time of two adjacent vehicles at the station as *interval*, such as t_2 - t_1. The size of the interval affects the total time the passengers wait for a bus and the number of vehicles that the fleet required to maintain daily operations. When passengers get on and off at each station, the number of passengers on board changes with the variation of time and station.

With the goal is to balance the interests of passengers and bus company, a bi-objective optimization model is established to generate optimal timetable by fusing Smart Card and GPS data sets. The timetable optimization model is composed of three parts: (1) decision variables: departure time of each vehicle leaving the first station; (2) objective functions: two optimization goals, one is the number of departure times for daily operations, and the other is the waiting time for passengers at all stations; (3) and constraints: operating period, interval limitation, arrival time and vehicle carrying capacity. The constraints are designed to establish the relationship between decision variables and optimization objectives. In the constraints, four input variables are estimated based on historical datasets including: time-dependent travel time ($Y_{j,j+1}(t)$), bus dwell time (D^i_j), and passenger flow ($\varphi_j(k)$ and $\gamma_j(k)$). For $Y_{j,j+1}(t)$, representing the travel time function from the j-th bus stop to the $(j+1)$-th with the start time t, we first match the GPS and Smart Card data and then accurately estimate the arrival time of the vehicles at each station; for D^i_j, denoting the bus dwell time of the i-th bus at the j-th station, it refers to the time a public transit bus spends at a stop without moving, and is determined by Smart Card dataset; for the passenger arrival and alighting rate $\varphi_j(k)$ and $\gamma_j(k)$, we calculate the number of people waiting and alighting at each station according to the Smart Card dataset, and then estimate the passenger number on the bus.

B. Data Source

Two kind of data source including GPS and Smart Card are used in the application of optimization, and the detailed information of data is described in Table I. The GPS navigator records the latitude, longitude and instantaneous speed of each bus every 3 seconds, and the Smart Card dataset contains the station and time of boarding and alighting events for each unique Smart Card. According to the information from two kinds of data source, the parameters and variables in the optimization model are estimated, which will be shown in next section.

C. Variables Analysis

1) Time-Dependent Travel Time: In this study, travel time is used to estimate the arrival time of buses, and then to calculate the total waiting time of the passengers. Travel time can reflect road congestion and it varies with different time periods during a day. In order to describe the variation of arrival time of the buses and waiting time of the passengers under daily time periods, the time-dependent travel time estimation method is conducted in this study. An example shown in Figure 2 is used to explain the calculation process of this method.

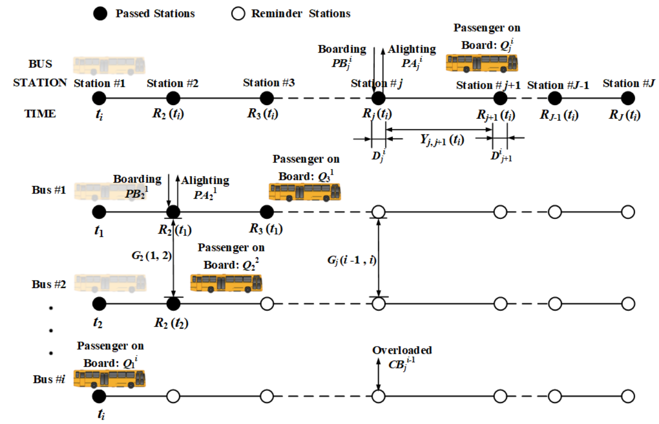


Fig. 1. An illustration of daily bus operation.

TABLE I
GPS Dataset and Smart Dataset

GPS dataset		Smart Card dataset	
Content	Remarks	Content	Remarks
Line ID	The line number of the bus	Line ID	The line number of the bus
Vehicle ID	Unique code for the vehicle	Vehicle ID	Unique code for the vehicle
Time	The time of collecting the location, format 'yyyy-mm-dd-hh-mm -ss'	Card ID	The unique number of the Smart Card
Longitude	The longitude of the vehicle	Mark Station	The boarding station number of the Smart Card
Latitude	The latitude of the vehicle	Mark Time	The boarding time of the Smart Card, format 'yyyy-mm-dd-hh-mm-ss'
Speed	The instantaneous speed of the vehicle	Trade Station	The alighting station number of the Smart Card
Direction	The running direction of the vehicle	Trade Time	The alighting time of the Smart Card, format 'yyyy-mm-dd-hh-mm-ss'

The driving route between two adjacent stations is called a *path*, and each path has its unique time-dependent travel time function to represent its variation during a day. Taking the edge between the Station #4 and the Station #5 as an example, the distance between the two stations is 400, and different departure time corresponds to different speeds. Then, we can obtain the time-dependent travel time function, which is represented by travel time graph in the right of Figure 2.

The instantaneous speed in the GPS dataset is used to determine the average travel speed of two adjacent bus stops in each time period. Then, a time-dependent travel time function is defined by the travel speed graph, which represents the relationship between the travel time and the departure time from the j-th station to the $(j+1)$-th station. In this study, there

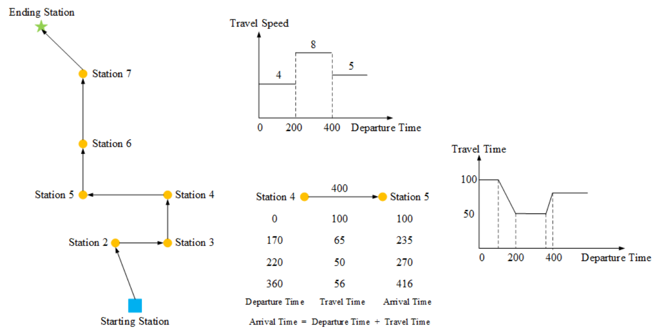


Fig. 2. An illustration of the time-dependent travel time.

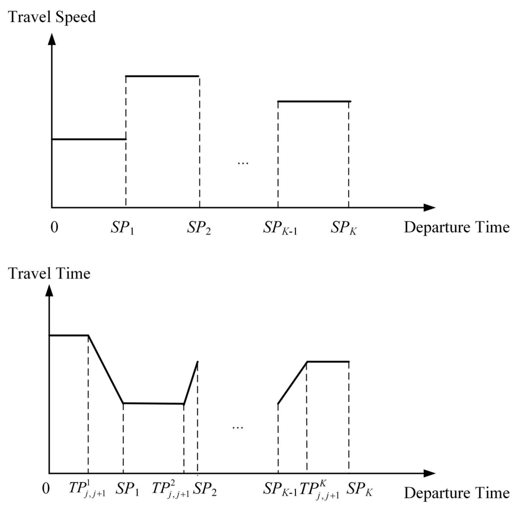


Fig. 3. The time-dependent travel time function.

are total of K time periods, and the travel speed in each time period is constant. Let $y^k_{j,j+1}$ denotes the travel time with constant speed from the j-th stop to the $(j+1)$-th stop in the k-th time period, then we have $y^k_{j,j+1} = L_{j,j+1} \ / \ V^k_{j,j+1}$. Besides, the junction point between two time periods in which the speed changes is called *speed breakpoint*, which is manually determined and represented by SP_k $(k = 1, 2, 3 \ldots K)$ shown in Figure 3. There also exists *travel time breakpoint* in the travel time function, denoted as $TP^k_{j,j+1}$ $(k = 1, 2, 3 \ldots K)$, which is the latest departure times to ensure an arrival at the j-th station before SP_k.

