Hindawi
Journal of Advanced Transportation
Volume 2019, Article ID 2781590, 13 pages
https://doi.org/10.1155/2019/2781590

Research Article

Optimizing Vehicle Scheduling Based on Variable Timetable by Benders-and-Price Approach

Zekang Lan, Shiwei He, Rui Song, and Sijia Hao

Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Ministry of Transport, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Shiwei He; shwhe@bjtu.edu.cn

Received 17 October 2018; Revised 18 December 2018; Accepted 28 December 2018; Published 6 January 2019

Academic Editor: Guohui Zhang

In practice, vehicle scheduling is planned on a variable timetable so that the departure times of trips can be shifted in tolerable ranges, rather than on a fixed timetable, to decrease the required fleet size. This paper investigates the vehicle scheduling problem on a variable timetable with the constraint that each vehicle can perform limited trips. Since the connection-based model is difficult to solve by optimization software for a medium-scale or large-scale instance, a designed path-based model is developed. A Benders-and-Price algorithm by combining the Benders decomposition and column generation is proposed to solve the LP-relaxation of the path-based model, and a bespoke Branch-and-Price is used to obtain the integer solution. Numerical experiments indicate that a variable timetable approach can reduce the required fleet size with a tolerable timetable deviation in comparison with a fixed timetable approach. Moreover, the proposed algorithm is greatly superior to GUROBI in terms of computational efficiency and guarantees the quality of the solution.

1. Introduction

In the planning process of public bus transport, timetabling problem, and vehicle scheduling problem (VSP) are dealt with by a sequential approach where the solution of previous subproblem is taken as input of the following subproblem, because of the complexity of solving the integrated model of the two planning processes [1]. Both the timetabling problem and VSP have been well-studied. The former one determines the departure and arrival times at each station for each trip, aiming to offer a high level of service for passengers [2–4]. Based on the timetabled trips, the latter one focuses on minimizing the operational cost by arranging the vehicles to cover each trip exactly once and turning back to their corresponding origin terminals or depots after finishing a defined work period, e.g., one day. The connections of trips depend on the arrival/departure times at the terminals. Hence, the process of vehicle scheduling is highly dependent on the timetable. Thus, in order to lower the number of costly vehicles used in vehicle scheduling, a variable timetable with trips shifting in acceptable tolerances is adopted instead of a fixed timetable. However, the process of shifting is a very time-consuming mission and usually not done in a systematic manner [2]. Ceder [2] and Liu and Ceder [5] have presented deficit-function procedures with and without insertions of deadheading trips to assist vehicle schedulers in achieving the most efficient shifts. However, when taking into account fuel restrictions, maintenance considerations, etc., each vehicle has a limited daily workload that implies that the number of trips taken by one vehicle in one day cannot exceed an upper bound (limited trips, LT). In this case, the deficit-function method is not applicable and a new solution method should be proposed. To the best of our knowledge, the VSP based on a variable timetable with limited trips (VSPVT-LT) has not been studied.

Previous studies typically researched into vehicle scheduling based on a fixed timetable. The usual modeling framework of the VSP is built on a connection-based network, where trips and depots are expressed as vertices, and compatible connections between any two vertices are connected using arcs [6–8]. Considering that the connection-based method could bring an immense number of deadhead connection arcs when the number of trips grows, a space-time network modeling concept has been commonly used [8, 9]. Bertossi et al. [10] have proved that single depot vehicle scheduling problem (SDVSP) can be solved in polynomial time; however multiple depot vehicle scheduling problem (MDVSP) is NP-hard even for two depots. Generally, the connection-based model is hard to solve by optimization software when thousands of trips have to be scheduled. Costa et al. [11] and Hadjar et al. [12] formulated the VSP as a Set Portioning Problem which can be solved successfully by column generation. Bodin et al. [13] and Freling and Paixão [14] discussed the vehicle scheduling problem with time constraint, and they constructed a special connection-based network with inserting additional arcs or alternative graphs with different levels to handle the time constraint. Desaulniers et al. [15] and Hadjar and Soumis [16] researched into the vehicle scheduling problem with time window (VSPTW). However, considering that the time windows of departure times may overlap, the minimum headway requirement, which is usually not considered in the VSPTW, has to be imposed in the VSPVT-LT.

In recent years, integrated optimization models for two consecutive phases of public transit have attracted more and more attention. Ibarra-Rojas et al. [17] proposed a completed biobjective integration model that combines the SDVSP and the timetabling problem, and a ε-constraint method was implemented to obtain Pareto-optimal solutions. On the other hand, most of the literature addresses a sequential integration of the timetabling problem with the VSP. A sequential integration is an approach that considers the characteristics of one subproblem while another subproblem is being optimized. Van den Heuvel et al. [18] applied an iterated heuristic method between a local search strategy to alter the timetable slightly and a network-flow model for vehicle scheduling. In each iteration, the modification of the timetable with an improvement is always adopted, while with a degradation it is accepted with a probability which decreases in the wake of the algorithm processing. Guihaire and Hao [19] also presented an iterated local search algorithm where each iteration implements trip shifting to obtain a neighboring timetable and then the VSP is optimized by an efficient auction algorithm on the neighboring timetable. Liu et al. [20] provided a new biobjective, bilevel mathematical programming model, and a novel deficit-function-based sequential search approach by combining a network-flow technique and a shifting departure time procedure, presented to solve the problem to achieve a set of Pareto-efficient solutions. Fonseca et al. [21] proposed a metaheuristic that iteratively optimized the mathematical formulation of the integrated timetabling and VSP that permitted a subset of timetabled trips shifting only, whereas solving the full VSP.

The remainder of this paper is organized as follows. In Section 2, the overall problem description and notations are elaborated, and a directed network is built to make provision for the connection-based model in the Appendix and the shortest path algorithm in Section 4.2. The path-based model is put forward in Section 3. A computationally efficient solution method that consists of two phases to solve the path-based model is presented in Section 4. The efficiency of the proposed algorithm and the benefits of the vehicle scheduling in a variable timetable are systematically evaluated with numerical experiments in Section 5. Finally, conclusion and future work extensions are discussed in Section 6.

2. The Problem Description and Notations

This study considers a bus line or a set of bus lines. At the ends of each line are terminals. An initial feasible daily timetable generated by timetabling specifies the origin terminal, destination terminal, departure time at origin terminal, and arrival time at destination terminal of each trip. A timetable is feasible if several constraints such as the minimum headway requirement and the minimum travel time of trip are satisfied, where the headway time is the time interval between the departure times of two adjacent trips with the same origin terminal. Considering that the departure time of each trip can be shifted backward or forward in a given range and respecting the minimum headway requirement of operation, there are a lot of possible feasible timetables. In our case, fuel restrictions and maintenance plans are taken into account; that is, the running mileage or operational time of each vehicle in one day cannot exceed a defined upper bound for the vehicle. Assuming that there are a few differences between the running mileages or travel times of any two trips, the restriction is equivalent to limiting the number of trips carried out by each vehicle. The VSPVT-LT has to find out a feasible timetable with the least fleet size in vehicle scheduling with LT constraints, but also needs to balance the timetable deviation in relation to the initial timetable. We do not consider the inclusion of deadheading trips insertions, and we assume that the travel time of each trip is fixed in timetable modification. For illustrative purpose, some notations for the entire article are firstly introduced in Table 1.