$$TP^k_{j,j+1} = \begin{cases} SP_k - y^k_{j,j+1}, & 1 \le k < K \\ SP_K, & k = K \end{cases} \tag{1}$$

With the speed and travel time breakpoints, the travel time function is updated as

$$Y_{j,j+1}(t) = \begin{cases} y^k_{j,j+1}, \quad SP_{k-1} < t \le TP^k_{j,j+1} \\ y^k_{j,j+1} + \dfrac{t - TP^k_{j,j+1}}{SP_k - TP^k_{j,j+1}}(y^{k+1}_{j,j+1} - y^k_{j,j+1}), \\ \qquad TP^k_{j,j+1} < t \le SP_k \end{cases} \tag{2}$$

2) Passenger Volume: According to the information contains in the Smart Card dataset, the boarding and alighting passengers at each station can be extracted, and they can be converted into the arrival rate and the alighting rate in units of "people/minute". Therefore, the longer the interval between arriving times of two adjacent buses at the same station, the more passengers would board or alight at the bus station. Let $H_{\max}$ denotes the maximum interval between two adjacent vehicles, t_i is the departure time when the i-th bus leaving the first station, and $R_j(t_i)$ represents the time of the i-th bus arrives at j-th station. Set $G_j(i-1, i)$ as the interval between the arrival time of $(i-1)$-th and i-th bus at the j-th station, then the time gap between two arrival times of adjacent buses at the same station is defined as:

$$G_j(i-1, i) = \begin{cases} H_{\min} & if \ i = 1 \\ t_i - t_{i-1} & if \ i > 1 \ and \ j = 1 \\ R_j(t_i) - R_j(t_{i-1}) & if \ i > 1 \ and \ j > 1 \end{cases} \tag{3}$$

Headway is a measurement of the time from the tip of one bus to the tip of the later one in a transit system. Thus, $G_j(i-1, i)$ can be regarded as the headway of two adjacent vehicles at the j-th station. From Eq. (3), we can see that the headway will vary with the travel time and the terminal dwell time. It is impossible to know the operating situation before the 1-st bus. Therefore, we assume that the passenger waiting time of the 1-st vehicle at each station is $H_{\min}$.

The operation period of the bus company is $[W_f, W_l]$, which is divided into a total of K time periods. The i-th bus arrives at the j-th station in the k-th time period. Based on the arrival time $R_j(t_i)$, the index of temporal partition k is shown as follows,

$$k = \left\lceil \frac{K \cdot (R_j(t_i) - W_f)}{W_l - W_f} \right\rceil \tag{4}$$

Based on the waiting time interval $G_j(i-1, i)$ and the passenger alighting rate $\gamma_j(k)$, the number of passengers getting off the bus PA^i_j can be obtained. Obviously, the number of alighting passengers cannot exceed the total number of passengers on the bus Q^i_j. Therefore, the number of people getting off the i-th bus at the j-th station is shown as:

$$PA^i_j = \min\left\{\gamma_j(k) \cdot G_j(i-1, i)\,, \quad Q^i_{j-1}\right\} \tag{5}$$

If $PA^i_1 = 0$, it means no one gets off the bus at the first bus station. As for the number of boarding passengers, due to the limited seats and standing space, there has a maximum capacity limitation indicated by CAP. In addition, there are PA^i_j passengers who get off the i-th vehicle at the j-th bus stop, so the number of passengers that can be accommodated in the i-th vehicle is CAP + PA^i_j - Q^i_j. Let PB^i_j is the number of passengers boarding on the i-th bus. Since the number of people on board should not exceed the remaining capacity of the i-th bus, PB^i_j is defined as:

$$PB^i_j = \min\left\{\varphi_j(k) \cdot G_j(i-1, i), \ CAP + PA^i_j - Q^i_{j-1}\right\} \tag{6}$$

$PB^i_J = 0$ represents that no one boards on the bus at the last bus station. Usually, the capacity of bus can meet the passenger demand, that is, passengers can take the first bus they are waiting for. However, during peak hours, the number of passengers increases and be more than remaining capacity of bus, many passengers have to wait for the second bus for boarding. The number of passengers who fail to board the i-th bus and need to wait for the $(i+1)$-th bus at the j-th bus station is CB_j^i, denoted as:

$$CB_j^i = \varphi_j(k) \cdot G_j(i-1,i) - PB_j^i \tag{7}$$

where $CB_j^0 = 0$ means that people on the 1-st bus successfully board the first bus they are waiting for at all bus stations. We assume that passengers who have not successfully boarded the first bus they are waiting for have priority to take the next bus. Considering the passengers who need to take the $(i+1)$-th bus, the number of boarding passengers in Eq. (6) is updated as:

$$PB_j^i = \min\left\{\varphi_j(k) \cdot G_j(i-1,i) + CB_j^{i-1}, \; CAP + PA_j^i - Q_{j-1}^i\right\} \tag{8}$$

Given the number of people getting on and off the i-th bus at the j-th station, the total number of passengers on the i-th bus when the bus passes the j-th station can be obtained as Q_j^i.

$$Q_j^i = Q_{j-1}^i + PB_j^i - PA_j^i \tag{9}$$

where $Q_0^i = 0$ denotes that there are no passengers in the bus when the bus arrives at the first bus station.

3) Bus Dwell Time: This section describes how to the use the number of passengers to estimate the bus dwells time, which is the time period the bus stays at the station. There are many factors that affect the bus dwell time, such as the number of passengers getting on and off, payment ways, the type of bus stations, etc., and the most important factor is the number of boarding and alighting passengers. When the bus arrives at the station, if the front door and the back door open at the same time, the boarding and alighting events occur simultaneously. Sometimes the number of passengers getting on the bus is more or less than the number of passengers getting off the bus. Therefore, the dwell time can be defined as the maximum value between the total boarding time and alighting time of passengers. Passengers usually board on the bus and pay their fees one by one, so the payment time can be approximately estimated by arithmetic sequence. Defining m as time gap between two adjacent passenger using Smart Card, and assuming the number of passengers at m obeys a certain probability distribution $P(m)$. Given a Smart Card payment time gap m, the probability P of m can be calculated. Then, the consuming time of the passengers whose boarding time is $m \cdot PB \cdot P_1(m)$. Similarly, the alighting time of passengers is $m \cdot PA \cdot P_2(m)$. Finally, we can calculate the dwell time of the i-th bus at the j-th station as follows:

$$D_j^i = \max\left\{PB_j^i \cdot \sum_{m=1}^{M} mP_1(m), \quad PA_j^i \cdot \sum_{m=1}^{M} mP_2(m)\right\} \tag{10}$$

IV. Methodology

In this section, we first establish a bi-objective timetable optimization model, and describe the process of building the mathematical model in detail. Then, an improved NSGA-II algorithm is applied to solve the multi-objective problem, and the algorithm steps based on the optimization model are designed.

A. Optimization Model

This section establishes a MOP (multi-objective programming) model for bus timetable problem. The maximum number of departure time for the bus fleet can provide is denoted as I. In actual operation, bus company can meet the demand of passenger volume without using all vehicles, and ensure a certain passenger satisfaction at the same time. Therefore, the purpose of the first objective function f_1 is to determine the actual number of departure time according to the operation of bus company, and it cannot exceed I. The decision variables of the optimization model are the departure moments t_i of each bus at the 1-st bus station, so there are a total of I departure time. Set N_i as a binary variable, we have

$$N_i = \begin{cases} 0 & if \;\; t_i = t_{i-1} \\ 1 & otherwise \end{cases} \tag{11}$$

where $N_i = 0$ means that no vehicle departs from the 1-st bus stop at t_i, and $N_i = 1$ means that there is a vehicle service passengers at t_i. Assume that there is a service vehicle at t_1, so the actual number of departure times is

$$f_1 = 1 + \sum_{i=2}^{I} N_i \tag{12}$$

Another objective function f_2 is the sum of the passengers waiting time at all the bus stops, which consists of two parts: one is the waiting time of passengers who normally take the first bus they are waiting for, and the second part is the additional waiting time of passengers who wait for the next bus to board because of the bus overloading. The precise moment when each passenger arrives at the bus stop is difficult to detect and collect. When the departure interval is small, the average passenger waiting time in most studies is considered to be half of the interval between two adjacent vehicles arriving at the station. For the PB_j^i, the average waiting time for each passenger can be defined as $G_j(i-1,i)/2$. For the CB_j^i, the time passengers wait for the $(i-1)$-th bus has been considered in PB_j^{i-1}. Since the time interval for the passenger waiting for the i-th bus is in the range of $[R_j(t_{i-1}), \; R_j(t_i)]$, the waiting time for this part is the headway of buses, i.e. $G_j(i-1,i)$. Therefore, the total waiting time of passengers is defined as:

$$f_2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(PB_j^i \cdot \frac{G_j(i-1,i)}{2} + CB_j^i \cdot G_j(i-1,i)\right) \tag{13}$$

Accordingly, an MOP model with two objectives can be proposed to optimize the bus timetable problem. The purpose is to search a set of Pareto solutions for timetable that minimizes the service times of bus fleet and the passengers waiting time:

$$\min \ f_1 \tag{14}$$
$$\min \ f_2 \tag{15}$$
$$\text{Subject to: } t_1 = W_f \tag{16}$$
$$t_I \leq W_l \tag{17}$$
$$0 \leq t_{i+1} - t_i \leq H_{\max} \tag{18}$$
$$R_j(t_i) = \begin{cases} t_i \quad if \ \ j = 1 \\ R_{j-1}(t_i)+D^i_{j-1}+Y_{j-1,j}(R_{j-1}(t_i)) \\ \quad otherwise \end{cases} \tag{19}$$
$$Q^i_j = Q^i_{j-1} + PB^i_j - PA^i_j \tag{20}$$

The Eq.(14)-(15) respectively denote the optimization goals, i.e. minimize the service times of bus and minimize the total waiting time of passenger. The constraints (16) and (17) are set according to the actual departure time of the first bus and the last bus. Constraint (18) indicates that there is an upper limit for the time interval between the departure moments of two adjacent buses. For the time when the i-th bus arrives at the j-th station $R_j(t_i)$, its value corresponds to the time in the final timetable when $j = 1$, otherwise it needs to be determined via the time-dependent travel time between two consecutive stations. Given the arrival time of the previous bus stop $R_{j-1}(t_i)$, the bus dwell time at the previous bus stop D^i_{j-1} and the travel time from the $(j-1)$-th station to the j-th station, we can infer the time when the bus arrived at the j-th bus stop, as shown in constraints (19). The number of passengers getting on and off at the j-th bus stop can be determined in the Section III-C(2), thus we define constraint (20) to calculate the number of passengers on the i-th bus when the bus leaves the station.

B. Multi-Objective Algorithm

An improved NSGA-II is proposed to solve the optimization of bus timetable with multiple objectives. A special coding scheme is designed to reduce computation time, and different ways of crossover and mutation operators are applied to maintain the diversity of the Pareto solutions. The procedure of improved NSGA-II includes following several steps [36].

Step 1: Set $gen = 0$, let archive set $Z = \emptyset$ to record the Pareto solutions, and initialize population to generate POP individuals to compose $P(gen)$;
Step 2: Calculate the objective function value, non-dominated rank and crowding distance for each individual in $P(gen)$;
Step 3: Apply genetic algorithm to $P(gen)$, that is, perform tournament selection, crossover and mutation to generate offspring population $Q(gen)$;
Step 4: Combine $P(gen)$ and $Q(gen)$ to create a new population $S(gen)$;
Step 5: Compute the objective function value of all individuals in $S(gen)$, and then obtain the non-dominated rank and crowding distance of each individual in $S(gen)$;

x_1	x_2	......	x_i	......	x_I
t_1	t_2	......	t_i	......	t_I

Fig. 4. Coding scheme of chromosome.

Step 6: Select POP individuals from $S(gen)$ based on Elitism to form $P(gen+1)$. Let archive set $Z = P(gen+1)$ and simultaneously record the non-dominated rank and crowding distance of each individual;
Step 7: Let $gen = gen + 1$, and return to Step3 until the stop number of iterations is reached.

The above steps of NSGA-II have three key sub-routines. First, non-dominated rank method stratifies population $S(gen)$ according to the number of non-inferior solutions for each individual, and the non-dominated ranking values in each level are the same. Second, all individuals in the same layer are resorted based on the values of crowding distance. Third, the elitism strategy is adopted to select the superior individuals in the parent population directly into the offspring population, preventing the loss of Pareto solution. Accordingly, the final recorded solutions in Z are considered to be the Pareto optimal solutions. Detailed design of NSGA-II is introduced as follows.

1) Solution Coding: We design a real-coded scheme based on timetable, which has a direct correspondence with the decision variables of the optimization model. As shown in Figure 4, the gene x_i on chromosome X refers to the departure time t_i of the i-th bus in the timetable. Since I is the maximum number of services the bus fleet can provide, the length of a chromosome is set to be I. However, if two or more identical genes appear in a chromosome, only one gene has practical significance, and the remaining identical genes are deleted in the final solutions. Thus, the length of the final timetable is actually less than I.

2) Population Initialization: The value corresponding to the gene x_i on the chromosome X is set as $w \in [W_f, Wl]$, where W_f and Wl are the start and end times of the timetable, respectively. Referring to the constraints (18) and (19), the values of x_1 and x_I are determined, and $t_{i+1} - t_i \geq 0$ means that the genes on the chromosome X should be arranged in ascending order, i.e. $x_1 = W_f$, $x_I = Wl$, and $x_1 \leq x_2 \leq \ldots \leq x_I$. With the generation rule of a single chromosome, *pop* chromosomes can be randomly generated to form an initial population $P(gen)$.