The maximum turnaround time $\chi_{\max}$ is used to reduce nonproductive time and the number of connection arcs in the connection-based model to improve computation efficiency. Two trips $i, j \in T$ are compatible in a given timetable $t \in \Omega$ if they can be performed by the same vehicle in sequence. The route is a sequence of trips, and any two consecutive trips in a route are compatible. For any two trips $i, j \in T$ in timetable t_0 such that (i) $d(i) = o(j)$, (ii) $(D_j + \Delta d_j^+) - (A_i - \Delta d_i^-) \geq \chi_{\min}$, and (iii) $(D_j - \Delta d_j^-) - (A_i + \Delta d_i^+) \leq \chi_{\max}$, we call them restrictively compatible, since i and j will be compatible by adjusting their departure times in a given range even if i and j are incompatible in t_0. If i and j are restrictively compatible, this is denoted as $i \longleftrightarrow j$.

To facilitate explanation for VSPVT-LT, consider a bus line with two terminals, which has 14 trips in its initial timetable as depicted in Figure 1. Assuming that the minimum turnaround time is 10 min, the minimum headway time is 3 min, and the maximum turnaround time and vehicle daily workload are not limited, we seek to obtain the vehicle schedule solution based on the fixed timetable with the least required fleet size, which has four routes and they are revealed by the different color trajectories. For instance, the route as the red trajectory shows starts from depot 1, travels trip 1, trip 10, trip 5, and trip 14, sequentially, and end at depot 1. Additionally, because a feasible route should always have the same starting and ending depots, trip 2 and trip 9 cannot put into service, so they have to be canceled in the final timetable.

Table 1: Some notations for the entire article.

Sets	
T	Set of trips, $T = \{i : i = 1, \cdots, n\}$,where n is the total number of trips
Ω	Set of all the feasible timetables
K	Set of terminals
R	Set of vehicles, $R = \{r : r = 1, 2, \cdots, u\}$, where u is the total number of vehicles
R_k	Set of vehicles located in terminal $k \in K$, $R_k \subset R$
$T^d(k)\,(T^a(k))$	Sets of trips that start from (end at) the terminal $k \in K$
Parameters	
t_0	The given initial daily timetable
$D_i\,(A_i)$	The departure (arrival) time of trip $i \in T$ in t_0
l_i	The travel time of trip $i \in T$
λ_i	The penalty cost for shifting the departure time of trip $i \in T$ per unit time, e.g., one minute
w_i	The penalty cost for canceling trip $i \in T$
c_r	The cost for vehicle $r \in R$ if it is used
c^R	$c^R = \min\{c_r \mid r \in R\}$
$o(i)\,(d(i))$	The origin (destination) terminal of trip $i \in T$
h	The minimum headway of operational requirement
$\Delta d_i^-\,(\Delta d_i^+)$	The maximum allowed earliness (tardiness) for the departure time of trip $i \in T$
$\chi_{\min}\,(\chi_{\max})$	The minimum (maximum) time for a turnaround operation
N_r	The maximum allowed number of trips executed by vehicle $r \in R$
Decision variables	
x_{ij}^k	Binary variables, 1 if vehicle $r \in R$ travels from vertex (trip) i directly to vertex (trip) j, and 0 otherwise
$d_i\,(a_i)$	Non-negative integer variables, indicating the departure (arrival) time of trip $i \in T$ in the final output feasible timetable
m_i	Binary variables, 1 if trip $i \in T$ is canceled, and 0 otherwise

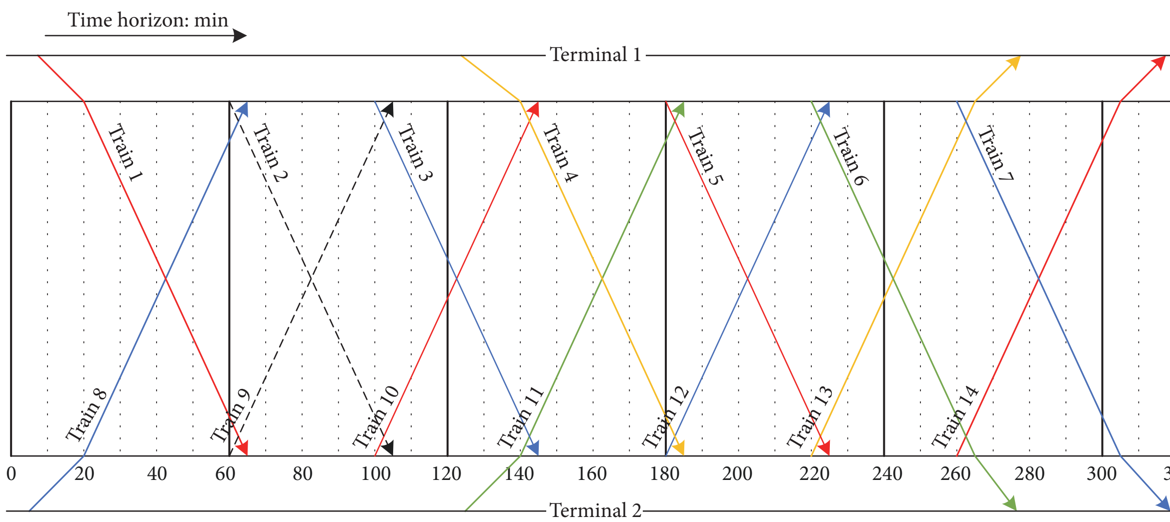

Figure 1: The initial timetable and the corresponding vehicle schedule solution.

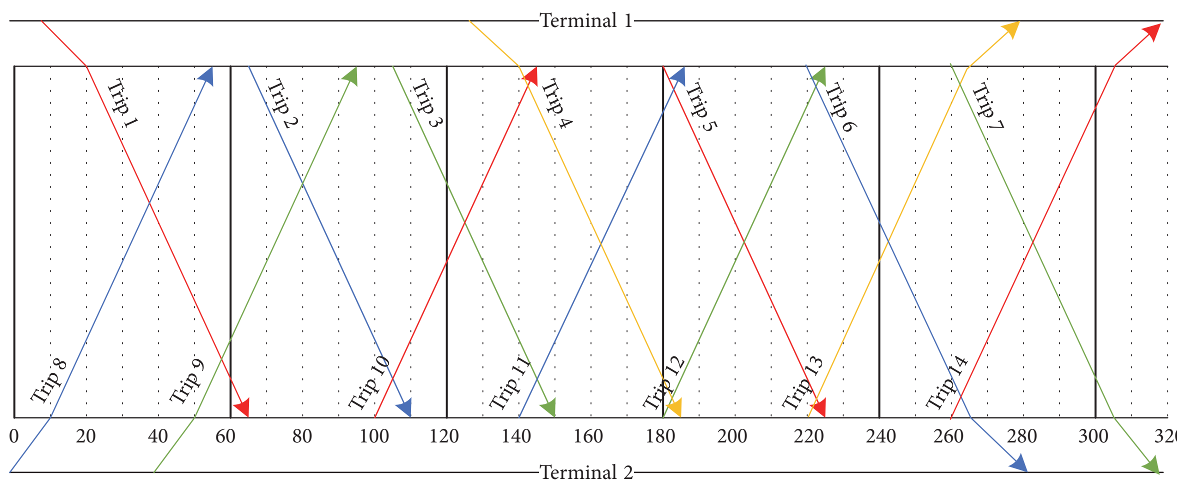

Figure 2: The modified timetable and the corresponding vehicle schedule solution.