3) Fast Non-Dominated Rank and Crowding Distance: Each solution has multiple objective function values that cannot be directly compared with each other, that is, *Pareto solutions*. More detail of Pareto can be found in [37]. For the Pareto solutions, we first calculate the objective function values of each chromosome, then use the fast non-dominated rank algorithm to sort all the individuals in $P(gen)$, and finally each chromosome will be marked with a non-dominated level F_i. Crowding distance is a measure that describes how close an individual is to its neighbors. Accordingly, the population in the same layer F_i needs to be resorted by their crowding distance.

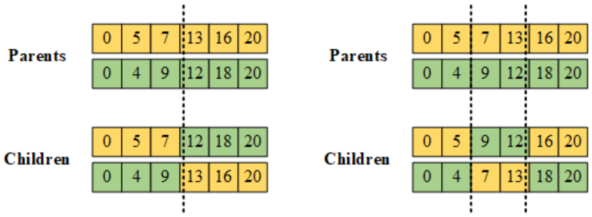


Fig. 5. Diagram for two crossover rules: single-point and two-point.

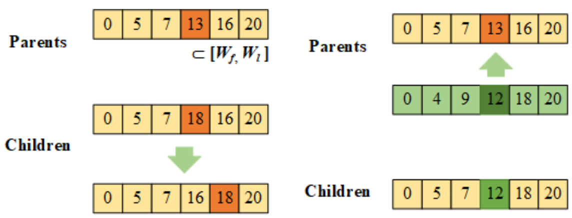


Fig. 6. Diagram for two mutation rules.

4) Tournament Selection, Crossover and Mutation: The tournament selection method are used to select two parent individuals. If the rank levels are the same, we prefer to select the one with a larger crowding distance. Then, the selected two individuals are implemented crossover and mutation operations to obtain two child individuals.

The probability P_c is used to judge whether the parent chromosome needs crossover. If the two parents are crossover, a single-point crossover is performed with a probability of 0.5, and two-point crossover with a probability of 0.5 as shown in Figure 5. Each gene position on the chromosome is mutated according to the probability P_m, and there are two ways to finish choice. One is that reselecting a value from $[W_f, W_l]$ randomly for the gene x_i, and the other is to replace x_i with a gene at the same position of another chromosome. Then, the chromosome is reordered after mutation as shown in Figure 6.

Repeat the crossover and mutation operations until the number of offspring population $Q(gen)$ reaches the pre-setting values.

5) Elitism-Based Selection: Based on the elite retention strategy, the population $R(gen)$ synthesized by the parent $P(gen)$ and the offspring $Q(gen)$ is selected to form a new parent population $P(gen+1)$, as shown in Figure 7.

6) Updating Archive Set: The Pareto solutions selected in $P(gen+1)$ are recorded by the archive set Z, and then the new parent population is iterated again. When the limitation of iterations is reached, the non-dominated solutions in the archive set are considered to be the Pareto optimal solutions.

V. Experiment and Result Analysis

A. Case Study

The Line 445 in transit system in Beijing City is used as experiment to verify the effectiveness of the proposed optimization model. There are 19 bus stops on this line, with the first bus leaves at 5:00 and the last bus departs at 22:40. The GPS and Smart Card data for all vehicles on the line were collected from November 1, 2017 to November 30, 2017.

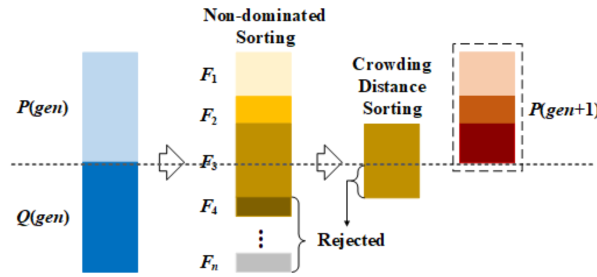


Fig. 7. Diagram for elitism-based selection.

We select the data during the period of 17:30-18:30 used in the case study. For the values of parameters, the input parameters are divided into two parts, (1) the input parameters of the optimized model, the time-dependent travel time, passenger volume and bus dwell time are estimated using aforementioned methods. Meanwhile, the distribution parameters, μ, λ and m, for boarding and alighting time in dwell time are determined based on the Smart Card dataset shown in Section IV. As the selected time period is 17:30-18:30, so to simplify the coding scheme and reduce the calculation time, we set $W_f = 0$ and $W_l = 60$. In the actual operation, the interval of two adjacent vehicles arriving at a station should be selected in [2], [20], so we set $H_{\min} = 2$ and $H_{\max} = 20$. The remaining parameters are set to $I = 18$, $J = 19$, $K = 4$, and $CAP = 50$. (2) for the parameters in the optimization algorithm NSGA-II, the $Gen = 400$ and $tour = 4$; for the POP, Pc and Pm, we conduct the sensitivity analysis to determine their values in Section V-C. The proposed optimization method is run on Matlab 2019a software in a computer with 4 GB memory, Intel i3 3.70 GHz CPU and Microsoft Windows 10.

B. Variables Estimation

According to Section III, the settings of several input parameters are driven by real data: (1) time-dependent travel time, (2) passenger flow, and (3) bus dwell time.

1) Time-Dependent Travel Time: We select the evening rush hour data of all vehicles on a bus line in the analysis. Since the mathematical model entirely considers the travel time from the first station to the last station, the travel time function needs to be extended to two hours. Then, a two-hour travel time step function diagram is shown in Figure 8. The x-axis represents the departure time of the vehicle leaving the current station, with a span of 30 minutes as speed breakpoints. In addition, the y-axis is the corresponding travel time, and each line represents a travel time function of the trip from the current station to the next station.

2) Passenger Volume: Since the Smart Card dataset contains information about every boarding and alighting event, we can directly count the passenger volume of bus line. First, data cleaning and site matching are performed on GPS and IC card data. Next, we divide the whole day into 144 time periods at 10-minute intervals, and then the number of passengers getting on and off at stations in each time period can be calculated. Taking the selected bus line as an example, the variation trend of passenger volume with time is extracted,

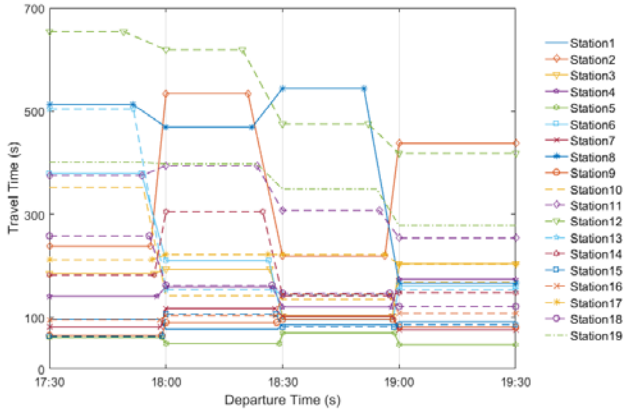


Fig. 8. Travel time distribution of different trips.