The result will certainly be unsatisfactory to the planning staffs.

Considering the case on a variable timetable, e.g., we set $\Delta d_i^- = \Delta d_i^+ = 10$ min for each trip $i \in T$, a modified timetable shown in Figure 2 is obtained from the initial timetable by shifting the departure times of trip 8 and trip 9 backward 10 min, trip 2 and trip 3 forward 5 min, respectively. Similarly, the corresponding vehicle schedule solution is generated and reported in Figure 2. Despite the same number of vehicles used as before adjustment, all trips are performed by exactly one vehicle. In the example, at first sight, it is a relatively simple structure. However, one should not draw a conclusion that these connections can be accomplished in manual planning, because the example exhibits an isolated part of a complicate structure, especially when LT constraints are taken into account.

We associate each terminal $k \in K$ with a directed network $G_k = (V_k, A_k)$, where V_k and A_k denote the sets of vertices and arcs in this network, respectively. Each trip $i \in T$ is represented by a trip-vertex, and each terminal $k \in K$ by a source-vertex $n + 2k - 1$ and sink-vertex $n + 2k$ (every terminal is duplicated into a source-vertex and a sink-vertex). The vertex set V_k contains all the trip-vertices, corresponding source-vertex $n+2k-1$, and sink-vertex $n+2k$. For simplicity, the set of trip-vertices is also represented by T. Thus $V_k = T \cup \{n + 2k - 1, n + 2k\}$. The set A_k contains the start arcs $\{n + 2k - 1\} \times T^d(k)$, the end arcs $T^a(k) \times \{n + 2k\}$, and the inner arcs $I = \{(i, j) \mid i, j \in T : i \longleftrightarrow j\}$. Thus, A_k is defined as $A_k = (\{n + 2k - 1\} \times T^d(k)) \cup (T^a(k) \times \{n + 2k\}) \cup I$. For any vertex $i \in V_k$ in each network G_k $(k \in K)$, $I_k(i)$ and $O_k(i)$ specify the set of ingoing and outgoing vertices of vertex i, respectively.

3. The Path-Based Model for VSPVT-LT

In order to evaluate the benefits of the path-based model, we also develop a connection-based model, which is detailed in the Appendix. We will verify that the connection-based model is hard to solve by GUROBI in the medium-scale or large-scale instances in Section 5.2. To overcome the handicap, the column generation technology should be employed. Embedded in Branch-and-bound (B&B) algorithm, column generation has been applied to successfully solve real-world instances of routing and scheduling problems [16].

Let us firstly define the term "path" that is distinct from the "route" defined before. The only difference is that any two consecutive trips in a path are restrictively compatible in the initial timetable t_0, without requiring being compatible. However, a path will be a route in a certain timetable $t \in \Omega$. Considering the example mentioned in Section 2 again, the sequence of trips (8,2,11,6) is a path in the initial timetable rather than a route, because trip 8 and trip 2 are incompatible but restrictively compatible. However, (8,2,11,6) is also a route in the modified timetable. The notations for the path-based model are presented in Table 2.

The path-based model is now given as follows.

$$\begin{aligned} \min \quad & \sum_{i\in T}\lambda_i \left|d_i - D_i\right| + \sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} c_s x_s(p) \\ & + \sum_{i\in T} w_i \left(1 - \sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \delta_p^i x_s(p)\right) \\ = & \sum_{i\in T}\lambda_i \left|d_i - D_i\right| + \sum_{i\in T} w_i \\ & + \sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \left(c_s - \sum_{i\in T} w_i \delta_p^i\right) x_s(p) \end{aligned} \tag{1}$$

$$\text{s.t.} \quad \text{constraints (A.2-A.4) (See the Appendix)} \tag{2}$$

$$d_j - a_i \geq \chi_{\min} - \left(1 - y_{ij}\right) M \quad \forall (i,j) \in I \tag{3}$$

$$d_j - a_i \leq \chi_{\max} + \left(1 - y_{ij}\right) M \quad \forall (i,j) \in I \tag{4}$$

$$\sum_{p\in P(k,s)} x_s(p) \leq \varsigma_{k,s} \quad \forall k \in K,\ s \in S_k \tag{5}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \delta_p^i x_s(p) \leq 1 \quad \forall i \in T \tag{6}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \theta_p^{ij} x_s(p) \leq y_{ij} \quad \forall (i,j) \in I \tag{7}$$

$$x_s(p) \in \{0,1\} \quad \forall k \in K,\ s \in S_k,\ p \in P(k,s) \tag{8}$$

The objective function (1) is to minimize the total cost as (A.1) (see the Appendix) expresses. Constraints (3) and (4) ensure that if trips i and j are compatible, then the difference between the departure time of j and the arrival time of i has to be greater than or equal to $\chi_{\min}$, and smaller than or equal to $\chi_{\max}$. Constraints (5) make sure that there are adequate vehicles to execute the paths being selected. Constraints (6) require that for each trip $i \in T$, at most one path is selected among those paths containing trip i. Constraints (7) state that if trips i and j are incompatible, the paths containing i and j in sequence are forbidden to be selected. Constraints (8) define the domains of the variables.

4. Solution Method

4.1. Benders-and-Price Solution Framework. In the path-based model, we observe that the variables associated with the timetabling problem, i.e., a_i and d_i, and the variables associated with the VSP, i.e., $x_s(p)$, are linked together by y_{ij}. For a given timetable and corresponding values of y_{ij} satisfying constraints (3) and (4), the LP-relaxation of path-based model only involves the variables $x_s(p)$ and can be solved easily by column generation. Therefore, it is reasonable to use Benders decomposition technology to divide the original problem into two subproblems as follows. The Benders master problem (BMP) is the timetabling problem that determines the timetable and corresponding variables of y_{ij}. The Benders subproblem (BSP) is a path-based MDVSP model which can be solved by column generation. The approach that combines Benders decomposition and column generation has been applied to other simultaneous problems in transportation, such as locomotives and cars assignment to passenger trains problem [22], and aircraft routing and crew scheduling problem [23]. For more recent literature about this approach, we refer the reader to Restrepo et al. [24].

Introducing an additional variable z_0 ($z_0 \geq 0$), the Benders master problem can be formulated as

$$\text{(BMP)} \min \quad \sum_{i\in T}\lambda_i \left|d_i - D_i\right| + z_0 \tag{9}$$

$$\text{s.t.} \quad \text{constraints (A.2) - (A.4), (3) - (4), and} \tag{10}$$

$$z_0 \geq z_0^{down} \tag{11}$$

where z_0^{down} is the lower bound of z_0. Because relative large penalty costs are set for canceling trips as mentioned in the Appendix, it is more cost-efficient to use a vehicle than to cancel a trip. Therefore, the lower bound of the objective function (1) can be the least cost of using vehicles, which can be expressed as $(n/\vartheta) * c^R$. Moreover, remember that the objective value of BMP is the lower bound of the objective function (1). By comparing the objective function (1) and (9), it is reasonable to set z_0^{down} as $(n/\vartheta) * c^R$. After solving BMP, the departure times of trips and the variables of $y_{ij}((i,j) \in I)$ are fixed to particular values $\overline{d}_i$ $(i \in T)$ and $\overline{y}_{ij}((i,j) \in I)$. The path-based model reduces to the following primal Benders subproblem (BSP).

$$\begin{aligned} \text{(BSP)} & \sum_{i\in T}\lambda_i \left|\overline{d}_i - D_i\right| + \sum_{i\in T} w_i \\ & + \min \sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \left(c_s - \sum_{i\in T} w_i \delta_p^i\right) x_s(p) \end{aligned} \tag{12}$$

$$\sum_{p\in P(k,s)} x_s(p) \leq \varsigma_{k,s} \quad \forall k \in K,\ s \in S_k \tag{13}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \delta_p^i x_s(p) \leq 1 \quad \forall i \in T \tag{14}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{p\in P(k,s)} \theta_p^{ij} x_s(p) \leq \overline{y}_{ij} \quad \forall (i,j) \in I \tag{15}$$

TABLE 2: Notations for the path-based model.

Sets	
S	Set of vehicle types. Different types vehicles can execute different maximum allowed number of trips
S_k	Set of vehicle types in terminal k, $S_k \subset S$
ϑ, ϑ_k	$\vartheta = \max\{s \mid s \in S\}$, $\vartheta_k = \max\{s \mid s \in S_k\}$
P	Set of all possible paths
$P(k, s)$	Set of paths that can be covered by a vehicle from terminal $k \in K$ of type $s \in S_k$
Parameters	
c_s	The cost of path $p \in P$ covered by a vehicle of type $s \in S$. We suppose that two different vehicles with the same type have the same cost
$\varsigma_{k,s}$	The number of vehicles of type $s \in S_k$ in terminal $k \in K$, $\sum_{k\in K} \sum_{s\in S_k} \varsigma_{k,s} = u$
δ_p^i	A binary constant taking the value 1 if the path p includes the trip i, 0, otherwise
θ_p^{ij}	A binary constant taking the value 1 if the path p includes trips i and j in sequence, 0, otherwise
Decision variables	
$x_s(p)$	A binary variable which takes value 1 if the path p is selected and covered by a vehicle of type $s \in S$ in an optimal solution, 0, otherwise
y_{ij}	A binary variable which takes value 1 if trips i and j are compatible, 0, otherwise

$$x_s(p) \in \{0,1\} \quad \forall k \in K,\ s \in S_k,\ p \in P(k,s) \tag{16}$$

Obviously, it is impractical to enumerate the huge number of paths in the set of P and corresponding $P(k,s)$ $(k \in K, s \in S_k)$. Therefore, a column generation method should be applied to generate the more profitable paths. Considering a LP-relaxation of restricted BSP (denoted by RBSP) obtained from the BSP by replacing $P(k,s)$ with its subset $\overline{P}(k,s)$ and replacing (16) with $x_s(p) \in [0,1]$. Let $\boldsymbol{\pi} = (\pi_{k,s} \le 0)$, $\boldsymbol{\mu} = (\mu_i \le 0)$, and $\boldsymbol{\sigma} = (\sigma_{ij} \le 0)$ be the dual variables associated with constraints (13), (14), and (15), respectively. For each $k \in K$ and $s \in S_k$, the reduced cost of path p for the dual solution $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ is given by $c_s - \sum_{i\in T} w_i \delta_p^i - \sum_{k\in K}\sum_{s\in S_k} \pi_{k,s} - \sum_{i\in T}\delta_p^i \mu_i - \sum_{(i,j)\in I}\theta_p^{ij}\sigma_{ij}$. Thus, identifying a column with minimum reduced cost of the pricing step for vehicle type $s \in S_k$ $(k \in K)$ is equivalent to the pricing subproblem.

$$\text{(PSP)}\ \zeta_{k,s} = \min_{p\in\overline{P}(k,s)} \left\{ c_s - \sum_{i\in T} w_i \delta_p^i - \sum_{k\in K}\sum_{s\in S_k}\pi_{k,s} - \sum_{i\in T}\delta_p^i\mu_i - \sum_{(i,j)\in I}\theta_p^{ij}\sigma_{ij} \right\} \tag{17}$$

If the objective value of PSP $\zeta_{k,s} < 0$, we can enter the corresponding path into the path set $\overline{P}(k,s)$ to improve the solution. If for each $k \in K$ and $s \in S_k$, $\zeta_{k,s} \ge 0$, the optimal solution of RBSP at the current iteration has been obtained. In order to solve the pricing subproblems faster, we present a label correcting shortest path algorithm, which will be talked about in the next section.

The RBSP is always feasible since the null vector **0** satisfies constraints (13)-(16), then artificial variables and Benders feasibility cuts are not essential to be generated in RBSP. The objective values of RBSP and BMP are the upper bounds and lower bounds at Benders-and-Price approach iterations, denoted as η_{UB} and η_{LB}, respectively. Note that the last convergence value of the Benders decomposition is actually the lower bound of the optimal objective value of the original path-based model, because we have applied LP-relaxation to RBSP. If $|\eta_{UB} - \eta_{LB}|/\eta_{UB} \le \varepsilon$, where ε is a predefined tolerance, Benders-and-Price can be terminated. Otherwise, a Benders cut as presented below has to be imposed into BMP.

$$z_0 \ge \sum_{i\in T} w_i + \sum_{k\in K}\sum_{s\in S_k} \varsigma_{k,s}\pi_{k,s} + \sum_{i\in T}\mu_i + \sum_{(i,j)\in I}\sigma_{ij} y_{ij} \tag{18}$$

4.2. Shortest Path Algorithm for Pricing Problem. In order to solve the problem PSP faster, a shortest path algorithm with vertices constrained, i.e., the shortest path consisting of a limited number of vertices, is proposed. In Section 2, we have associated each terminal $k \in K$ with a directed network $G_k = (V_k, A_k)$. Each arc $(i,j) \in A_k$ has a cost $cost(i,j)$, which can be determined by function (19) below. More specifically, for each start arc (i,j), the cost is mapped as $-w_j - \mu_j$. For each inter-arc $(i,j) \in I$, the cost includes not only $-w_j - \mu_j$, but also the connection cost $-\sigma_{ij}$ between vertices i and j. For each end arc, the cost is 0. The reason that the negative sign "-" remains before the dual values $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ is to comply with (17) and to keep the problem as a minimization problem.

$$cost(i,j) = \begin{cases} -w_j - \mu_j & (i,j) \in \{n+2k-1\} \times T^d(k) \\ 0 & (i,j) \in T^a(k) \times \{n+2k\} \\ -\sigma_{ij} - w_j - \mu_j & (i,j) \in I \end{cases} \tag{19}$$

The shortest path algorithm is to find a least cost path from the source-vertex $n+2k-1$ to the sink-vertex $n+2k$ for each $k \in K$ in the corresponding graph G_k. Since the cost of arcs may be negative numbers, the Bellman-Ford algorithm is adopted to get the shortest path. Remember that there is an upper bound of intermediate vertices requirement on the shortest path, and although the standard version of Bellman-Ford algorithm does not comprise a constraint on the number of vertices used, it is not difficult to modify the algorithm by making full use of the methodology in which each iteration of the Bellman-Ford algorithm finds the shortest path containing one vertex more than previous iteration to add such a restriction. Obviously, with different upper bounds of intermediate vertices, the output shortest paths may be different. Before detailing the shortest path, some notations are firstly given. For each $k \in K$, let $w(m,i)$ be the least cost (i.e., label) from source-vertex $n+2k-1$ to vertex i with at most m $(m \le \vartheta_k + 2)$ vertices and $pv(m,i)$ be the preceding vertex of vertex i on the shortest path that uses at most m $(m \le \vartheta_k + 2)$ vertices. Pseudocode for the shortest path algorithm is given in Algorithm 1 .