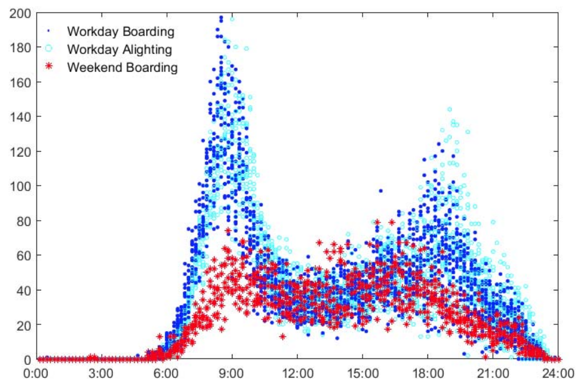


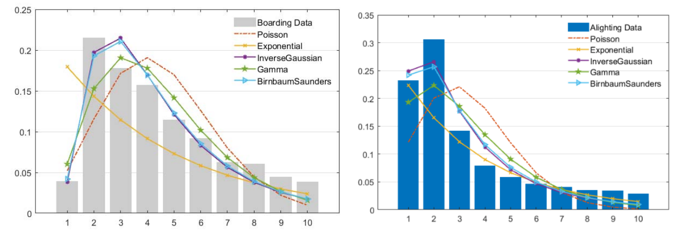
Fig. 9. Time-dependent passenger volume.

as shown in Figure 9. We only express three passenger distributions for comparison, namely, the boarding passenger flow on weekdays, the alighting passenger flow on weekdays and the boarding passenger flow on weekends. We can see that for the weekdays, both boarding and alighting passenger volume have obvious peaks in the morning and evening peak hours; for the weekends, the passenger volume during the day time vary slightly, with a slow decreasing trend after 5 pm. In addition, compared with the boarding flow and the alighting flow on weekdays, we can see that the distribution of alighting passenger flow deviates to the right, which is due to the alighting behavior occurs after the boarding behavior. Also, we can intercept passenger volume for a specific time period, and count the number of each category based on the index number of the bus stops. In this way, we can calculate the passenger arrival rate, alighting rate and number of passengers on the bus.

3) Bus Dwell Time: Passengers' payment events are independent of each other and are not affected by factors such as service routes and road congestion. Thus, one-month IC card data of 10 bus lines in Beijing are used to further estimate the bus dwell time. We calculate the time gaps of each two adjacent payment times and establish two histograms of the boarding time interval and the alighting time interval, respectively. Taking the alighting time interval as example, five different distribution models are used to fit the histogram and the values of R-square and RMSE are shown in Table II.

TABLE II
DISTRIBUTION FITTING EFFECT OF ALIGHTING DATA

Alighting Data	Poisson	Exponential	Inverse Gaussian	Gamma	Birnbaum Saunders
R-square	0.4422	0.7520	0.9341	0.8139	0.9213
RMSE	0.0683	0.0455	0.0235	0.0394	0.0257



(a) boarding time (b) alighting time

Fig. 10. The fitting results of time gaps distribution of boarding and alighting events.

We find that the Inverse Gaussian distribution has the largest R-square and the smallest RMSE, so the Inverse Gaussian distribution has the best fitting effect. In addition, the probability density function of the Inverse Gaussian distribution is

$$P(m) = \sqrt{\frac{\lambda}{2\pi m^3}} \exp\left\{\frac{-\lambda(m-\mu)^2}{2\mu^2 m}\right\} .m > 0, \quad \mu > 0, \quad \lambda > 0. \tag{21}$$

where μ is the mean and λ is the shape parameter. Similarly, for the boarding time interval, we obtain similar results, and we also use Inverse Gaussian function to fit the distribution of the boarding data.. The fitting results are shown in Figure 10. The x-axis represents the range of time intervals between two consecutive events, and the y-axis represents the distribution probability of m. For the boarding data, $\mu_1 = 4.45025$, $\lambda_1 = 11.9262$, and for the alighting data $\mu_2 = 3.30381$, $\lambda_2 = 5.40797$.

C. Parameter Sensitivity Analysis

The parameters need to be determined in NSGA-II include the crossover rate P_c, the mutation rate P_m, and the population size POP. To observe and compare the effect of each parameter on performance, five runs are implemented under different values of parameters, and the results of f_1 and f_2 at the two endpoints of the Pareto Front are recorded, the average values are listed from Table IV to Table III.

In Table IV, we set three different crossover rates, and run at 5 times for each value, then record the average Pareto solutions for $f1$ and $f2$ optimal, respectively. It can be seen that the results under three rates are similar. Thus, we can conclude that the size of the crossover rates has little influence on the search range of the solution space. The parameter setting of Pareto optimal solution is chosen the crossover rate with best performance, as shown in the bold in Table III.

Table IV expresses the endpoint values of the Pareto Front at different mutation rates, and we can see that the range of solution space becomes smaller when the mutation rate is small. The two objectives f_1 and f_2 conflict with each

TABLE III
AVERAGE BOUNDARY VALUES OF PARETO FRONT ABOUT CROSSOVER RATE P_c

P_c	When f_2 is optimal		When f_1 is optimal	
	f_1	f_2	f_1	f_2
0.70	17.2	35.2	5.6	195.8
0.80	**17.17**	**33.17**	**5.67**	**194.83**
0.90	17.4	34.4	5.8	199.8

TABLE IV
AVERAGE BOUNDARY VALUES OF PARETO FRONT ABOUT MUTATION RATE P_m

P_m	When f_2 is optimal		When f_1 is optimal	
	f_1	f_2	f_1	f_2
0.01	17	35	7.4	135.8
0.05	**17.17**	**33.17**	**5.67**	**194.83**
0.10	18	33.4	5.8	180.6
0.15	17.8	33.6	5.6	169.4

TABLE V
AVERAGE BOUNDARY VALUES OF PARETO FRONT ABOUT POPULATION SIZE *POP*

POP	When f_2 is optimal		When f_1 is optimal	
	f_1	f_2	f_1	f_2
200	16.6	36.6	6.6	151.4
300	17	37	6.83	172.5
400	**16.67**	**35.17**	**6.33**	**163.67**
500	16.8	35.8	7.2	184.6
600	17	34.4	7.2	188

other and cannot be optimal at the same time. In general, a reduction in one objective function leads to an increase in another objective function. If both f_1 and f_2 are simultaneously reduced, it means that a better Pareto solution appears. Since too many mutations will affect the stability of the Pareto solutions, we choose a relatively low mutation rate at 0.05.

The population size also affects the optimization performance. The corresponding results of five different population sizes are listed in Table V. We find that too much or little populations did not make the results better. At the minimum endpoint of f_2, as *POP* increases, the search boundary of the solution also expands. At the optimal endpoint of f_1, the search boundary of the solution is independent to *POP*. Combining this two situations, we choose $POP = 400$ as the best population size *POP*.

D. Experimental Results

Given the optimal parameter settings, we run the NSGA-II to solve the timetable optimization problem. After several experiments, we find that the results are adequately converged when 400 iterations are performed, and 5 runs of the Pareto optimal solution are recorded in Table VI. In each run, we can obtain multiple sets of Pareto optimal solutions, only the boundary values of the Pareto Front are illustrated in the table.