In Step 3 of Algorithm 1, a shortest path is fetched for each $s \in S$; thus the number of output paths may be more than one. Note that the same paths may be generated with different values of $s \in S$. If this happens, what we need to do is just to eliminate the duplicates path(s) in the output result.

4.3. Overall Algorithm. The overall algorithm comprises two phases. In the first stage, the Benders-and-Price approach alternates between RBSP and BMP with Benders cuts (18) generation until the gap between η_{LB} and η_{UB} is small enough. The BMP can be solved by IP solvers, like GUROBI, and RBSP is solved by column generation method (not embedded in B&B). If the final solution is integer valued in the end of the first phase algorithm, then the second phase can be eliminated. If not, the algorithm then enters the second phase, where it seeks integer solution of RBSP by performing a Branch-and-Price (BAP) algorithm. Note that, after the first phase, the timetable has been fixed. Figure 3 gives a flowchart for the overall algorithm.

As we can see in the BAP algorithm, the branching rule which branches on the variables $x_s(p)$ with fractional values near 1 and Depth-First node picking strategy with focus on "$x = 1$" cuts are adopted. Actually, an exact branching scheme is to branch decisions on two compatible trips to be covered by the same vehicle [15, 16]. However, for large instances, this scheme is likely to require excessive computing time because of the large number of possible connections

Step 1. Initialization.
Set $w(m,j)=\infty, \forall j\in V_k \setminus \{n+2k-1\}, m=1,2,\cdots,\vartheta_k+2; w(m,n+2k-1)=0, \forall m=1,2,\cdots,\vartheta_k+2;$
$pv(m,j)=none, \quad \forall j\in V_k, \quad m=1,2,\cdots,\vartheta_k+2.$
Step 2. Label updating
For $m=2$ to ϑ_k+2
    For each arc $(i,j)\in A_k$
        If $w(m,j) > w(m-1,i)+cost(i,j)$ Then
            $w(m,j)=w(m-1,j)+cost(i,j)$
            $pv(m,j)=i$
        End if
    End for
End for
Step 3. Fetch the shortest paths
For each $s\in S$, do
    Set the sink-vertex $n+2k$ as the current vertex i;
    For $m=s+2$ to 1
        If vertex i is not the source node $n+2k-1$
            Find the preceding vertex $pv(m,i)$ of the current vertex i, and update $pv(m,i)$ as the current vertex i;
        End if
    End for
    Reverse the backward path and output the least cost path from $n+2k-1$ to $n+2k$ with at most s intermedium vertices.
End for

Algorithm 1: The shortest path algorithm with vertices constrained in G_k.

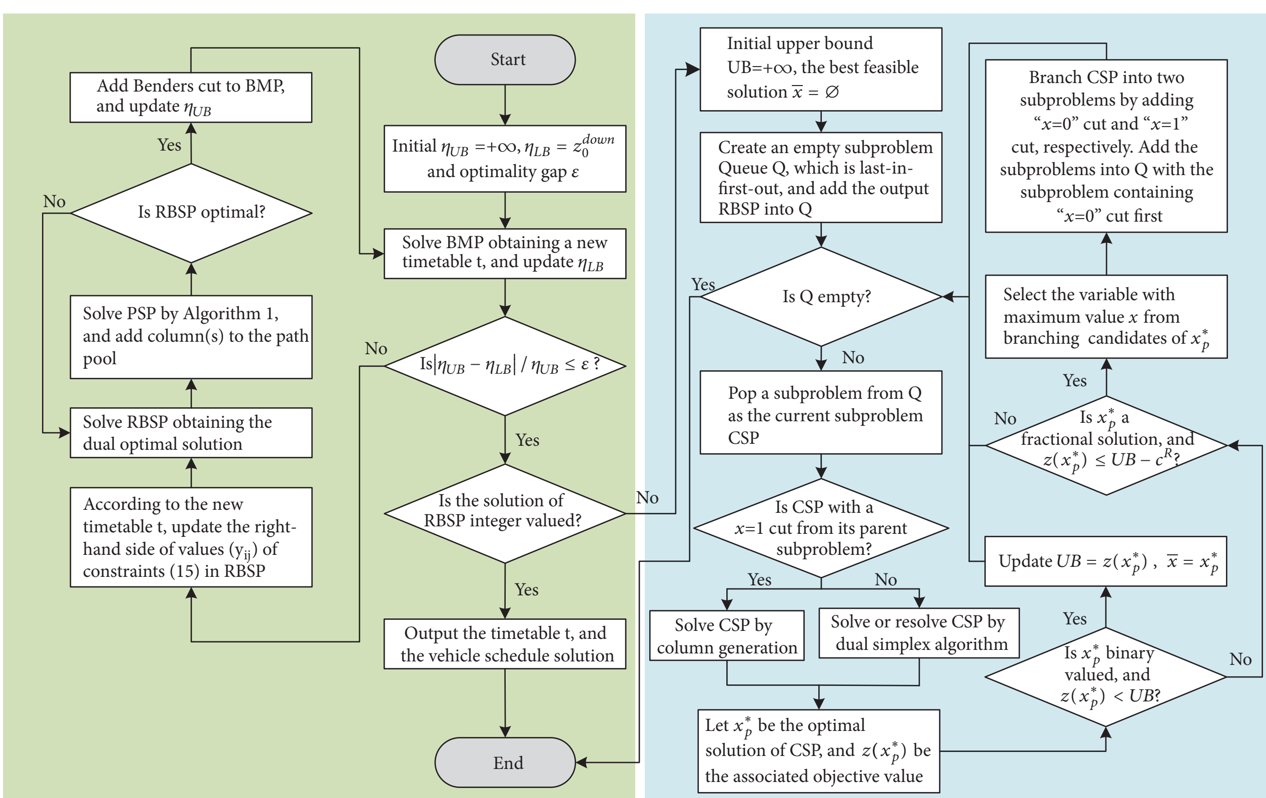

Figure 3: Flowchart of the overall algorithm.

Table 3: Results and comparisons for different approaches.

Instance	GUROBI			The proposed approach	
	Time	Obj.	Gap	Time	Obj.
(80,60,30)	4 s	800	0	3	800
(60,40,42)	21 s	800	0	6	800
(40,20,68)	487 s	1000	0	9	1000
(20,10,132)	3 h	1610	17.1%	193	1610
(10,8,228)	3 h	3056	25.4%	3481	2470

[23]. On the other hand, in the way of branching on the path variables, once one path (also a route) is fixed to be selected, all the paths which are conflicted with the fixed path will be eliminated. Moreover, recall that canceling trips is allowed, so infeasibility never occurs during the search. Then, it is hopeful to obtain a satisfactory upper bound. For a CSP imposed by an "$x = 0$" cut from its parent subproblem, we refrain from solving it by column generation, but dual simplex method, because the type of "$x = 0$" cut changes the solution space slightly, and using column generation to solve will result in the same paths being generated again, which hinders the convergence of algorithm. Although the final solution can be a suboptimal solution without solving each subproblem by column generation, the final solution is always high-quality and even optimal in our test cases. Last but not least, if x_p^* is a fractional solution, it is not always necessary to branch on the fractional variable, only when it is likely to save at least one vehicle, i.e., $z(x_p^*) \le UB - c^R$.

5. Numerical Experiments

5.1. Input Data and Parameter Settings. For simplicity, the proposed models and algorithms are verified on one of the bus lines in Beijing city of China. There are two terminals; thus $K = \{1, 2\}$. The total number of vehicles $u = 30$, and 20 of them can execute no more than 10 trips, 10 of which can execute no more than 8 trips. All the vehicles are equally assigned to two terminals; i.e., each terminal has 10 vehicles which can perform no more than 10 trips. For each trip $i \in T$, let $\Delta d_i^- = \Delta d_i^+ = \phi = 5$ min, the travel time $l_i = 45$ min. Additionally, we set $h = 3$ min, $\chi_{\min} = 10$ min, and $\chi_{\max} = 60$ min. The optimality gap ε is 0.1%. The cost factors λ_i, c_r, and w_k are set as 1, 100, and 200, respectively. All experiments are performed on a laptop equipped with 2.20 GHz Intel(R) Pentium(R) CPU and 8 GB memory, and all the algorithms are coded by C#. GUROBI 7.0.2 Optimizer is adopted as both the MIP and LP solvers.

5.2. Comparisons between GUROBI and the Proposed Approach. In this case study, we will compare the solving efficiency between using GUROBI to solve the connection-based model and using the proposed approach to solve the path-based model. Table 3 shows the experiment results on five different scale instances. Each ordered triple from the first column represents the headway time in off-peak periods, the headway time in peak periods, and total number of trips, respectively. For example, the instance (10,8,228) denotes an initial timetable t_0, for which the headway times are 10 min in off-peak periods and 8 min in peak periods, and it has 228 trips in total. As we can see, the connection-based model is difficult to solve by GUROBI. For example, in the last instance, the best feasible solution obtained by GUROBI in 3 h has an optimality gap 25.4%. Those demonstrate the limited capability of GUROBI solver on finding satisfactory solutions for somewhat large instances. Instead, the proposed approach can deal with the problem more efficiently in terms of solving time and solution quality.

5.3. Comparisons between the Fixed Timetable and Variable Timetable Approaches. In this section, we will solve all the instances by the designed approach which has two main procedures, namely, the Benders-and-Price approach and the BAP algorithm, with different values of ϕ. If $\phi = 0$, the fixed timetable (FT) approach is implemented, whereas $\phi > 0$ means the variable timetable (VT) approach is adopted. If $\phi = 0$ is set, i.e., timetable modification is not allowed, the path-based model is actually solved by the BAP algorithm without involving Benders decomposition. Table 4 reports the experimental results for FT approach and VT approach.

For the first three instances, the integer optimal solution can be found within 9 s without implementing the BAP algorithm. That is, after Benders-and-Price algorithm is terminated, the solutions obtained are integer valued. The solution quality in terms of the number of used vehicles does not have any improvement. For the last two instances, the final solutions are fractional in the first phase, so we have to perform the BAP algorithm to obtain integer solutions, which consumes most of the solving time. The number of used vehicles reduces by 1 and 2 in the instances (20,10,132) and (10,8,228), respectively. However, it should be borne in mind that not all the nodes in the BAP trees are solved by column generation in order to accelerate the convergence of BAP algorithm, which may make the final solution not optimal. That is, one should not come to the conclusion that the solution quality is improved only according to the required fleet size. Coincidently, both instances solved by VT approach have no branching procedures. Therefore, the solutions obtained by VT approach are optimal. As for FT approach, the lower bound of fleet size for FT approach (denoted as $\varphi_{\mathrm{LBFZ}}(\mathrm{FT})$, $\varphi_{\mathrm{LBFZ}}(\mathrm{VT})$ for VT approach correspondingly) is available, and if the actually required fleet size for FT approach (denoted as $\varphi_{\mathrm{ARFZ}}(\mathrm{FT})$, $\varphi_{\mathrm{ARFZ}}(\mathrm{VT})$ for VT approach correspondingly) in the final result is equal to $\varphi_{LBFZ}(\mathrm{FT})$, namely, $\varphi_{\mathrm{ARFZ}}(\mathrm{FT}) = \varphi_{LBFZ}(\mathrm{FT})$, then the corresponding solution of FT approach is optimal. To derive $\varphi_{\mathrm{LBFZ}}(\mathrm{FT})$, let us see Figure 4, which shows the lower and upper bound of objective values at iterative process for VT approach, and timetable deviations are shown as the scatter plot. At first iteration, from the upper bound value 2533 (the optimal objective value of the root node of BAP tree in FT approach is the same as this value), we can deduce the $\varphi_{LBFZ}(\mathrm{FT}) = \lceil 2533/c^R \rceil = 26$, where $\lceil\cdot\rceil$ denotes the arithmetic that rounds up to the nearest integer. Furthermore, $\varphi_{ARFZ}(\mathrm{FT}) = 26$ (see Table 4) is the same as $\varphi_{LBFZ}(\mathrm{FT})$, so the final solutions obtained by FT approach are also optimal. Thus, compared to FT approach, the solution quality is certainly improved by VT approach concerning the required

Table 4: The solutions for FT approach and VT approach.

Instance	FT approach				VT approach			
	Used vehicles	Canceled trips	Timetable deviation (min)	Solving time (s)	Used vehicles	Canceled trips	Timetable deviation (min)	Solving time (s)
(80,60,30)	4	2	0	3	4	2	0	3
(60,40,42)	8	0	0	5	8	0	0	6
(40,20,68)	10	0	0	9	10	0	0	9
(20,10,132)	17	0	0	690	16	0	10	193
(10,8,228)	26	0	0	2581	24	0	70	3481

fleet size and also the total cost. In addition, we can deduce that $\varphi_{LBFZ}(\mathrm{VT}) = \lceil (2470 - 70)/c^R \rceil = 24$, which is equal to $\varphi_{ARFZ}(\mathrm{VT}) = 24$ (see Table 4).

Figure 5 displays the timetable with vehicle schedule solution of VT approach for the instance (10,8,228). In this figure, the routes are divided into two parts according to the origin terminal of the first trip in the corresponding route for clarity. In order to display the alterations of the departure times of trips, the trips whose departure times are changed in t_0 are also depicted as dotted lines show.

The local enlarged picture of the right upper corner (the hatched rectangle area) of Figure 5 is shown in Figure 6, which gives more details about the alterations. For example, let us see a part of one route as the thick line shows. For the trips in the route, the departure time of trip 199 is 1 min ahead of schedule to satisfy the minimum turnaround time between trips 199 and 91, trip 211 is 5 min backward to satisfy the minimum turnaround time between trips 211 and 102, and trip 221 is 5 min forward to fulfill the minimum turnaround time between trips 102 and 221.

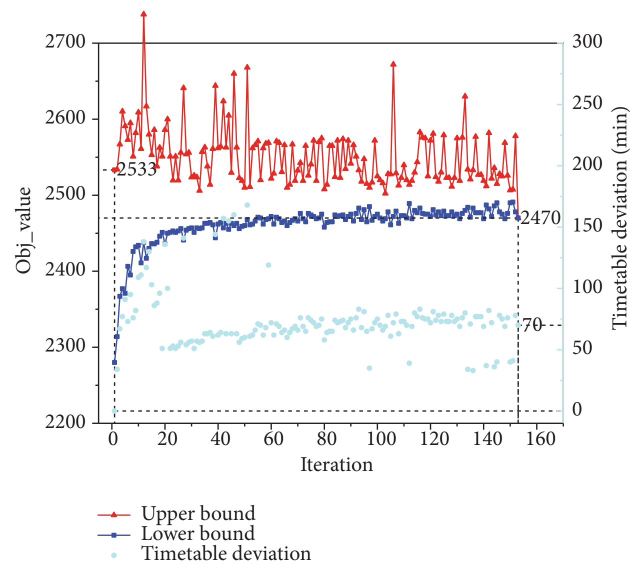

Figure 4: Upper bounds and lower bounds of VT approach with timetable deviations at Benders-and-Price iterations.