TABLE VI
PARETO OPTIMAL SOLUTIONS OF NSGA-II

Runs	When f_2 is optimal		When f_1 is optimal		Used Time (s)
	f_1	f_2	f_1	f_2	
1	17	35	6	211	996
2	18	35	6	170	924
3	17	33	6	157	952
4	17	35	6	178	948
5	16	37	5	218	993

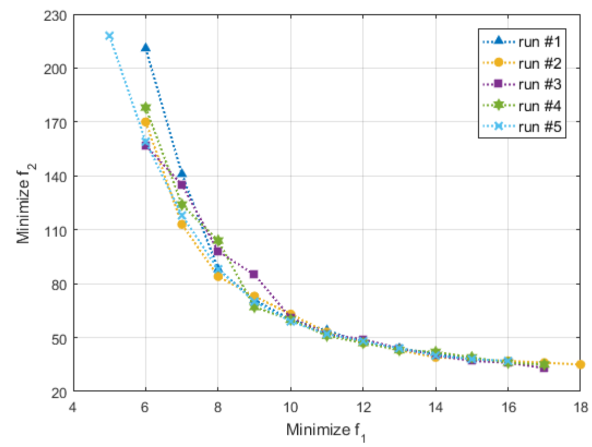


Fig. 11. Approximate Pareto Front of NSGA-II.

"When f_1 is optimal" indicates the respective values of the f_1 and f_2 when f_1 reaches the minimum. The first and sixth columns list the index of runs and the corresponding compute time, respectively. The second and third columns are the Pareto solutions when the passenger waiting time is minimum, while the fourth and fifth columns are the Pareto solutions when the number of departure times is optimal. We can see from the table that the range of f_1 and f_2 are [5], [18] and [33], [211], respectively. In addition, the boundary values of the Pareto Front for each run are similar, indicating that the performance of the multi-objective optimization algorithm is stable.

In order to clearly show the performance of the NSGA-II, the five-run solutions in Table VI are described in Figure 11 to allocate these Pareto optimal solutions in the objective space. In Figure 11, all the Pareto optimal solutions extend along the coordinate axis and are evenly distributed in the solution space. In addition, the Approximate Pareto Front is a concave curve, which tends to approach the origin of the coordinate. It can be seen that the five runs of Pareto optimal solutions all have good diversity and convergence. Therefore, we can conclude that the NSGA-II algorithm designed in this article has good performance to solve the system optimization problem.

The NSGA-II algorithm is a posterior method, that is, the preference of the decision maker is unknown before solving. In addition, we can directly obtain solutions with different decision preferences by running NSGA-II once, according to the proposed optimization method. By calculating the percentage difference of the objective value among different solutions, the decision maker can more intuitively weigh the two goals: the passenger waiting time and the service times of bus. Then, they can choose the best scheme according to

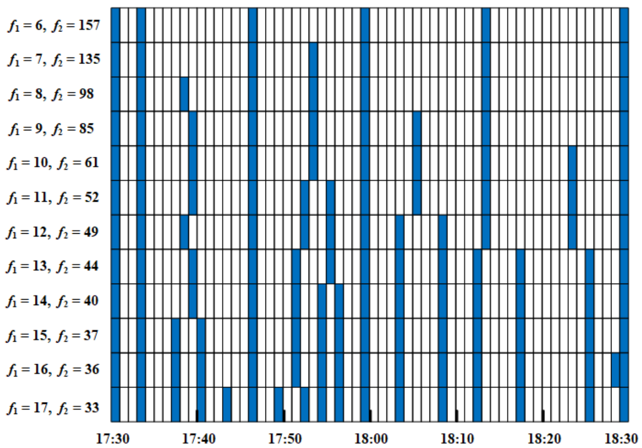


Fig. 12. Timetable schemes of the 3-th run via the proposed optimization model.

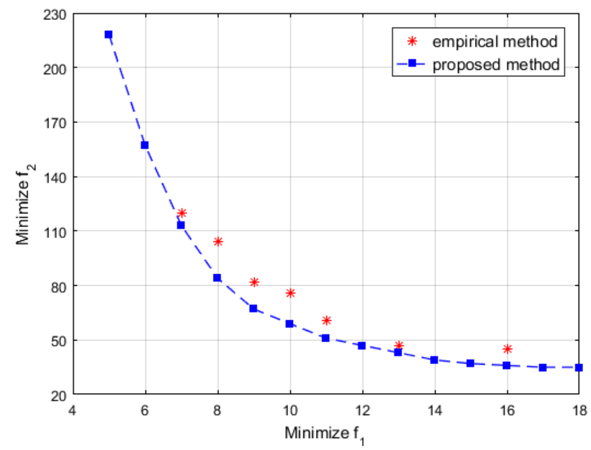


Fig. 13. Comparison between the proposed method and empirical scheduling scheme.

actual situations. For example, in Figure 11, suppose that 7 buses are used in the current period. In order to reduce the total passenger waiting time by more than 50%, what is the cheapest operation plan for the bus company? It can be seen that all the schemes with $f_1 \geq 10$ meet this requirement of reducing the waiting time by 50%. In this case, the decision maker chooses the timetable with $f_1 = 10$, which has the least expensive.

Taking the result of the third run as an example, the result can also be expressed as Figure 12. The x-axis represents the selected time period, and the y-axis represents all the Pareto optimal solutions of the third run in the Table VI. We divide the time period into cells in minutes, and the solid grid represents that a bus leaves the starting station at corresponding time. In Figure 12, we find that the timetable schemes are reasonable, because the departure time varies with the passenger flow, which indicates that the proposed optimization method can obtain high-quality scheduling schemes, and the designed algorithm has a good performance to generate Pareto optimal solutions.

E. Comparison

1) With Empirical Scheduling Method: Traditional bus scheduling is manually designed based on the experience. The empirical scheduling method first divides a day into several time periods, and then determines the departure frequency in each time period, so that an equal departure time interval can be obtained by dividing the length of each time period. Moreover, there is a maximum and minimum limit on the departure interval. Based on this equal interval method, we obtain several timetable schemes in the time period of 17:30-18:30 and compare with the proposed method, as shown in Figure 13. The red dots represent the value of the objective function according to the equal interval time method. When using heuristic algorithms to solve problems, it needs to run multiple times and then choose the smallest (or largest) solution or the average value as the final answer. So, we select the best solutions among the five-run programs to compare with the results obtained by empirical methods, as the blue line in Figure 13.

Figure 13 shows the objective values of the two timetable design methods. The horizontal axis represents the service times of the bus fleet during the selected time period, and the vertical axis represents the waiting time of passengers in hours. In Figure 13, all the blue dots are below the red ones, closer to the coordinate origin, and extends evenly outward along the coordinate axis. It means that the Pareto optimal solutions generated by the proposed optimization method dominates the solutions obtained by the empirical scheduling. In other words, the schemes introduced in this study have less passenger waiting time and fewer service times. Besides, the Pareto solutions obtained by our method are evenly spread at the Pareto Front and are closer to the Pareto Front, i.e. the solutions have good convergence and rich diversity. However, the solution based on the empirical scheduling method highly depends on the experience of manager. Moreover, the time required to formulate the timetable scheme is relatively long, and the departure intervals have no flexibility. The optimization method we proposed can obtain multiple sets of Pareto solutions with one run, so that the decision makers can choose the best solution based on a preference between fewer services times and less passenger waiting time. In summary, the proposed optimization method can obtain a better timetable plan.