6. Conclusions and Future Works

Considering the inefficiency of using GUROBI to solve the connection-based model for VSPVT-LT when the size of instances is relatively large, this paper formulated a path-based model, which can be solved by a bespoke approach working in two phases. In the first phase, Benders decomposition is used to decompose the original model into a timetabling problem in BMP and a VSP in BSP. Moreover, the LP-relaxation of RBSP is solved by column generation. After the Benders-and-Price algorithm is finished, a BAP algorithm is utilized to obtain an integer solution. Additionally, a vertices-constrained shortest path algorithm modified from Bellman-Ford algorithm is proposed to deal with the pricing problems of column generation. The experimental results show that the VT approach can reduce the needed number of vehicles in comparison with FT approach on the test cases, and the proposed algorithm performs well to handle the VSPVT-LT. Our future researches will concentrate on the following main extensions:

(1) In our case studies, with the Benders cuts added to BMP, the model becomes more and more difficult to handle by optimization software. An efficient algorithm should be further developed to solve BMP, such as the three-phase method [24, 25], which has one more phase because it relaxes all integrality constraints in the BMP in first phase and reintroduces the integrality constraints to the BMP in the next phase.

(2) Regarding the decrease in the number of iterations of Benders decomposition, Pareto-optimal cut method [25] can be employed. The approach exploits the fact that for a degenerate RBSP there exists multiple optimal dual solutions. Thus, it is likely to select the dual solution that is the nearest to the interior of the BMP polyhedron.

Appendix

The Connection-Based Model for VSPVT-LT

The objective is to minimize the total costs including (1) the timetable deviation cost, (b) canceling trips cost, and (c) using vehicles cost, as (A.1) shows. Canceling trips is allowable to make sure that the model is always feasible, and relative large penalty costs are set for canceling trips. The connection-based formulation for VSPVT-LT is now formulated as follows.

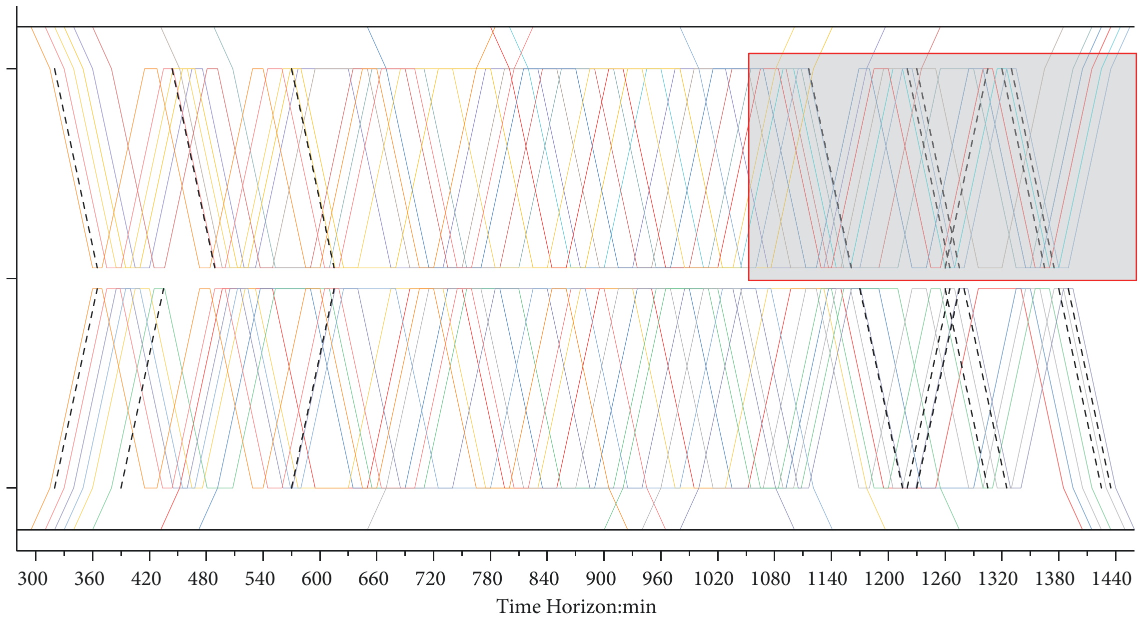

FIGURE 5: The schedule solution by VT approach.

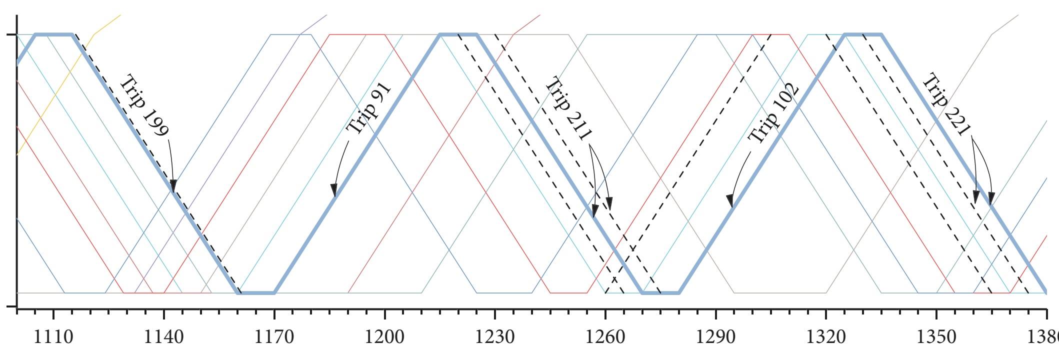

FIGURE 6: The more details of modifications for the timetable in Figure 5.

$$\min \quad \sum_{i\in T}\lambda_i \left|d_i - D_i\right| + \sum_{i\in T} w_i m_i + \sum_{k\in K}\sum_{r\in R_k}\sum_{j\in O_k(n+2k-1)} c_r x^r_{n+2k-1,j} \tag{A.1}$$

$$\text{s.t.} \quad a_i - d_i = l_i \quad \forall i \in T \tag{A.2}$$

$$-\Delta d_i^- \le d_i - D_i \le \Delta d_i^+ \quad \forall i \in T \tag{A.3}$$

$$d_i - d_j \ge h \quad \forall k \in K,\ i, j \in T^d(k) : j < i \tag{A.4}$$

$$d_j - a_i \ge \chi_{\min} - \left(1 - \sum_{r\in R} x^r_{ij}\right) M \quad \forall (i,j) \in I \tag{A.5}$$

$$d_j - a_i \le \chi_{\max} + \left(1 - \sum_{r\in R} x^r_{ij}\right) M \quad \forall (i,j) \in I \tag{A.6}$$

$$\sum_{k\in K}\sum_{r\in R_k}\sum_{j\in O_k(i)} x^r_{ij} = 1 - m_i \quad \forall i \in T \tag{A.7}$$

$$\sum_{i\in I_k(j)} x^r_{ij} - \sum_{i\in O_k(j)} x^r_{ji} = 0 \quad \forall k \in K,\ r \in R_k,\ j \in T \tag{A.8}$$

$$x^r_{n+2k-1,i} + x^r_{n+2k-1,j} \le 1 \quad \forall k \in K,\ r \in R_k,\ i, j \in T^d(k) : j < i \tag{A.9}$$

$$\sum_{(i,j)\in A_k} x^r_{ij} \le N_r + 1 \quad \forall k \in K,\ r \in R_k \tag{A.10}$$

Constraints (A.2) link the arrival time and the departure time of each trip. Constraints (A.3) require that the departure time deviation of each trip should not exceed the corresponding given range. Constraints (A.4) ensure that the minimum headway time is satisfied. For any two trips $i, j \in T$ with the same origin terminal, $j < i$ means that the departure time of trip j is before i in timetable t_0. We suppose that the departure orders of trips with the same origin terminal cannot be changed in the timetable alteration. Constraints (A.5) and (A.6) describe the connections between the variables in timetabling and vehicle scheduling. Constraints (A.5) guarantee that if trips i and j are operated consecutively by the same vehicle, there is sufficient time for vehicle preparation to execute trip j after finishing trip i. The maximum permissible times are set for the turnaround operations as constraints (A.6) show. Constraints (A.7) make sure that a noncanceled trip is executed by exactly one vehicle, and a canceled trip should not be covered by any vehicle. Constraints (A.8) are flow conservation constraints that guarantee that each used vehicle can turn back to its starting terminal after finishing daily service. Constraints (A.9) ensure that if trips i and j use different vehicles, they are both the first trips in their corresponding routes. Constraints (A.10) guarantee that the number of trips executed by vehicle $r \in R$ should not be more than N_r.

Data Availability

The data used to support the findings of this study are included in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Key R&D Program of China (2018YFB1201402, 2016YFE0201700) and Key issues of China Railway Corporation (No. 2017X004-D, No. 2017X004-E).

References

[1] A. Ceder, "Urban transit scheduling: Framework, review and examples," *Journal of Urban Planning and Development*, vol. 128, no. 4, pp. 225–244, 2002.

[2] A. Ceder, *Public Transit Planning and Operation: Modeling, Practice and Behavior*, CRC Press, Boca Raton, Fla, USA, 2016.

[3] Y. Wu, H. Yang, J. Tang, and Y. Yu, "Multi-objective re-synchronizing of bus timetable: Model, complexity and solution," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 149–168, 2016.

[4] H. Niu, X. Zhou, and R. Gao, "Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: nonlinear integer programming models with linear constraints," *Transportation Research Part B: Methodological*, vol. 76, pp. 117–135, 2015.

[5] T. Liu and A. A. Ceder, "Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: a bi-objective bi-level model using deficit function approach," *Transportation Research Part B: Methodological*, vol. 23, pp. 341–361, 2017.

[6] D. Huisman, *Integrated and dynamic vehicle and crew scheduling [Ph.D. thesis]*, Erasmus University Rotterdam, 2004.

[7] S. Bunte and N. Kliewer, "An overview on vehicle scheduling models," *Public Transport*, vol. 1, no. 4, pp. 299–317, 2009.

[8] H. Niu, X. Zhou, and X. Tian, "Coordinating assignment and routing decisions in transit vehicle schedules: A variable-splitting Lagrangian decomposition approach for solution symmetry breaking," *Transportation Research Part B: Methodological*, vol. 107, pp. 70–101, 2018.

[9] I. Steinzen, V. Gintner, L. Suhl, and N. Kliewer, "A time-space network approach for the integrated vehicle-and crew-scheduling problem with multiple depots," *Transportation Science*, vol. 44, no. 3, pp. 367–382, 2010.

[10] A. A. Bertossi, P. Carraresi, and G. Gallo, "On some matching problems arising in vehicle scheduling models," *Networks. An International Journal*, vol. 17, no. 3, pp. 271–281, 1987.

[11] A. Costa, I. Branco, and J. M. Pinto Paixão, "Vehicle scheduling problem with multiple type of vehicles and a single depot," in *Computer-Aided Transit Scheduling*, vol. 430 of *Lecture Notes in Economics and Mathematical Systems*, pp. 115–129, Springer, Berlin, Germany, 1995.

[12] A. Hadjar, O. Marcotte, and F. Soumis, "A branch-and-cut algorithm for the multiple depot vehicle scheduling problem," *Operations Research*, vol. 54, no. 1, pp. 130–149, 2006.

[13] L. Bodin, B. Golden, A. Assad, and D. Bull, "Routing and scheduling of vehicles and crews: the state of the art," *Computers and Operations Research*, vol. 10, no. 2, pp. 63–211, 1983.

[14] R. Freling and J. M. P. Paixão, "Vehicle scheduling with time constraint," in *Computer-Aided Transit Scheduling*, vol. 430 of *Lecture Notes in Economics and Mathematical Systems*, pp. 130–144, Springer, Berlin, Germany, 1995.

[15] G. Desaulniers, J. Lavigne, and F. Soumis, "Multi-depot vehicle scheduling problems with time windows and waiting costs," *European Journal of Operational Research*, vol. 111, no. 3, pp. 479–494, 1998.

[16] A. Hadjar and F. Soumis, "Dynamic window reduction for the multiple depot vehicle scheduling problem with time windows," *Computers & Operations Research*, vol. 36, no. 7, pp. 2160–2172, 2009.

[17] R. G. Ibarra-Rojas and Y. A. Rios-Solis, "An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks," *Transportation Research Part B: Methodological*, vol. 70, no. 7, pp. 35–46, 2014.

[18] A. P. R. Van den Heuvel, J. M. van den Akker, and M. E. van Kooten Niekerk, Integrating timetabling and vehicle scheduling in public bus transportation, Erasmus University Rotterdam, Rotterdam, The Netherlands, 2008.

[19] V. Guihaire and J.-K. Hao, "Transit network Re-timetabling and vehicle scheduling," *Communications in Computer and Information Science*, vol. 14, pp. 135–144, 2008.

[20] T. Liu, A. Ceder, and S. Chowdhury, "Integrated public transport timetable synchronization with vehicle scheduling," *Transportmetrica A: Transport Science*, vol. 6, pp. 1–23, 2017.

[21] J. P. Fonseca, E. van der Hurk, R. Roberti, and A. Larsen, "A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling," *Transportation Research Part B: Methodological*, vol. 109, pp. 128–149, 2018.

[22] J.-F. Cordeau, F. Soumis, and J. Desrosiers, "Benders decomposition approach for the locomotive and car assignment problem," *Transportation Science*, vol. 34, no. 2, pp. 133–149, 2000.

[23] J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers, "Benders decomposition for simultaneous aircraft routing and crew scheduling," *Transportation Science*, vol. 35, no. 4, pp. 375–388, 2001.

[24] M. I. Restrepo, B. Gendron, and L.-M. Rousseau, "Combining Benders decomposition and column generation for multi-activity tour scheduling," *Computers & Operations Research*, vol. 93, pp. 151–165, 2018.

[25] A. Mercier, J.-F. Cordeau, and F. Soumis, "A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem," *Computers & Operations Research*, vol. 32, no. 6, pp. 1451–1476, 2005.