2) With GA: There are many ways to solve the multi-objective optimization problems. Weighted-average is a widely used method that incorporates multiple objectives into several single objectives. Since the number of bus service times and the passenger waiting time are two different units, we first convert f_1 and f_2 into the same units by the cost coefficients ξ_1 and ξ_2, and then we have $f = \alpha \cdot \xi_1 \cdot f_1 + \beta \cdot \xi_2 \cdot f_2$, where $\alpha + \beta = 1$. The traditional GA is then used to solve the problem. To make a fair comparison with NSGA-II, GA adopts the same coding scheme as NSGA-II. We increase α from 0.1 to 0.9 with an increase of 0.1, and the corresponding results are showed in Figure 14.

Figure 14 shows the mapping of NSGA-II and GA solutions in the objective space. First of all, in terms of f_1 and f_2 values, there is little difference between the solutions obtained by these two algorithms. This indicates that the proposed optimization method is reasonable and effective, and the results are highly robust.

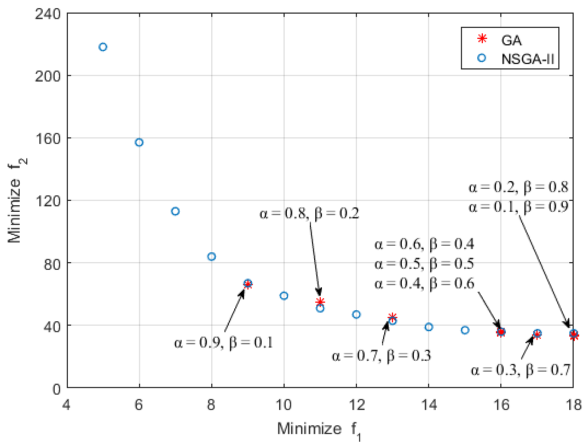


Fig. 14. Comparison between NSGA-II and GA.

Secondly, we analyze the value of α in GA method. When $\alpha \leq 0.6$, GA is less sensitive to α, and both NSGA-II and GA can produce effective timetable schemes. When $\alpha > 0.6$, GA is sensitive to α. The solution interval among different settings of GA is large, and GA cannot search for some solutions of NAGA-II. While, the Pareto solution generated by NSGA-II is not only has good convergence and diversity, but also not worse than that obtained by GA.

Thirdly, in terms of the solution range. The f_1 and f_2 ranges of NSGA-II are [5], [18] and [33, 211], while the GA's are [9], [18] and [33], [66]. In addition, the proposed optimization method can generate a set of Pareto solutions by running NSGA-II once, which directly and comprehensively contains multiple possible decision preferences. Then, decision-makers can choose the best scheme according to their own preferences. Therefore, the NSGA-II algorithm has a wider solution range.

Finally, as for the calculation speed. Since the NSGA-II needs to calculate two target dimensions of fitness value during the iteration process, while GA only needs to calculate once and has not the non-dominated sorting step, the computing speed of GA is faster than that of NSGA-II. When f_1 is large, it is better to use the GA method. However, it is difficult for GA to produce a similar Approximate Pareto Front likes NSGA-II.

In summary, to solve the proposed optimization model, the NSGA-II algorithm is more appropriate than GA method.

VI. Conclusion

This article proposes a data-driven bi-objective programming model to optimize the bus timetable problem, with the objective of minimizing the total waiting time of passengers and reducing the departure times of bus company. Different from the previous method, the model takes into account the time-dependent travel time and estimates the arrival time of each bus more accurately via the station dwell time. In addition, we analyze the GPS and Smart Card datasets to estimate parameters of the optimization model: $Y_{j,j+1}(t)$, D_j^i, $\varphi_j(k)$ and $\gamma_j(k)$. Based on the features of timetable, we design an improved NSGA-II with a specific coding scheme to quickly generate multiple a set of Pareto optimal solutions. The proposed method is applied for real datasets of a bus line in Beijing, China. Experiment shows that our approach provides the timetable solution with less passengers waiting time and number of departures times than traditional GA and empirical scheduling method.

According to the limitation of current works, future research will focus on following several directions: (1) this study only uses the GPS and Smart Card datasets to estimate parameters. Actually, it can be more accurate and reliable by combining with other data sources (such as query service from mobile app and video analysis from bus CCTV); (2) the alighting and boarding passenger volume is estimated from the historical dataset. In the future, we will focus on predicting the passenger flow to provide the real-time bus timetable; (3) this study applies a heuristic algorithm in the timetable optimization, and there are also many other multi-objective optimization algorithms that can be used to solve timetable problem. We will consider applying different algorithm to different timetable problems in future studies.

References

[1] J. Zhang *et al.*, "A real-time passenger flow estimation and prediction method for urban bus transit systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3168–3178, Nov. 2017.

[2] Y. Wang, D. Zhang, L. Hu, Y. Yang, and L. H. Lee, "A data-driven and optimal bus scheduling model with time-dependent traffic and demand," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2443–2452, Sep. 2017.

[3] A. Ceder, "Optimal multi-vehicle type transit timetabling and vehicle scheduling," *Procedia Social Behav. Sci.*, vol. 20, pp. 19–30, Jan. 2011.

[4] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz, "Planning, operation, and control of bus transport systems: A literature review," *Transp. Res. B, Methodol.*, vol. 77, pp. 38–75, Jul. 2015.

[5] M. Owais, M. K. Osman, and G. Moussa, "Multi-objective transit route network design as set covering problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 670–679, Mar. 2016.

[6] B. Yu, Z.-Z. Yang, P.-H. Jin, S.-H. Wu, and B.-Z. Yao, "Transit route network design-maximizing direct and transfer demand density," *Transp. Res. C, Emerg. Technol.*, vol. 22, pp. 58–75, Jun. 2012.

[7] J. Chen, Z. Liu, S. Wang, and X. Chen, "Continuum approximation modeling of transit network design considering local route service and short-turn strategy," *Transp. Res. E, Logistics Transp. Rev.*, vol. 119, pp. 165–188, Nov. 2018.

[8] K. Gkiotsalitis and F. Alesiani, "Robust timetable optimization for bus lines subject to resource and regulatory constraints," *Transp. Res. E, Logistics Transp. Rev.*, vol. 128, pp. 30–51, Aug. 2019.

[9] J. Tang, Y. Yang, and Y. Qi, "A hybrid algorithm for urban transit schedule optimization," *Phys. A, Stat. Mech. Appl.*, vol. 512, pp. 745–755, Dec. 2018.

[10] Y. Wu, H. Yang, J. Tang, and Y. Yu, "Multi-objective re-synchronizing of bus timetable: Model, complexity and solution," *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 149–168, Jun. 2016.

[11] C. Tang, A. Ceder, S. Zhao, and Y.-E. Ge, "Vehicle scheduling of single-line bus service using operational strategies," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1149–1159, Mar. 2019.

[12] T. Liu and A. (Avi) Ceder, "Deficit function related to public transport: 50 year retrospective, new developments, and prospects," *Transp. Res. B, Methodol.*, vol. 100, pp. 1–19, Jun. 2017.

[13] L. Desfontaines and G. Desaulniers, "Multiple depot vehicle scheduling with controlled trip shifting," *Transp. Res. B, Methodol.*, vol. 113, pp. 34–53, Jul. 2018.

[14] J. Heil, K. Hoffmann, and U. Buscher, "Railway crew scheduling: Models, methods and applications," *Eur. J. Oper. Res.*, vol. 283, no. 2, pp. 405–425, Jun. 2020.

[15] M. Fuentes, L. Cadarso, and Á. Marín, "A hybrid model for crew scheduling in rail rapid transit networks," *Transp. Res. B, Methodol.*, vol. 125, pp. 248–265, Jul. 2019.

[16] C. P. Medard and N. Sawhney, "Airline crew scheduling from planning to operations," *Eur. J. Oper. Res.*, vol. 183, no. 3, pp. 1013–1027, Dec. 2007.

[17] A. Moreno, P. Munari, and D. Alem, "A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration," *Eur. J. Oper. Res.*, vol. 275, no. 1, pp. 16–34, May 2019.

[18] F. Delgado, J. C. Munoz, and R. Giesen, "How much can holding and/or limiting boarding improve transit performance?" *Transp. Res. B, Methodol.*, vol. 46, no. 9, pp. 1202–1217, Nov. 2012.

[19] O. Cats, A. N. Larijani, Á. Ólafsdóttir, W. Burghout, I. J. Andréasson, and H. N. Koutsopoulos, "Bus-holding control strategies: Simulation-based evaluation and guidelines for implementation," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2274, no. 1, pp. 100–108, Jan. 2012.

[20] Z. Dai, X. C. Liu, Z. Chen, R. Guo, and X. Ma, "A predictive headway-based bus-holding strategy with dynamic control point selection: A cooperative game theory approach," *Transp. Res. B, Methodol.*, vol. 125, pp. 29–51, Jul. 2019.

[21] K. Sivakumaran, Y. Li, M. J. Cassidy, and S. Madanat, "Cost-saving properties of schedule coordination in a simple trunk-and-feeder transit system," *Transp. Res. A, Policy Pract.*, vol. 46, no. 1, pp. 131–139, Jan. 2012.

[22] J. A. Mesa, F. A. Ortega, and M. A. Pozo, "Locating optimal timetables and vehicle schedules in a transit line," *Ann. Oper. Res.*, vol. 222, no. 1, pp. 439–455, Nov. 2014.

[23] H. Martínez, A. Mauttone, and M. E. Urquhart, "Frequency optimization in public transportation systems: Formulation and Metaheuristic approach," *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 27–36, Jul. 2014.

[24] N. Seshagiri, R. Narasimhan, S. L. Mehndiratta, and B. K. Chanda, "Computer generated time-tables and bus schedules for a large bus transport network," *Transp. Sci.*, vol. 3, no. 1, pp. 69–85, Feb. 1969.

[25] B. Varga, T. Tettamanti, and B. Kulcsár, "Optimally combined headway and timetable reliable public transport system," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 1–26, Jul. 2018.

[26] K. Nitisiri, M. Gen, and H. Ohwada, "A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling," *Comput. Ind. Eng.*, vol. 130, pp. 381–394, Apr. 2019.

[27] X. Guo, H. Sun, J. Wu, J. Jin, J. Zhou, and Z. Gao, "Multiperiod-based timetable optimization for metro transit networks," *Transp. Res. B, Methodol.*, vol. 96, pp. 46–67, Feb. 2017.

[28] X. Yang, A. Chen, B. Ning, and T. Tang, "A stochastic model for the integrated optimization on metro timetable and speed profile with uncertain train mass," *Transp. Res. B, Methodol.*, vol. 91, pp. 424–445, Sep. 2016.

[29] I. Louwerse and D. Huisman, "Adjusting a railway timetable in case of partial or complete blockades," *Eur. J. Oper. Res.*, vol. 235, no. 3, pp. 583–593, Jun. 2014.

[30] H. Niu and X. Zhou, "Optimizing urban rail timetable under time-dependent demand and oversaturated conditions," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 212–230, Nov. 2013.

[31] Z. Wang *et al.*, "Two-step coordinated optimization model of mixed demand responsive feeder transit," *J. Transp. Eng. A, Syst.*, vol. 146, no. 3, Mar. 2020, Art. no. 04019082.

[32] D. Sun, Y. Xu, and Z.-R. Peng, "Timetable optimization for single bus line based on hybrid vehicle size model," *J. Traffic Transp. Eng.*, vol. 2, no. 3, pp. 179–186, Jun. 2015.

[33] P. Sun, L. P. Veelenturf, M. Hewitt, and T. Van Woensel, "The time-dependent pickup and delivery problem with time windows," *Transp. Res. B, Methodol.*, vol. 116, pp. 1–24, Oct. 2018.

[34] J. Lu, Z. Yang, H. Timmermans, and W. Wang, "Optimization of airport bus timetable in cultivation period considering passenger dynamic airport choice under conditions of uncertainty," *Transp. Res. C, Emerg. Technol.*, vol. 67, pp. 15–30, Jun. 2016.

[35] X. Zuo, C. Chen, W. Tan, and M. Zhou, "Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 1030–1041, Apr. 2015.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[37] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

Jinjun Tang received the Ph.D. degree in transportation engineering from the Harbin Institute of Technology, Harbin, China, in 2016. From 2014 to 2016, he was a Visiting Scholar with the Smart Transportation Applications and Research Laboratory (STAR Lab), University of Washington, Seattle, WA, USA. He is currently an Associate Professor with the School of Traffic and Transportation Engineering, Central South University, Changsha, China. He has published more than 70 technical articles in the journal and conference proceeding as the author and a coauthor. His research interests include traffic flow prediction, data mining in transportation systems, intelligent transportation systems, and transportation modeling.

Yifan Yang received the B.S. degree in logistics engineering from the Shaanxi University of Science and Technology, China, in 2017, and the M.S. degree from the School of Traffic and Transportation Engineering, Central South University, China, in 2020. Her research interests include mathematical modeling, heuristics algorithms, and operations research optimization with applications.

Wei Hao (Member, IEEE) received the Ph.D. degree from the Department of Civil and Environmental Engineering, New Jersey Institute of Technology. He is currently a Professor with the Changsha University of Science and Technology, China. His research interests include traffic operations, ITS, planning for operations, traffic modeling and simulation, connected automated vehicles, and travel demand forecasting.

Fang Liu received the B.S. and M.S. degrees in traffic engineering from the Shandong University of Technology, Zibo, China, in 2006 and 2009, respectively. She is currently pursuing the Ph.D. degree with the School of Transportation Engineering, Changsha University of Science and Technology, China. Her research interests include traffic safety, and transportation modeling and simulation.

Yinhai Wang received the Ph.D. degree from The University of Tokyo, Tokyo, Japan, in 1998. He is currently a Professor with the Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA. He has published 85 peer-reviewed journal articles and made 101 invited talks and over 170 other technical presentations. His research interests include traffic detection systems, traffic accident modeling, transportation data management and analysis, traffic control, and traffic simulations. He is a member of the IEEE Smart Cities Steering Committee. He was on the IEEE ITS Society Board of Governors from 2010 to 2013.